JURNAL BUFFER INFORMATIKA
Volume 12 Nomor 1, April 2026
p-ISSN : 2527-4856, e-ISSN : 2614-5413
https://journal.fkom.uniku.ac.id/index.php/buffer

# IMPLEMENTASI ALGORITMA TIGER HASH SEBAGAI PENGAMANAN *APPLICATION PROGRAMMING INTERFACE* PADA KOMUNIKASI DATA BERBASIS *WEB SERVICES*

**Aah Sumiah[1)], Fitra Nugraha, M. Kom[2)] Iwan Lesmana,M.Kom[3)], Mohammad Irwansyah Somantri[4)]**

[1,2,3),] *Fakultas Ilmu Komputer, Universitas Kuningan*
*Jl. Cut Nyak Dhien No.36A, Cijoho, Kec. Kuningan, Kabupaten Kuningan, Jawa Barat 45513*
*Email : aah.sumiah@uniku.ac.id[1)], fitra@uniku.ac.id[2)], iwanlesmana@uniku.ac.id[3)],*
mirwansyah1923@gmail.com [4)]

***Abstrak***

Web Services merupakan suatu sistem perangkat lunak yang menyediakan antarmuka untuk mengakses sekumpulan data maupun layanan yang tersimpan dalam basis data melalui jaringan internet. Pada kondisi standarnya, web services tidak dilengkapi dengan mekanisme keamanan yang memadai untuk melindungi akses data. Meskipun sebagian implementasi web services telah menggunakan JSON Web Token (JWT) sebagai lapisan keamanan autentikasi dengan algoritma HS256 berbasis SHA256, algoritma tersebut diketahui memiliki celah keamanan berupa serangan Preimage Attack dalam Pseudo Collision yang dapat mengancam integritas data. Serangan Preimage menargetkan fungsi hash kriptografi dengan tujuan menemukan pesan yang menghasilkan nilai hash tertentu, sehingga kerahasiaan token dapat terancam. Untuk mengatasi kelemahan tersebut, diperlukan penerapan algoritma hash yang lebih kuat sebagai pengganti SHA256 pada JWT. Algoritma Tiger Hash dipilih sebagai solusi karena mampu memproses pesan dalam blok berukuran 512 bit dan menghasilkan message digest berukuran 192 bit. Tiger Hash bekerja melalui 24 putaran yang terbagi dalam tiga pass, masing-masing terdiri dari delapan putaran dalam, dengan memanfaatkan operasi XOR, penambahan atau pengurangan, rotasi, serta pencarian S-boxes. Hasil akhir penelitian ini adalah sebuah web services yang mengintegrasikan algoritma Tiger Hash sebagai mekanisme pengamanan komunikasi data.

***Kata Kunci : Web Services; JSON Web Token (JWT); Keamanan Data; Komunikasi Data; Algoritma Tiger Hash***

***Abstract***

*Web Services is a software system that provides an interface for accessing a collection of data or services stored in a database over the internet. In its default state, web services are not equipped with adequate security mechanisms to protect data access. Although some web services implementations have adopted JSON Web Token (JWT) as an authentication security layer using the HS256 algorithm based on SHA256, this algorithm is known to have a security vulnerability in the form of a Preimage Attack in Pseudo Collision that can threaten data integrity. Preimage attacks target cryptographic hash functions with the aim of finding messages that produce a specific hash value, thereby threatening the confidentiality of the token. To address this weakness, a stronger hash algorithm is needed to replace SHA256 in JWT. The Tiger Hash algorithm was chosen as a solution because it is capable of processing messages in 512-bit blocks and producing a 192-bit message digest. Tiger Hash operates through 24 rounds divided into three passes, each consisting of eight inner rounds, utilizing XOR operations, addition or subtraction, rotation, and S-boxes lookup. The final result of this research is a web service that integrates the Tiger Hash algorithm as a data communication security mechanism.*

***Keywords: Web Services; JSON Web Token (JWT); Data Security; Data Communication; Tiger Hash Algorithm***

## 1. PENDAHULUAN

Perkembangan teknologi informasi mendorong semakin luasnya penggunaan web services sebagai media pertukaran data antar sistem. Web services memungkinkan berbagai aplikasi untuk saling berkomunikasi dan berbagi sumber daya secara terbuka melalui jaringan internet. Namun, keterbukaan ini juga membawa risiko keamanan yang tidak dapat diabaikan. Berdasarkan kajian yang dilakukan oleh Rahmatulloh, Sulastri, dan Nugroho (2018), keamanan web services masuk ke dalam sepuluh kerentanan teratas pada keamanan Application Programming Interface (API) menurut *The Open Web Application Security Project* (OWASP), yang menunjukkan bahwa web services yang kurang terlindungi menjadi sasaran empuk bagi penyerang [1][2].

Salah satu arsitektur yang umum digunakan dalam web services adalah Representational State Transfer (REST). REST memanfaatkan protokol Hypertext Transfer Protocol (HTTP) sebagai standar komunikasi data berbasis web. Danutirta et al. (2021) menyatakan bahwa layanan RESTful memiliki karakteristik yang lebih terukur, andal, dan transparan, serta dinilai cocok untuk digunakan pada jaringan wireless karena paket data dapat ditransmisikan dari node sumber ke tujuan secara fleksibel [3].

Meskipun demikian, RESTful API pada umumnya tidak dilengkapi dengan mekanisme autentikasi bawaan, sehingga siapapun berpotensi mengakses, mengubah, bahkan menghapus data yang tersimpan di sisi server. Danutirta et al. (2021) menawarkan solusi atas permasalahan ini dengan menyatakan bahwa JSON Web Token (JWT) dapat menjadi solusi efektif dalam meningkatkan keamanan autentikasi pada aplikasi berbasis RESTful API [3].

JWT adalah token berbentuk string yang tersusun atas tiga komponen utama, yaitu header, payload, dan signature, yang berfungsi dalam proses otentikasi dan pertukaran informasi [4]. Ketiga komponen tersebut dienkripsi menggunakan Base64, dengan penambahan algoritma hash pada bagian signature untuk memperkuat keamanannya.

Algoritma hash yang lazim digunakan pada signature JWT adalah HS256 atau HMAC menggunakan SHA256. Namun, Li, Isobe, dan Shibutani (2012) menemukan bahwa SHA256 memiliki kelemahan yang dapat dieksploitasi melalui serangan Preimage Attack dalam Pseudo Collision dengan teknik Meet In The Middle, sehingga penggunaan HS256 pada JWT web services dinilai belum sepenuhnya aman.

Sebagai respons terhadap kelemahan tersebut, Algoritma Tiger Hash hadir sebagai alternatif yang lebih andal. Tiger Hash dirancang pada tahun 1996 oleh Ross Anderson dan Elli Biham, dan dikembangkan khusus sebagai solusi ketika ditemukan kelemahan pada algoritma MD dan SHA. Yuni (2021) menjelaskan bahwa Tiger Hash merupakan fungsi hash kriptografis yang bekerja secara iteratif dalam memproses blok 512-bit dan menghasilkan nilai hash 192-bit [5]. Lebih lanjut, Anderson dan Biham (1997) menegaskan bahwa Tiger merupakan algoritma yang kuat sekaligus efisien, dengan kecepatan yang melampaui SHA-1 pada prosesor 32-bit dan bahkan tiga kali lebih cepat pada prosesor 64-bit [6].

## 2. METODE PENELITIAN

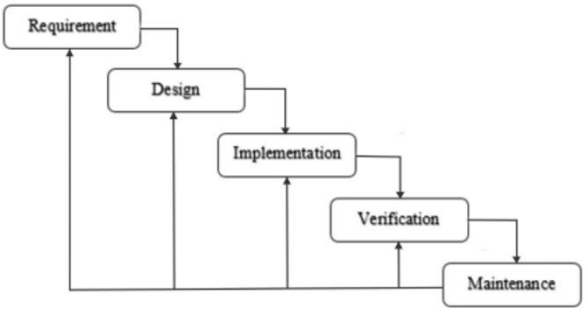
Penelitian ini menggunakan dua pendekatan utama, yaitu metode pengumpulan data dan metode pengembangan sistem. Pengumpulan data dilakukan melalui studi pustaka, yakni dengan mengkaji berbagai literatur ilmiah yang relevan dengan topik penelitian, meliputi jurnal, buku, dan artikel yang berkaitan dengan web services, JWT, dan algoritma kriptografi.

Sementara itu, pengembangan sistem dilakukan menggunakan metode Waterfall. Metode ini merupakan pendekatan pengembangan perangkat lunak yang bersifat sistematis dan sekuensial, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahapan berikutnya [7]. Pendekatan ini dipilih karena kebutuhan sistem telah terdefinisi dengan jelas sejak awal penelitian.

Metode Waterfall dalam penelitian ini terdiri dari lima tahapan, yaitu requirements, design, implementation, verification, dan maintenance, sebagaimana digambarkan pada Gambar 1.

Pada tahap requirements, dilakukan identifikasi dan pendefinisian kebutuhan sistem secara menyeluruh, mencakup kebutuhan fungsional dan non-fungsional. Tahap design berfokus pada perancangan arsitektur sistem berdasarkan

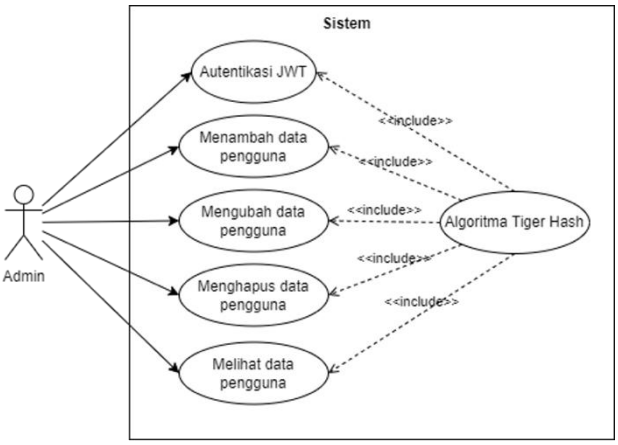
kebutuhan yang telah ditetapkan, dengan menggunakan Unified Modeling Language (UML) yang meliputi Use Case Diagram, Activity Diagram, dan Sequence Diagram. Tahap implementation merupakan tahap realisasi sistem berdasarkan rancangan yang telah dibuat. Tahap verification dilakukan melalui pengujian Black Box Testing untuk memastikan sistem berjalan sesuai spesifikasi. Terakhir, tahap maintenance bertujuan untuk memastikan sistem tetap berjalan optimal dan data yang tersimpan terorganisir dengan baik.




*Gambar 1 Tahapan metode waterfall (Pressman, 2015)*

## 3. HASIL DAN PEMBAHASAN

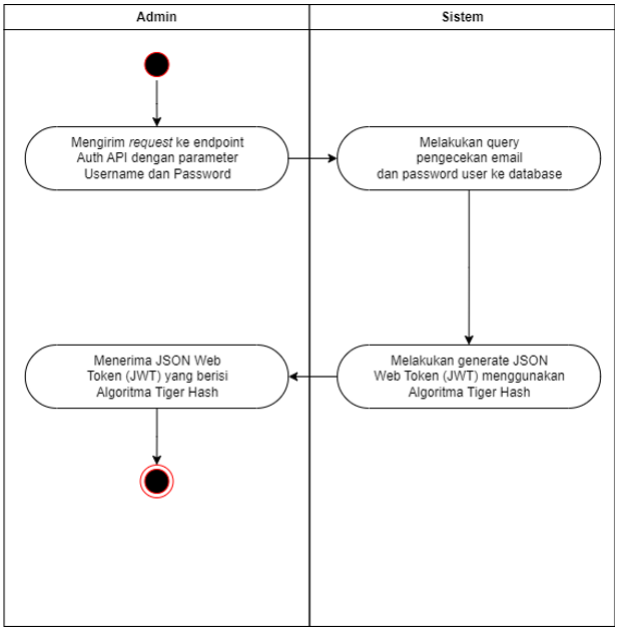
Tahap perancangan sistem dalam penelitian ini menggunakan pemodelan UML. Alat bantu UML digunakan untuk menganalisis dan merancang sistem usulan. sekumpulan pemodelan konvensional yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan obyek[8]. *Use Case Diagram* menggambarkan fungsionalitas sistem atau persyaratan yang harus dipenuhi sistem dari pandangan pemakai[9]. Dalam aplikasi yang dibuat, aktor memiliki beberapa perlakuan umum yang dapat dilakukan. *Use Case Diagram* menggambarkan pengguna dengan aplikasi. Pada gambar 2 dapat kita lihat *Use Case Diagram* untuk penelitian ini.




*Gambar 2 Usecase Diagram*

Gambar 2 menunjukan bahwa terdapat satu user yang memiliki beberapa tindakan pada menu utama, yaitu pada bagian ini akan dijelaskan masing-masing deskripsi dari Use Case Diagram yang dilakukan oleh Admin per Use Case nya, mulai dari Admin, Autentikasi JWT, menambahkan data pengguna, mengubah data pengguna, menghapus data pengguna, dan melihat data pengguna.

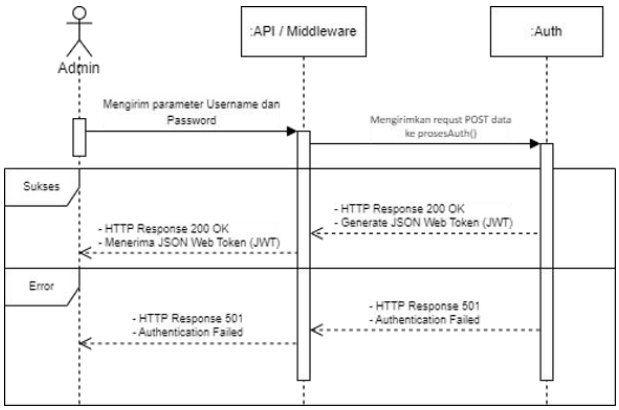
Activity Diagram merupakan gambaran workflow (alirankerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak[10]. Pada activity diagram ini menggambarkan proses yang berjalan di sistem yang dilakukan oleh pengguna dari awal membuka aplikasi sampai menutup aplikasi. Gambar 3 merupakan activity diagram autentikasi JWT.




*Gambar 3 Activity Diagram Autentikasi JWT*

Perancangan ini akan menampilkan aksi maupun reaksi yang didapat akibat interaksi yang diberikan. Proses selanjutnya, setelah dibuat activity

diagram adalah membuat sequence diagram. Sequence diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output. Adapun sequence diagram yang dirancang dari aplikasi ini digambarkan pada Gambar 4.

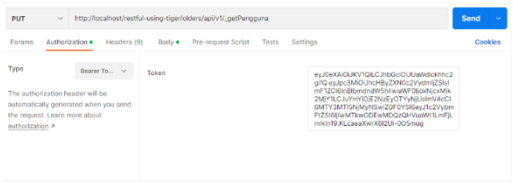



*Gambar 5 Sequence Diagram Autentikasi JWT*

Setelah tahapan *Requirements* dan *Design* dilakukan, maka selanjutnya akan dilakukan tahapan *Implementation* dan *Verification*. Pada tahapan *Implementation* akan lebih fokus pada hasil perancangan yang diusulkan. Sementara itu, pada tahapan *Verification* lebih fokus pada proses pengujian dari aplikasi yang dirancang.

### 3.1. *Implementation*

Setelah mengalami sederetan fase, pada tahapan *Implementation* ini sudah dapat menghasilkan suatu program. Autentikasi JWT merupakan sebuah proses untuk mendapatkan nilai token pada sebuah *web services* dengan mengirimkan parameter *Username* dan *Password*. Gambar 5 merupakan hasil dari autentikasi JWT.



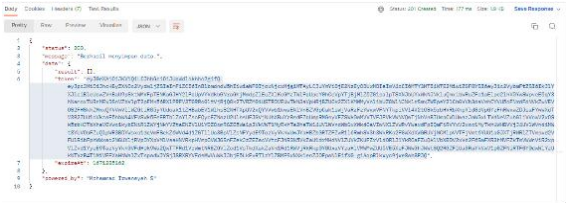
*Gambar 6 Peletakan token JWT*

Proses menambahkan data pengguna akan dapat berjalan jika token telah didapatkan baik melalui autentikasi JWT sebagai pembuka, maupun setelah melakukan request terhadap usecase lainnya. Gambar 6 menggambarkan hasil dari penambahan data pengguna.

*Gambar 4 Autentikasi JWT*

Token JWT akan didapatkan setelah melakukan *request* apapun baik *request* menambah data pengguna, mengubah data pengguna, menghapus data pengguna, maupun melihat data pengguna.

Token JWT digunakan sebagai kunci untuk melakukan segala *request* yang pengguna ingin lakukan, token JWT bertipe *Bearer Authentication*. Gambar 7 merupakan pengisian Token JWT sebagai kunci dalam segala *request* pada *web services*.



*Gambar 7 Menambah data pengguna*

### 3.2. *Verification*

Tahap *verification* merupakan tahap uji coba atau *testing* dari aplikasi yang dirancang. Proses pengujian sendiri memiliki banyak jenis. Proses pengujian merupakan proses mengeksekusi aplikasi untuk menentukan apakah aplikasi perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan lingkungan yang diinginkan.

Pengujian BlackBox adalah proses pengujian aspek fundamental aplikasi tanpa memperhatikan struktur logika internal perangkat lunak. Proses pengujian ini dilakukan untuk mengetahui apakah aplikasi perangkat lunak dapat berjalan dan berfungsi dengan benar. Proses pengujian blackbox dilakukan pada *endpoint* autentikasi JWT. Adapun hasil pengujian blackbox yang telah dilakukan pada *endpoint* autentikasi JWT dapat dilihat pada Tabel 1.

*Tabel 1 Blackbox autentikasi JWT*

| No | Fungsi yang di uji | Cara Menguji | Hasil yang diharapkan | Hasil yang keluar |
|---|---|---|---|---|
| 1 | Autentikasi JWT | Admin mengirim *parameter Username* dan *Password* ke *endpoint API* | Admin menerima status 200 OK dan *JSON Web Token.* | Sesuai dengan harapan *valid.* |
| 2 | Autentikasi JWT | Admin tidak mengirimkan *parameter Username* ke *endpoint API* | Admin menerima status 404, dan pesan "Maaf, Username salah!" | Sesuai dengan harapan *valid.* |
| 3 | Autentikasi JWT | Admin tidak mengirimkan *parameter Password* ke *endpoint API* | Admin menerima status 404, dan pesan "Maaf, Password salah!" | Sesuai dengan harapan *valid.* |

## 4. KESIMPULAN

Berdasarkan hasil penulisan dan penelitian skripsi implementasi algoritma *Tiger Hash* pada *web services*, dapat diperoleh kesimpulan sebagai berikut:

1. Algoritma *TigerHash* dapat diimplementasikan untuk pengamanan *web services* dengan *JSON Web Token* dengan cara mengubah *default hashing* dari *SHA256* menjadi *TigerHash*. Proses *hashing* akan menggunakan *library php* yaitu *hash_hmac()* dengan pemanggilan kode algoritma *TigerHash (php/hash_tiger.c)*. Proses akan dieksekusi oleh fungsi *JWT::Encode* dimana *Header, Payload dan Key* akan dikirim melalui fungsi *JWT::Sign()*, dan akan menghasilkan *Signature*.
2. Komunikasi atau bertukar informasi menggunakan *web services* dengan pengamanan menggunakan *JSON Web Services* dan algoritma *Tiger Hash* dapat mengamankan data agar pengguna yang tidak memiliki akses kepada database (Autentikasi JWT) tidak dapat menambah, melihat maupun menghapus data yang terdapat di database.

## 5. SARAN

Adapun saran yang dapat diberikan untuk mengembangkan penelitian ini adalah sebagai berikut:

1. Otorisasi *web services* dengan menggunakan algoritma Tiger Hash dapat dikembangan agar admin dapat mengelola permission setiap pengguna *web services* dalam mengkonsumsi data yang terdapat di database.
2. Algoritma Tiger Hash dapat diimplementasikan untuk mengamankan komunikasi antar perangkat IoT.

## UCAPAN TERIMA KASIH

Penelitian ini saya persembahkan untuk Universitas Kuningan, sebagai pemberi dana penelitian ini;

## DAFTAR PUSTAKA

[1] J. Li, T. Isobe, and K. Shibutani, "Converting meet-in-the-middle preimage attack into pseudo collision attack: Application to SHA-2," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7549 LNCS, pp. 264–286, 2012, doi: 10.1007/978-3-642-34047-5_16.

[2] A. Rahmatulloh, H. Sulastri, and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 7, no. 2, 2018, doi: 10.22146/jnteti.v7i2.417.

[3] N. R. Danutirta, Y. L. Christy, U. Pembangunan, and N. Veteran, "JSON Web Token Implementation for Dynamic Access Rights Authentication in Klinik Pratama UPN ' Veteran ' Yogyakarta Application Based on RESTful," pp. 51–60, 2021.

[4] R. A. Siregar, N. Hidayat, P. Studi, T. Informatika, F. I. Komputer, and U. Brawijaya, "Implementasi Algoritme BLAKE2B Pada JSON Web Token Untuk Mekanisme Autentikasi Query Language," vol. 5, no. 1, pp. 88–96, 2021.

[5] Yuni, "Mendeteksi Otentikasi File Dokumen Menerapkan Metode Tiger," *Pelita Informatika : Informasi dan Informatika*, vol. 9, pp. 155–160, 2021.

[6] R. (Cambridge U. Anderson and E. Biham, "Tiger : A Fast New Hash Function," 1997.

[7] Pressman R., *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku 1*. 2015.

[8] J. L. Whitten, L. D. Bentley, and K. C. Dittman, *Systems analysis and design methods*, 6th ed. New York: McGraw-Hill, 2004.
[9] M. K. Setiawan, Heru dan Khairuzzaman, "Perancangan Sistem Informasi Manajemen ProyekM : Sistem Informasi Kontraktor," *Jurnal Khatulistiwa Informatika*, vol. V, No 2. D, no. 2, pp. 103–111, 2017.
[10] R. AS and M. Shalahudin, *Modul Pembelajaran Rekayasa Perangkat Lunak : Terstruktur dan Beorientasi Objek*. Bandung: Modula, 2011.