JEPIN
(Jurnal Edukasidan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 5
No. 1
April 2019

JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service

Rohmat Gunawan[#1], Alam Rahmatulloh[#2]

[#]*Teknik Informatika, Fakultas Teknik, Universitas Siliwangi*
Jalan Siliwangi No.24 Kota Tasikmalaya 46115
[1]rohmatgunawan@unsil.ac.id
[2]alam@unsil.ac.id

Abstrak—**Permasalahan donor darah merupakan masalah disetiap negara, termasuk di Indonesia. Walaupun sudah ada sistem di Palang Merah Indonesia (PMI) namun belum bisa mengatasi permasalahan pencarian maupun distribusi donor darah. Sesuai *trend* sekarang di jaman gadget yaitu maraknya penggunaan Android, maka untuk mengatasi masalah ini diperlukan aplikasi berbasis Android. Sementara untuk integrasi dengan sistem yang sudah ada diperlukan *web service* sebagai *backend system* sehingga layanan donor darah dapat diakses oleh berbagai *platform*. Arsitektur yang digunakan pada *web service* menggunakan REST, namun masih ada beberapa masalah pada REST yaitu mengenai keamanan pada proses otentikasi. Pada arsitektur REST diperlukan metode otentikasi yang tidak bernegara *(stateless)*, salah satunya dapat menggunakan *JSON Web Token*. Hasil penelitian ini menunjukan bahwa penggunaan JSON Web Token *Authentication* pada *Web Service* and *Backend System Blood Donors* dapat membentuk sistem yang sangat skalabel, aman, mampu berinteraksi multi-platform serta dapat diandalkan.**

Kata kunci*—*Authentication*, *Interoperability*, JSON Web Token, REST, *Web Service

I. PENDAHULUAN

Interoperabilitas merupakan suatu kemampuan dua atau lebih sistem untuk dapat melakukan pertukaran informasi. Mengembangkan perangkat lunak sistem informasi yang dapat mendukung interoperabilitas merupakan salah satu hal yang tidak mudah dilakukan [1]. Interoperabilitas sistem melibatkan berbagai komponen yang heterogen. Arsitektur sistem yang dibangun dari komponen heterogen, seperti : platform, sistem operasi, bahasa pemgroman yang berbeda dan basis data, harus dapat diintegrasikan guna menyediakan layanan yang optimal. *Web Service* merupakan seperangkat standar dan metode pemrograman untuk berbagi data antara aplikasi perangkat lunak yang berbeda, mendistribusikan layanan melalui internet yang mendukung interoperabilitas sistem [2] [3].

Terdapat beberapa arsitektur *web service* yang dapat digunakan diantaranya : *Extensible Markup Language Remote Procedure Call (XML-RPC)*, *Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST)* [4], [5], [6]. XML-RPC merupakan cara tradisional yang digunakan dalam pertukaran dan integrasi sistem [6]. Web Service dengan arsitektur SOAP telah digunakan untuk mendukung proses integrasi dan scalability sistem [7], [8], [9]. Arsitektur metode REST merupakan konfigurasi dengan nilai *latency* terbaik untuk diimplementasikan dalam proses integrasi data [6]. Selain dari itu REST juga lebih baik dibanding SOAP dalam *respon time* dan *respon* berdasarkan ukuran data [10], [11], [12], [13].

Interoperabilitas perangkat lunak sistem informasi yang melibatkan berbagai macam komponen, yang memungkinkan menimbulkan celah yang dapat mengganggu keamanan sistem. Berbagai cara untuk mengurangi ancaman terhadap keamanan pada *web service* pernah dilakukan pada penelitian sebelumnya, diantaranya : pendekatan *Claim based Authentication*, *Token based Authentication*, *Secure with SSL* menggunakan ASP.NET MVC web API [14]. Penggunaan otentikasi berbasis *token* juga pernah dilakukan dalam penelitian [15], [16], [17]. Penelitian lainnya, otentikasi berbasis klaim ID *(identity)* [18], *SAML Technology* [19], dilakukan pada penelitian integrasi dan pertukaran data menggunakan format *Java Script Object Notation (JSON)* sehingga otentikasinya menggunakan *JSON Web Token (JWT)*. Keamanan REST menggunakan otentikasi JWT sudah pernah dilakukan [20] namun belum dicoba diterapkan pada beberapa platform. Berdasarkan latar belakang masalah tersebut, pada penelitian ini akan dirancang arsitektur *multi-platform* dengan memanfaatkan teknologi *web service* berbasis RESTful. JWT diterapkan untuk otentikasi sistem yang dapat diakses dari *platform* yang berbeda.

II. LANDASAN TEORI

A. *Representational State Transfer (REST)*

REST merupakan arsitektur *web service* yang dikembangkan dari beberapa gaya arsitektur berbasis jaringan yang sering diterapkan dalam layanan berbasis web [21]. Arsitektur REST pada umumnya dijalankan melalui HTTP *(Hypertext Transfer Protocol)*, melibatkan proses pembacaan *web page* tertentu yang memuat sebuah *file* XML atau JSON. Setiap permintaan bersifat independen, server tidak menyimpan keadaan permintaan apa pun. *Application Programming Interface* (API) yang mengikuti gaya REST disebut RESTful API. RESTful API menggunakan *Uniform Resource Identifier* (URI) untuk mewakili sumber daya. Setiap sumber data diidentifikasi menggunakan tautan URI. Metode yang digunakan dalam REST diantaranya: *GET* untuk mendapatkan sumber daya, *POST* digunakan untuk membuat sumber daya baru dan metode *PUT* digunakan untuk memperbarui sumber daya berdasarkan sumber daya. Sedangkan metode DELETE digunakan untuk menghapus sumber daya atau kumpulan sumber daya. Secara umum metode yang digunakan pada REST ditampilkan pada Tabel I.

TABEL I
CONTOH RESTFUL API

Resource	Method			
	Get	**Post**	**Put**	**Delete**
/api/student	Get a list of all student	Create a new list of student.	Update a list of student	Delete all student
/api/student/1	Get a student by student's ID	Treat as a collection. Create a new student in it.	If student exists, update the student. If student does not exist. Create a new student.	Delete the student.

B. *JSON Web Token (JWT)*

JWT merupakan sebuah token berbentuk string yang terdiri dari tiga bagian yaitu : *header*, *payload* dan *signature* yang digunakan untuk proses otentikasi dan pertukaran informasi [22]. Token terdiri dari dua jenis : token pembawa dan token pemegang kunci. Sedangkan berdasarkan tujuan terdapat dua skema : token identitas dan token akses [23]. Cara kerja JWT sama seperti password, ketika pengguna berhasil login maka server akan memberikan token yang disimpan di *local storage* atau *cookies browser*. Token digunakan untuk mengakses halaman tertentu, pengguna akan mengirim balik token tersebut sebagai bukti bahwa pengguna sudah berhasi login. Secara umum struktur JWT ditampilkan pada gambar 1.

```
Header : Algoritma dan Tipe Token
  {
  "alg" : "HS256",
  "typ" : "JWT"
  }
Payload : Data
  {
  "iss" : "#appbackend",
  "user" : "#username",
  "pass" : "password"
  }
Signature : hasil dari Hash
  {
  "Base64-encoded(Header.Payload)" + "key" +
"Algorithm"
  }
```

Gambar. 1 Struktur JWT

Gambar 1, menampilkan tiga bagian utama *(header, payload dan signature)* yang dapat digunakan dalam penggabungan untuk menghasilkan JWT. Adapun token yang dihasilkan dari JWT dibangun dengan formula seperti ditampilkan pada gambar 2.

$$\boldsymbol{Token} = f(Base64Encode) \sum_{n=\alpha,\beta}^{\infty} (header.payload.signature)$$

Gambar. 2 Formula JWT

Gambar 2 menampilkan formula JWT yang merupakan fungsi dari *Base64Encode* dengan parameter *header*, *plyaload* dan *signature*. Token yang diperoleh dari server, akan disisipkan pada *header Hyper Text Trasnfer Protocol (HTTP)*, seperti ditampilkan pada gambar 3

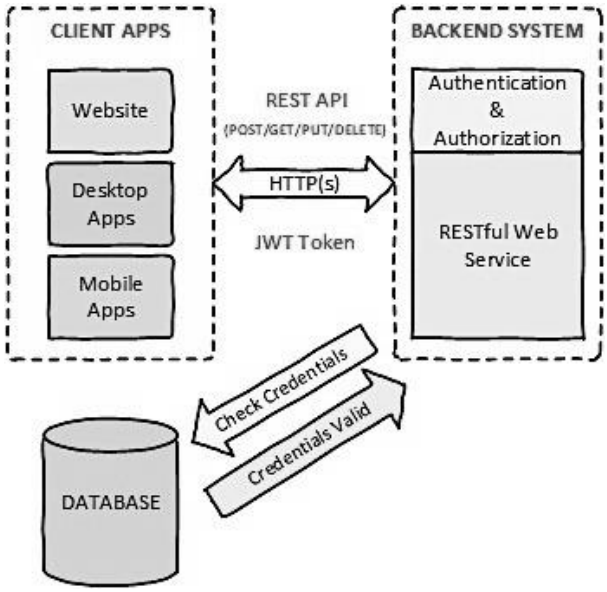
```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJrZF9wZW
dhd2FpIjoiUC0wMDYiLCJ1c2VybmFtZSI6InphbCIsImlhd
CI6MTUyNzM5NzIyMywiZXhwIjoxNTMwMDI2OTY5fQ.F
qHaqcJUOgXlSIg1Q7MuBhpqkWEqdGIAovzPOxqV2jg
```

Gambar. 3 Contoh Token pada *header HTTP*

III. METODOLOGI

A. *Perancangan Model*

Model yang diusulkan dalam penerapkan otentikasi berbasis token (JWT) pada arsitektur REST Web Service, secara umum dapat dilihat pada gambar 4.

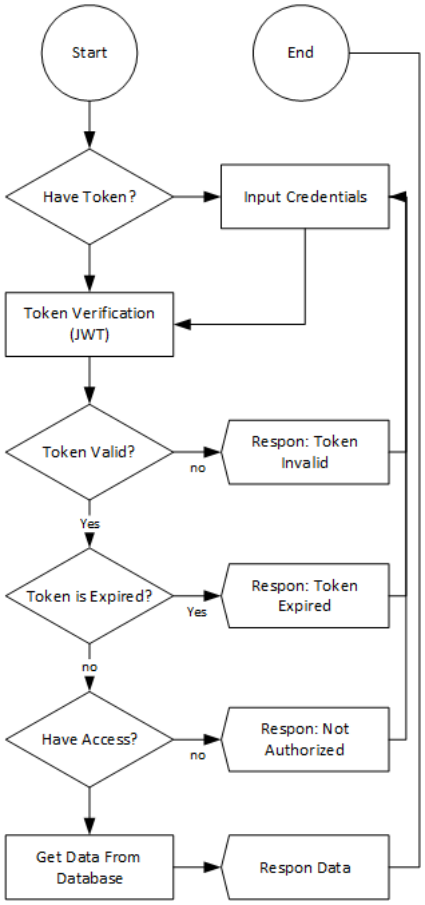


Gambar. 4 Penerapan JWT pada arsitektur RESTful *Web Service*

Pada gambar 4 ditampilkan *backend system* yang berperan menangani proses otentikasi dan otorisasi pada RESTful Web Service. setelah proses login berhasil server memberikan *respons* berupa JWT token sebagai kunci untuk mengakses sumber daya di server. REST API dapat diakses oleh berbagai client berbasis web, desktop maupun *mobile*. Adapun model yang diusulkan meliputi proses-proses sebagai berikut:

a. Pengguna terdiri dari: *client* berbasis web, desktop maupun *mobile*.
b. Sistem melakukan pemeriksaan terhadapan token. Apabila token tidak tersedia atau sudah kadaluwarsa pengguna diarahkan ke login page.
c. Pengguna melakukan login, *credentials passed to* API
d. *Backend system*: otentikasi user diperlukan untuk melakukan beberapa aksi
e. Database: melakukan pemeriksaan *credentials*
f. Jika kredensial valid, token JWT dikirim kembali ke *client*

Jika token tersebut belum kadaluwarsa, tahap selanjutnya adalah memeriksa hak akses yang dimiliki token dalam *payload*. Jika token tersebut memiliki hak akses ke sumber daya, *web service* donor darah akan meresponnya dengan sumber daya yang dibutuhkan oleh pengguna. Deskripsi dan alur *verifikasi token* ditampilkan pada gambar 5.

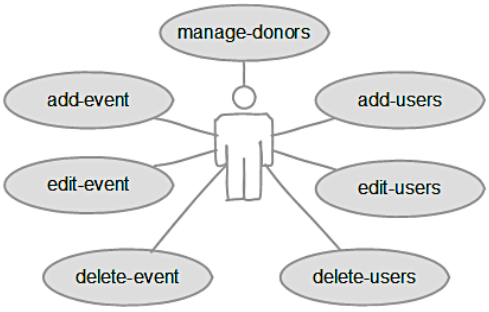


Gambar. 5 JSON Web Token Verification

Ada 3 level hak akses aktor yang berbeda dalam aplikasi ini, yaitu:

a. Administrator

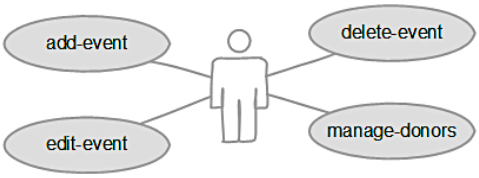
Level hak akses yang paling tinggi dimiliki admin, tugas utama admin adalah mengatur semua konfigurasi yang ada pada aplikasi. Adapun hak akses yang dimiliki admin dapat dilihat pada gambar 6.



Gambar. 6 Peran Admin

b. Operator

Level kedua adalah operator, peran ini dapat digunakan oleh lembaga pengelola donor darah. Hak akses pada level ini dapat dilihat pada gambar 7.



Gambar. 7 Peran Operator

c. Pengguna

Pengguna berperan sebagai end user, hak akses yang dimiliki pengguna diantaranya dapat berperan sebagai pendonor sekaligus orang yang *request* donor darah, secara rinci hak akses pengguna dapat dilihat pada gambar 8.

Gambar. 8 Peran Pengguna

B. Implementasi dan Pengujian

Pada tahap ini dilakukan implementasi JWT pada arsitektur RESTful webservice. Rancangan service diimplementasikan ke dalam pemrograman menggunakan bahasa pemgrograman PHP. JWT menggunakan standard HMAC-256. Pengujian Token dilakukan dari *client* dengan platform berbeda.

IV. HASIL DAN ANALISA

A. Implementasi

Perancangan yang telah dilakukan pada tahap sebelumnya diimplementasikan *(coding)* ke dalam bahasa pemrograman PHP dan pertukaran data menggunakan format *eXtensible Markup Language* (XML). Tahapan yang dilakukan dalam implementasi diantaranya pembuatan login form seperti ditampilkan pada gambar 9.

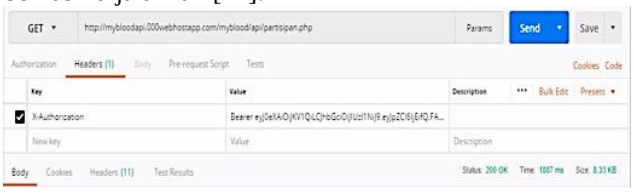
Gambar. 9 Login Form

Gambar. 10 Halaman *Event*

B. Pengujian

1. Pengujian *respon HTTP server*

Sebelum dilakukan pengujian, perlu disiapkan beberapa tahap seperti pembuatan atau penambahan data *dummy* yang terdiri dari pengguna, operator, admin, *event* kegiatan, dan lain sebagainya. Kemudian ujicoba dilanjutkan dengan login dari berbagai peran seperti admin, operator dan pengguna. Selanjutnya selama proses login akan diamati token JWT yang dihasilkan serta sumber daya yang diperoleh apakah sudah sesuai dengan hak akses yang dimiliki oleh peran tersebut. Pada proses parsing data JWT tersebut menggunakan beberapa tools diantaranya aplikasi *Postman*. Gambar 11. adalah contoh isi dari *header* untuk menguji proses *login* dengan hak akses sebagai Admin. Status *respon* HTTP dari *server* memberikan kode 200 yang artinya permintaan telah berhasil dijalankan [24].

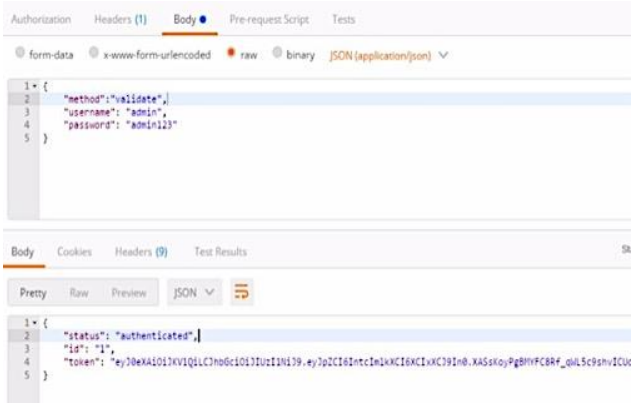


Gambar. 11 Header Login Form

2. Pengujian *Authentication* dengan JWT

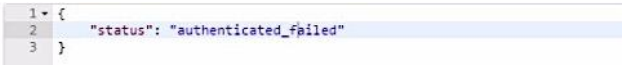
Konten login yang dikirim ke server dalam format JSON yang dikemas dalam tipe konten. HTTP body terdiri dari username dan password dalam format JSON.

Kemudian request dikirim ke layanan web, maka server akan memberikan respon seperti pada gambar 12.



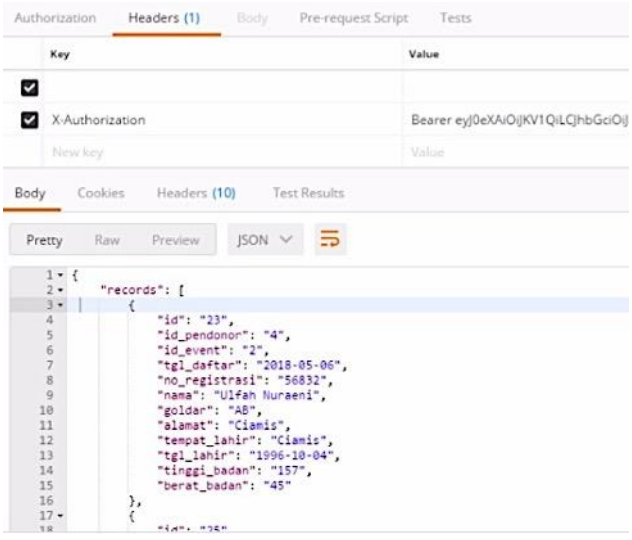
Gambar. 12 Body Login Form

Respon yang diterima berupa format JSON yang terdiri dari token dan data HTTP, nilai token tersebut akan berbeda beda karena token tersebut memiliki data pengguna dan hak akses yang berbeda. Pengujian selanjutnya dapat dilihat pada gambar 13. dengan memasukan kredensial login yang salah.



Gambar. 13 Respon Login Gagal

Percobaan selanjutnya menguji sumber daya yang dimiliki oleh token, hasilnya layanan web akan memerikan respon dari server jika konten tersedia dapat dilihat seperti pada gambar. 14.



Gambar. 14 Respon Content

3. Pengujian Data Token

Percobaan terakhir dilakukan dengan cara mengubah nilai token, token ditambahkan satu karakter. Token yang asli adalah sebagai berikut:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEifQ.FAxBH_j1Euf9xO2dYA8Yp4zYc8SNkFandAafMG4tz6E
```

Sedangkan token yang sudah dimodifikasi adalah sebagai berikut:

```
eyJA0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjEifQ.FAxBH_j1Euf9xO2dYA8Yp4zYc8SNkFandAafMG4tz6E
```

Karakter “A” yang ditulis dengan tebal adalah karakter yang disisipkan pada token, hasil yang diperoleh setelah memasukan token yang dimodifikasi adalah pesan token yang tidak valid.

V. KESIMPULAN

Hasil penelitian ini dapat disimpulkan bahwa otentikasi berbasis token menggunakan JSON Web Token telah berhasil diterapkan pada layanan web dan*backend systemblood donors*. Sehingga layanan web ini mampu mengatasi permasalahan interoperabilitas, sistem yang sudah ada di PMI mampu berinteraksi dengan aplikasi lain baik berbasis web, desktop maupun *mobile*. Selanjutnya, dengan penggunaan token kita bisa mengetahui siapa yang mengakses aplikasi, hak akses apa serta sumber daya apa saja yang dimilikinya. Jika konten token dimodifikasi atau dirubah secara ilegal, maka layanan web akan memberikan respon bahwa token itu tidak valid. Sehingga dengan penggunaan JSON Web Token Authentication dianggap aman.

Namun masih banyak kekurangan dari penelitian ini yaitu belum ada diskusi tentang penyimpanan token pada *local storage* (*cookies*), apakah sudah aman dan bagaimana cara menyimpan token yang benar. Selanjutnya perlu ada penelitian untuk mengetahui algoritma enkripsi yang lebih tepat digunakan baik kunci simetris maupun asimetris untuk JWT. Pada intinya ketika penggunaan JSON Web Token pada REST Web Service dirancang dengan perencanaan yang baik, JWT dapat membentuk sistem yang sangat skalabel, aman, mampu berinteraksi *multi-platform* serta dapat diandalkan.

REFERENSI

[1] R. Rezaei, T. K. Chiew, S. P. Lee and Z. S. Aliee, "Interoperability evaluation models: A systematic review," *Computers in Industry*, 2014.

[2] H. Hamad, M. Saad and R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," *International Arab Journal of e-Technology*, Vols. Vol. 1,, no. No. 3, January 2010.

[3] R. Gunawan and A. Rahmatulloh, "Implementasi Web Service pada Sistem Host-To-Host Pembayaran Biaya Akademik," *Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, vol. 7, no. 2, pp. 320-329, 2019.

[4] A. Dudhe and S. Sherekar, "Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment," *International Journal of Computer Applications*, 2014.

[5] P. Markey and T. G. Clynch, "A performance analysis of WS-* (SOAP) and RESTful Web Services for Implementing Service and Resource Orientated Architectures," in *The 12th Information*

Technology and Telecommunications (IT&T) Conference, Athlone IT, 2013.

[6] Yogiswara, Wijono and H. S. Dahlan, "Kinerja Web Service pada Proses Integrasi Data," *Jurnal EECCIS*, vol. 1, no. 1, 2014.

[7] R. I. Perwira and B. Santosa, "IMPLEMENTASI WEB SERVICE PADA INTEGRASI DATA AKADEMIK DENGAN REPLIKA PANGKALAN DATA DIKTI," *TELEMATIKA*, vol. 14, no. 1, 2017.

[8] A. Adi and Riyanto, "Web Service Useness as a Pharmacy Data Integration in RSU Banyumas," *Jurnal Informatika*, vol. 2, no. 4, 2013.

[9] M. R. S. Surendra, "Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile," *Jurnal Teknik Informatika (ULTIMATICS)*, vol. 6, no. 2, 2014.

[10] R. K. Sungkur and S. Daiboo, "SOREST, A Novel Framework Combining SOAP and REST for Implementing Web Services," in *The Second International Conference on Data Mining, Internet Computing, and Big Data (BigData2015)*, 2015.

[11] D. D. Rathod, "PERFORMANCE EVALUATION OF RESTFUL WEB SERVICES AND SOAP / WSDL WEB SERVICES," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 7, pp. 415-420, 2017.

[12] A. Memeti, F. Imeri and B. Cico, "REST vs. SOAP: Choosing the best web service while developing in-house web applications," *Journal of Natural Sciences and Mathematics of UT*, vol. 2, no. 3, pp. 63-68, 2017.

[13] V. Kumari, "Web Services Protocol: SOAP vs REST," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 4, no. 5, 2015.

[14] M. I. Hussain and N. Dilber, "Restful web services security by using ASP.NET web API MVC based," *Journal of Independent Studies and Research – Computing*, vol. 12, no. 1, 2014.

[15] 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.

[16] 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.

[17] 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.

[18] L. Xinhua, "The Design of Digital Campus Unified Identity Authentication System Based on Web Services," *Applied Mechanics and Materials*, pp. 2301-2304, 2013.

[19] I. I, P. M. R. Anand and V. Bhaskar, "Encrypted Token based Authentication with Adapted SAML Technology for Cloud Web Services," *Journal of Network and Computer Applications 99*, 2017.

[20] 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, pp. 131-137, 2018.

[21] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, University of California, Irvine, 2000.

[22] M. Jones, J. Bradley and N. Sakimura, "JSON Web Token (JWT)," 2015.

[23] K. Zheng and W. Jiang, "A Token Authentication Solution for Hadoop Based on Kerberos Pre-Authentication," *International Conference on Data Science and Advanced Analytics (DSAA)* , October 2014.

[24] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," *Internet Engineering Task Force (IETF)*, June 2014.