

INTEGRASI DATA BARANG PADA SISTEM INVENTORY CV. XYZ MENGUNAKAN WEB SERVICE

Dhani Trianggara¹⁾, Wina Witanti²⁾, Herdi Ashaury³⁾

^{1,2,3}Program Studi Informatika Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani,
Jl. Terusan Jenderal Sudirman, Cibeber, Kec. Cimahi Selatan, Kota Cimahi, Jawa Barat
Co Responden Email: dhanitrianggara17@if.unjani.ac.id

Abstract

In various fields the industry has used as well as implemented the concept of web services that provide message interoperability and data exchange interfaces. One of the uses for sharing data on data integration of one system with another system in this case study was carried out to manage iron supply goods data, in CV. XYZ Pratama Manufacture Engineering uses research methods of collecting data structures, architectural design data, building application web services, building Rest APIs, testing, and measuring and securing JWT HS 256 tokens. Building integration can formulate how to use CV Goods data. XYZ, becomes an Inventory system built using Web Service technology by managing item API data of json data type on the Rest Server so that it can be used by Client Rest. This research resulted in a Web Service through testing stages with different services with the total data requested and received which is 62.48 kb and the total Response time from data retrieval 2.2 seconds with an average execution time of Request data of 177 milli seconds.

Abstrak

Dalam berbagai bidang industri telah menggunakan serta menerapkan konsep layanan web yang menyediakan interoperabilitas pesan dan antarmuka pertukaran data. Salah satu kegunaan untuk berbagi data integrasi data terhadap sistem yang satu dengan sistem lain pada studi kasus ini dilakukan untuk mengelola data barang pasokan besi, di CV. XYZ Pratama Manufacture Engineering menggunakan metode penelitian pengumpulan struktur data, data desain arsitektur, membangun layanan web aplikasi, pembangunan Rest API, pengujian, dan pengukuran serta keamanan token JWT HS 256. Membangun integrasi dapat merumuskan bagaimana menggunakan data Barang CV. XYZ, menjadi sebuah sistem Inventory yang dibangun menggunakan teknologi Web Service dengan mengelola data API barang yang berjenis data Json pada Server Rest sehingga dapat digunakan oleh Client Rest. Penelitian ini menghasilkan Web Service melalui tahapan testing dengan service yang berbeda dengan total data yang direquest dan diterima yaitu 62.48 kb dan total waktu Response dari pengambilan data 2.2 detik dengan waktu rata-rata eksekusi Request data sebanyak 177 mili detik.

Article history

Received 06 Sep 2022
Revised 13 Oct 2022
Accepted 16 Oct 2022
Available 27 Oct 2022

Keywords

Web Service,
Data Integration,
Rest API,
JWT,
HS 256

Riwayat

Diterima 06 Sep 2022
Revisi 13 Okt 2022
Disetujui 16 Okt 2022
Terbit 27 Okt 2022

Kata Kunci

Web Service,
Integrasi Data,
Rest API,
JWT,
HS 256

PENDAHULUAN

Web Service teknologi interaksi antara Aplikasi yang satu dengan sistem lain telah digunakan di berbagai kebutuhan seperti pada sistem penjualan furniture (Maulidin et al., 2020) dalam pemanggilan data dengan *Json Web Token* dimana metode tersebut berguna untuk pengamanan data (Rahmatulloh et al., 2018), (Sriwijaya, 2021) dalam pemanggilan data menggunakan akses token mempunyai sumber yang dapat diakses melalui protokol alamat yang unik serta dapat melakukan

pengambilan alamat protokol HTTP dengan mudah menggunakan *Uniform Resource Locators* (Taufiqurrohman & Iwan Nurhidayat, 2016). Sedalam ini, *Web Service* banyak digunakan dari sisi pengembangan program *Back End*. *Web Service* berbasis *Json Web Token* sangat terkenal karena penggunaan dan penerapannya mudah digunakan, dan memiliki keamanan yang sangat baik sekali. Selain itu, arsitektur tersebut menjembatani pengiriman akses integrasi data menggunakan format data yaitu (*Javascript Object Notation*)

(Arianto, Mukhammad Agus Munir, Sirojul Khotimah, 2016) dari hasil riset ilmiah yang telah dilaksanakan pada aplikasi *mobile computing* bahwa kapasitas pesan pada Restfull API mencapai sembilan puluh kali lebih sedikit perbandingan ukuran pesan dari *Web Service* berbasis SOAP. Pada penelitian *Web Service* suku cadang mendapatkan hasil Test Measurement dengan waktu respon 2000 ms serta data yang diterima sebanyak 18.31 kb (Gunawan I., Witanti W., Renaldi F., 2021), setelah diteliti terdapat kerentanan bahwa Restfull API rendah dalam segi keamanannya untuk mengatasi masalah tersebut digunakan *Json Web Token* (JWT). Penerapan ini telah dicoba sebelumnya, yang menggunakan JWT dengan algoritma HS 256 yang masih umum digunakan, sehingga dapat menjadi kerentanan bagi keamanan Restfull API (Tanaem et al., 2016). *Json Web Token* telah banyak digunakan di berbagai macam, salah satunya dalam perusahaan manufaktur (Sabir et al., 2019) web token adalah cara ringkas dan aman untuk URL mewakili transfer antara dua pihak. Dalam JWT dikodekan sebagai *Json* yang digunakan sebagai muatan struktur *Json Web Signature* (JWS) atau sebagai teks biasa dari struktur *Json Web Encryption* (JWE), sekuritas pada *Json Web Token* menjadi sangat penting karena sekuritas data yang diintegrasikan dapat berupa data penting. Untuk mengatasi permasalahan tersebut, contohnya seperti data Client CV. XYZ untuk pertukaran data yang terintegrasi maka dibutuhkan teknologi *Web Service* dan pengamanan data API dengan *Json Web Token* berisi data barang, data stok barang pada perusahaan.

Penelitian yang telah dilakukan oleh (Danele et al., 2020) membahas tentang distribusi integrasi data manufaktur menggunakan cloud dimana data yang diintegrasikan dengan web perusahaan dengan web vendor kemudian data tersebut disimpan didalam cloud maka data yang terintegrasi menjadi lebih aman. Ketika suatu perusahaan tersebut meminta data akses dengan mudahnya sistem cloud ini dapat dikirim menurut (Munonye & Martinek, 2018) Rest API layanan web berbasis Java dan implementasi .Net dan menyimpulkan kinerjanya. Pada berbagai skenario pengujian, ditemukan bahwa layanan web berbasis Java berkinerja lebih baik untuk metode GET sekitar 80,36% dan .Net berkinerja lebih baik untuk metode

PUT dan kinerjanya sebanyak 11,6% lebih rendah.

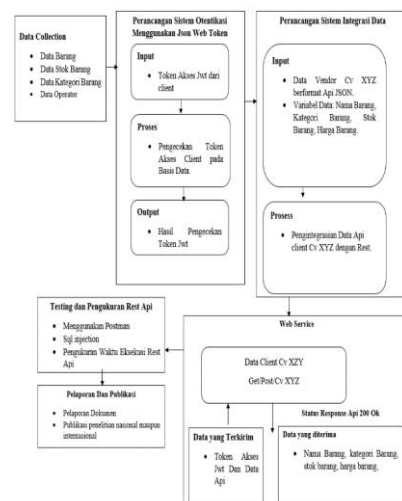
METODE PENELITIAN

Pada penelitian ini terdapat beberapa tahapan yang dikerjakan yaitu pengumpulan data collection client, penggunaan authorization menggunakan json web token, perancangan sistem integrasi data, dapat dilihat pada Gambar 1. Diagram alur penelitian perancangan web service, Testing dan Pengukuran Waktu Rest Api, pelaporan dan publikasi.

1. Tahap Data Collection

Pada tahap ini dilakukan pengambilan data yang diperlukan bagi peneliti. Data tersebut yaitu data vendor untuk digunakan sebagai integrasi data dari sistem inventory data yang diperlukan yaitu data barang tersebut berisikan nama barang, harga barang, kuantitas barang, data stock barang, data harga barang dan data foto, data operator serta data kategori barang merupakan data yang akan dikategorikan sebagai nama katagori yang didalamnya ada kategori rak, Besi 1, kabel, *background signal*, *bracket* dan bahan jadi mekanik.

Alur penelitian merupakan tahapan metode penelitian untuk penerapan penggunaan web service pada sistem inventory cv xyz dimana mulai dari tahap pengambilan data, pemasukan Token JWT HS 256 token tersebut di verivikasi apakah token tersebut berhasil atau tidak, integrasi data melalui komunikasi *Web Service* pada tahap berikutnya pengujian waktu *Rest Api* dengan tools postman.



Gambar 1. Diagram Alur Penelitian

2. Tahap Perancangan Sistem Otentikasi menggunakan Json Web Token

Pada tahap ini sistem inventory menggunakan *Json Web Token*. Otentikasi tersebut berisikan algoritma yang berfungsi sebagai penghubung untuk memvalidasi token akses dan menggunakan oleh client untuk mendapatkan data. *Json Web Token*. Token yang dihasilkan dari JWT dirancang dengan Rumus sebagai berikut dapat dilihat pada Formula JWT:

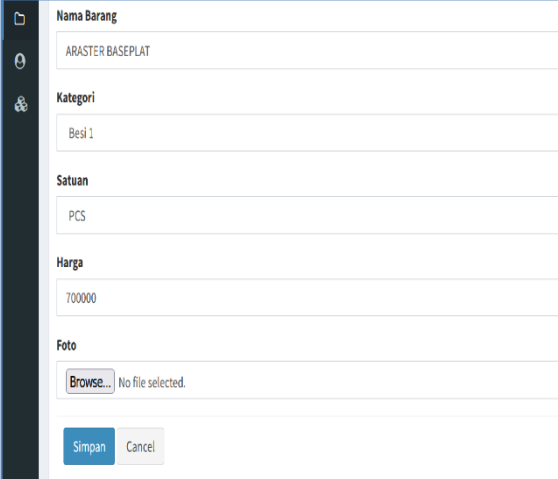
$$Token = f(Base64Encode) \sum_{n=a,\beta} (header.payload.signature)$$

Formula diatas menjelaskan tentang formula JWT yang berfungsi dari Base64 Encode dengan parameter header, payload dan signature. Token akses yang diperoleh dari server akan dimasukkan pada *header Hyper Text Transfer Protocol (HTTP)*. Sistem *authorization JWT* sama seperti penggunaan password pada login sistem ketika user memberikan token yang tersimpan didalam basis data lokal atau pada *Cookies Browser* (Gunawan & Rahmatulloh, 2019).

Token digunakan sebagai penghubung dan mejembatani antar sistem untuk mengakses halaman tertentu.

3. Tahap Perancangan Integrasi Data

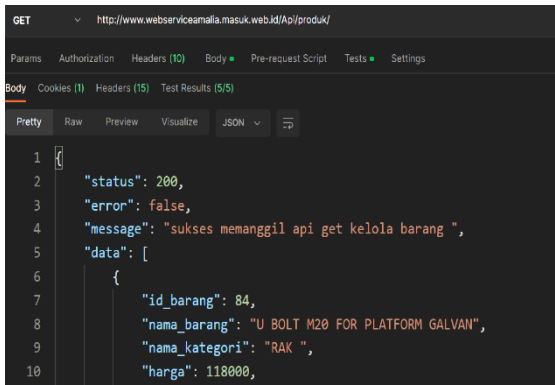
Integrasi data adalah proses penggabungan antara data yang berasal dari sumber dan sistem yang berbeda antar client dan vendor maupun dengan website. Pada tahap tersebut data diintegrasikan menggunakan algoritma untuk integrasi data. Data yang diperoleh pada proses pertukaran data data Amalia yang berisi data berformatkan json Nama barang, Harga Barang, Stock Barang, Foto Barang. Data tersebut adalah data yang terdapat pada CV. XYZ Tahapan pertukaran data dimulai dengan memasukkan data manufaktur yang awalnya terpisah sebagai input pada sistemnya nantinya. Pada tahap selanjutnya proses data yang terpisah antara data vendor digabungkan menjadi satu kesatuan. Hasil dari proses ini adalah data manufaktur yang telah terintegrasi layanan service produk dapat dilihat pada Gambar 3 serta perancangan program integrasi data dapat dilihat pada Tabel 1.



Gambar 2. Input Data Integrasi Barang

pada gambar 2 menjelaskan proses memasukan data barang kepada sistem inventory yang akan diintegrasikan tahap pertama memasukan nama barang *araster baseplat*, berkategori Besi 1, dengan harga 70000 pcs(*pieces*) kemudian dengan klik tombol simpan data barang akan tersimpan kepada sistem. Pengiriman data barang pada sistem inventory sesuai kebutuhan dari data yang akan diintegrasikan, dan data yang akan dimasukkan melalui layanan *Web Service* untuk pertukaran layanan yang nantinya akan dikirimkan dengan tools postman sebagai analisa bahwa layanan *Web Service* berstatus 200 yang artinya service tersebut berhasil dihubungkan dengan penggunaan Rest Api

pada gambar 3 adalah tahap pengujian API dengan postman pertama user memasukan url Api/produk kepada *form* yang disediakan oleh postman kemudian user mengirim url dengan memijit tombol *send* menggunakan *method Get* mengambil data dari pengiriman client *Web Service* terhadap server dan mengecek APIproduk apakah data tersebut sukses memanggil API atau tidak jika tidak maka *response* yang diberikan akan berstatuskan 404 atau *response not found*.



Gambar 3. Script Data Barang

Tabel 1. Perancangan Integrasi Data

| No | Program Integrasi Data |
|----|---|
| 1 | Kamus data inventory: array of string; integrateInventory: array of string; i: int; Algoritma {load data yang akan diintegrasikan} Cv. XYZ <- [read('inventory.json'),16 read('vendor.json'),]; {gabungkan semua data Web Service ke array baru} FOR i <- 1 to webserviceData.length DO integratedCvAmalia.push(webserviceDataCvXYZ[i]); ENDFOR Output(integrated); |

4. Perancangan Web Service

Web Service dibuat agar client dapat mendapatkan layanan service. Web Service ini berguna untuk jalur komunikasi antara client dengan Web Service agar client dapat mendapatkan data Vendor yang akan dimintanya oleh sistem tersebut. Web Service berfungsi sebagai tempat untuk authorization Json Web Token yang akan mengakses dari pengintegrasian data yang diterima melalui Web Service yaitu id_barang, nama_barang, nama_kategori, harga barang, nama_satuan, foto barang Perancangan Endpoint API dapat dilihat pada tabel 2. (Hu et al., 2017)

Tabel 1. Kelola Endpoint API

| No | Url api | Method | Param eter | Deskripsi |
|----|---------|--------|------------|-----------|
|----|---------|--------|------------|-----------|

| | | | | |
|---|----------------------------------|------|-----------------|--------------------------|
| 1 | /KelolaUser | Get | Token JWT | Mengambil Token |
| 2 | {Api/operato/} /Token/Regist | Post | Kode_Registrasi | Regist |
| 3 | {Token/Regist} /Token/Verifikasi | Post | Kode_Verifikasi | Untuk Authorization |
| 4 | /KelolaStok/api {Api/stok} | Get | Kode_Stok | Meminta data Stok Barang |
| 5 | /KelolaProduk/ {Api/produk} | Get | Kode_Produk | Mengambil data barang |
| 6 | /KelolaApiKategori/ | Get | Kode_Kategori | Mengambil data kategori |

5. Pembangunan Perangkat Lunak



Gambar 4 Arsitektur Perancangan Web Service

Sistem inventory CV. XYZ berfungsi sebagai client berfungsi data yang akan didapatkan. Perangkat lunak ini adalah prototipe yang dirancang untuk mengimplementasikan fungsi pada Web Service. Perangkat lunak dibuat untuk simulasi dimana perancangan perangkat lunak lain dapat menggunakan data Manufaktur untuk digunakan pada aplikasi yang mereka buat.

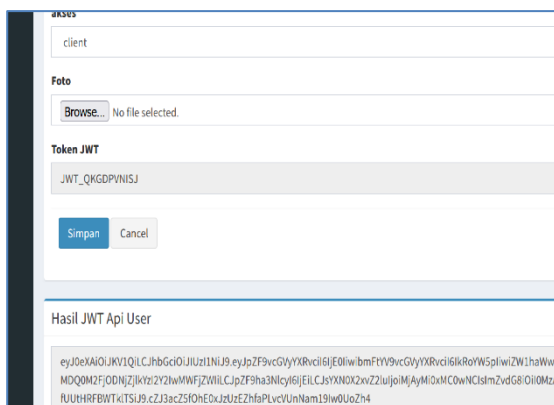
Perangkat lunak hanya memiliki fitur untuk mengirim *request data Api* dan menerima *response data API* dari *Web Service* dan menampilkan data yang diterimanya oleh sistem. (Soni & Ranga, 2019) tahap pertama client akan mengirim request pesan data layanan terhadap *Web Service* kemudian server akan memberikan status response berupa hasil pengiriman API jika response API berhasil, pertukaran data yang baik akan menghasilkan status code 200 jika gagal akan memberikan response API 400 permintaan tidak dapat dipahami oleh server karena sintaks yang salah arsitektur perancangan web service dapat dilihat pada gambar 5.

6. Testing dan Pengukuran Waktu Rest Api

Pengujian testing Rest Api dengan mendapatkan request data dari Server sistem Inventory dengan menguji sebanyak 5 kali iterasi dengan method Get data dari Api produk, Api operator, Api Post Registrasi Token JWT, serta verifikasi token JWT untuk Measurement menguji Eksekusi Waktu response Rest Api.

HASIL DAN PEMBAHASAN

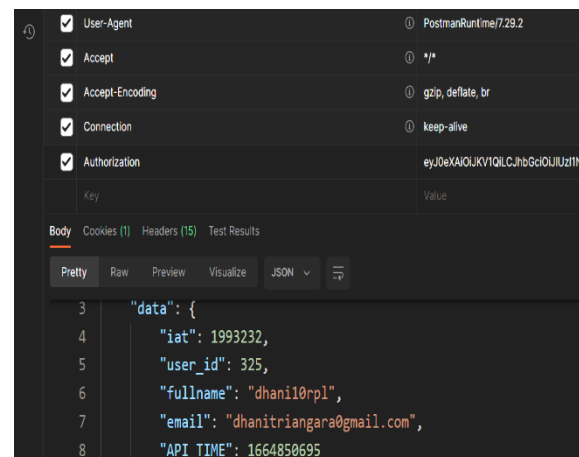
Penelitian ini telah diterapkan bahwa layanan *web service* yang diimplementasikan terhadap sistem inventory CV. XYZ Tahap Implementasi terdiri dari 2 fase Penerapan pertama sebagai Authorization saat masuk sistem dan implementasi pada tahap kedua adalah verifikasi token SHA 256 menggunakan algoritma baseurl64.



Gambar 5. Registrasi Token

Hasil yang didapatkan yaitu Token Json Web Token pada penelitian ini menggunakan HS 256 dimana fungsi hash tersebut menerjemahkan kode ke pesan asli yang nantinya akan diamankan oleh algoritma baseurl64 (Minarni, 2019) menghasilkan berupa data Api, HS 256 dapat dilihat pada gambar 5. menunjukkan proses pengambilan data dengan Token Api user.

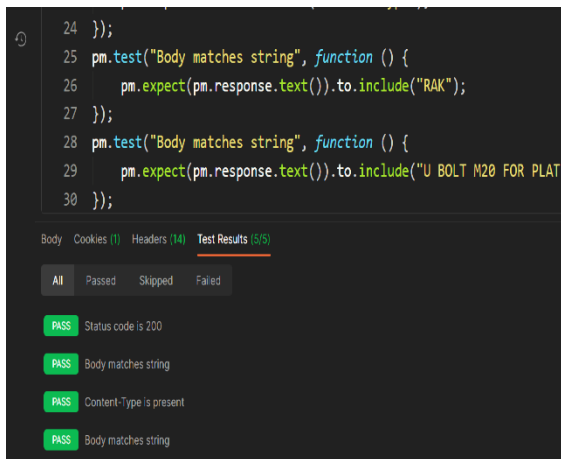
Hasil Verifikasi Token JWT menghasilkan data hash HS 256 yang di Encoded menggunakan baseurl64 yang dimasukkan pada Bearer token yang terdapat pada header sebagai Authorization didalam Api tersebut terdapat data *user_id* sebagai id user, *fullname* *dhani10rpl*, *email* serta untuk *Api_Time* adalah waktu expired saat pengambilan data token ketika waktu tersebut habis maka token JWT akan generate *Authorization* baru pada saat menggunakan token HS 256 tersebut. dan menampilkan informasi token Expired Gambar verifikasi Token dapat dilihat pada gambar 6.



Gambar 6. Verifikasi Token

Hasil pengujian Integrasi data produk Dari hasil pengujian terdapat response time sebanyak 44 ms dan untuk pengujian request get data Rest API produk sebanyak 1 kali iterasi eksekusi request data serta menguji test status code get API produk bersarkan *id_barang* yaitu mempunyai status code 200 yang artinya pengambilan data berhasil didapatkan. telah dilakukannya pengujian string berdasarkan body yang terdapat pada Rest Api Produk matches string bahwa hasil pengujian tersebut menghasilkan Kode PASS adalah pelaporan pengujian *response* yang diujikan melalui unit test berfungsi

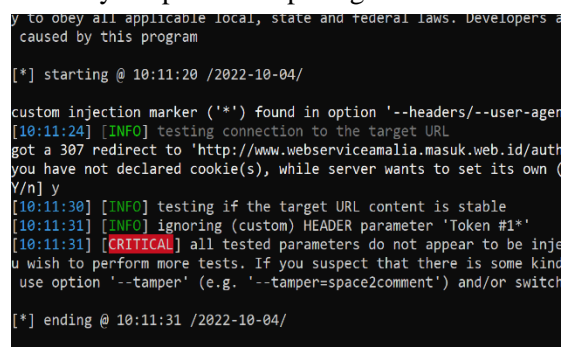
memberikan informasi kepada selaku pengujian API bahwa unit *test* yang diujikan *success*. Pengujian unit test dapat dilihat pada gambar 7.



Gambar 7. Pengujian Integrasi data

Open Application Security Project (OWASP) Menyatakan bahwa serangan injeksi adalah serangan standar pada keamanan pada aplikasi web. Penyerangan data dapat dikelabui dengan menyisipkan kode yang sangat berbahaya terhadap aplikasi web berfungsi untuk mengeksekusi perintah yang digunakan agar bisa digunakan untuk mendapatkan hak akses yang valid (Dawson et al., 2019). Injeksi penyerangan terjadi ketika penyerang mengeksploitasi kode yang menjadi rentan dengan menyisipkan kode yang sangat berbahaya, ke dalam aplikasi Web.

Dalam melakukan pelaksanaan pengujian *sql Injection* ini penulis menggunakan perangkat lunak *tools* bernama *SQLMap*. Pertama menentukan target uji dengan memasukan *endpoint* API dari hasil penerapan *Web Service* yang dibangun dengan memasukan parameter token JWT melalui *header* untuk menginjeksi kode yang berbahaya dapat dilihat pada gambar 8.



Gambar 8. Pengujian Sql Injection

Menurut penelitian sebelumnya mempunyai kekurangan pada token akses Restful Api tidak adanya response waktu dari Request Token Rest Api (Suryanto et al., n.d.), untuk meneruskan pada penelitian sebelumnya Dari hasil pengujian Execution Measurement Result Rest Api get data barang, Post Registrasi JWT Mendapatkan Token JWT, serta Get Data Rest API pengujian waktu eksekusi yang akan dilakukan menggunakan *newman* adalah koleksi test json untuk menjalankan *automation* API di postman pada pengujian waktu koleksi tes mendapatkan total durasi waktu 2.2 *second* dengan total data 62.48 kb average 177 ms, dengan waktu minimal 43 ms, serta waktu maximal 121 ms, 24 ms. Dapat dilihat pada gambar 9.

| | executed | failed |
|---|----------|--------|
| iterations | 1 | 0 |
| requests | 6 | 0 |
| test-scripts | 2 | 0 |
| prerequest-scripts | 0 | 0 |
| assertions | 7 | 0 |
| total run duration: 2.2s | | |
| total data received: 62.48kB (approx) | | |
| average response time: 177ms [min: 43ms, max: 349ms, s.d.: 121ms] | | |

Gambar 9. Pengujian Waktu Eksekusi Rest Api

Dijelaskan pengujian unit testing waktu eksekusi menggunakan *newman* sebanyak 1 iterasi, *request* 6 data layanan service dikirim oleh *client* melalui *postman*, untuk *test-script* sebanyak 6 kali eksekusi, *prerequest-script* 0 eksekusi sedangkan test *assertions* pada kode Rest Api sebanyak 7 penyisipan tes sehingga memperoleh total durasi waktu sebanyak 2.2 *second*.

Berikut adalah hasil waktu respon sistem pengujian pengukuran menggunakan metode Restful Api menggunakan *Newman*

Tabel 2. Waktu Respons Hasil Pengukuran

| N | Total Data Eksekusi | Metode | Url | Waktu Respon s |
|----|---------------------|--------|----------------|----------------|
| 1. | 8 | GET | Api/p roduk | 109 ms |
| 2. | 1 | GET | | 106 ms |

| | | | | |
|----|---|------|--------------------------|--------|
| | | | Api/o perato r | |
| 3. | 1 | POST | Api/r egistr asi | 109 ms |
| 4. | 1 | GET | Api/st ok | 114 ms |
| 5. | 1 | POST | Toke n/Ver ifikasi | 109 ms |
| 6. | 1 | GET | Api/K ategor i | 445 ms |

KESIMPULAN

Berdasarkan penelitian dan analisa yang telah dilakukan maka dapat disimpulkan bahwa penelitian *Web Service* dalam pengintegrasian data Rest Api dari client menuju vendor terbukti berjalan dengan baik untuk panjang yang dihasilkan dari setiap token json web token HS 256 Memperoleh sebanyak 256 bit untuk penggunaan pengamanan Rest Api dengan Token JWT HS 256 menggunakan algoritma baseurl 64 berjalan dengan baik untuk *Web Service* pada penelitian ini mampu mengamankan data client CV. XYZ dan vendor dari pihak orang yang akan mencuri data. Penulis telah melakukan pengujian get data Rest Api produk menggunakan tools postman untuk merequest dan melihat data rest Api yang dintegrasikan pengujian dengan postman serta penulis juga telah menguji Waktu Response Time dari Testing Measurement Test terdapat waktu response sebesar 2.2 detik, untuk total data dari hasil eksekusi request data sebesar 62.48 kb dengan rata-rata waktu 177 ms setiap 1 get data Api Saran untuk penelitian selanjutnya pengamanan jenis Hash JWT yang berbeda. Serta pengujian integrasi data menggunakan method web service yang lain seperti method PUT dan Delete Api.

REFERENSI

Arianto, Mukhammad Agus; Munir, Sirojul; Khotimah, K. (2016). Analisis dan Perancangan Representational State Transfer

(REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework. *Jurnal Teknologi Terpadu*, 2(2), 1–8.

Danele, S., Stefano, D. P., Davide, S., Andrea, C., & Roberto, A. (2020). *Dynamic integration of distributed, Cloud-based HPC and HTC resources using JSON Web Tokens and the INDIGO IAM Service*. 07020.

Dawson, M., Taveras, P., & Taylor, D. (2019). Applying Software Assurance and Cybersecurity NICE Job Tasks through Secure Software Engineering Labs. *Procedia Computer Science*, 164, 301–312. <https://doi.org/10.1016/j.procs.2019.12.187>

Gunawan, R., & Rahmatulloh, A. (2019). JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 5(1), 74. <https://doi.org/10.26418/jp.v5i1.27232>

Hu, W., Huang, Y., Liu, X., & Xu, C. (2017). Study on REST API Test Model Supporting Web Service Integration. *Proceedings - 3rd IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2017, 3rd IEEE International Conference on High Performance and Smart Computing, HPSC 2017 and 2nd IEEE International Conference on intelligent data and securit* 133-138. <https://doi.org/10.1109/BigDataSecurity.2017.35>

Kornienko, D. v, Mishina, S. v, Setiawan, E. B., Setiyadi, A., Kornienko, D. v, & Mishina, S. v. (2021). *Integration of Supply Management System in*

- Auto Parts Company Using Web Services Integration of Supply Management System in Auto Parts Company Using Web Services.* <https://doi.org/10.1088/1742-6596/1845/1/012022>
- Maulidin, A. J., Renaldi, F., & Umbara, F. R. (2020). Online Integration of SQL and No-SQL Databases using RestAPIs: A Case on 2 furniture e-Commerce Sites. *2020 3rd International Conference on Computer and Informatics Engineering, IC2IE 2020*, 261–266. <https://doi.org/10.1109/IC2IE50715.2020.9274613>
- Minarni, R. (2019). Implementasi Algoritma Base64 untuk Mengamankan SMS pada Smartphone. *Technology and Science (BITS)*, 1(1), 28–33.
- Munonye, K., & Martinek, P. (2018). Performance Analysis of the Microsoft. Net- and Java-Based Implementation of REST Web Services. *SISY 2018 - IEEE 16th International Symposium on Intelligent Systems and Informatics, Proceedings*, 191–196. <https://doi.org/10.1109/SISY.2018.8524705>
- Rahmatulloh, A., Sulastri, H., & Nugroho, R. (2018). Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(2). <https://doi.org/10.22146/jnteti.v7i2.417>
- Sabir, B. E., Youssfi, M., Bouattane, O., & Allali, H. (2019). Authentication and load balancing scheme based on JSON Token for Multi-Agent Systems. *Procedia Computer Science*, 148, 562–570. <https://doi.org/10.1016/j.procs.2019.01.029>
- Soni, A., & Ranga, V. (2019). API features individualizing of web services: REST and SOAP. *International Journal of Innovative Technology and Exploring Engineering*, 8(9 Special Issue), 664–671. <https://doi.org/10.35940/ijitee.I1107.0789S19>
- Sriwijaya, U. (2021). Sistem Keamanan Web Service (*RESTful API*) Pada *Json Web Token* untuk mengukur token dan Authorization Dengan hashing algoritma RSA-SHA-512 Sistem keamanan Web Service (*RESTful API*).
- Suryanto, H., Renaldi, F., & Santikarama, I. (n.d.). *Implementing Secure Rest API on The Integration of Electronic Medical Records Between A Local Hospital and Nearby Private Clinics.*
- Tanaem, P. F., Manongga, D., & Irian, A. (2016). RESTful Web Service Untuk Sistem Pencatatan. *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(April), 2443–2229.
- Taufiqurrohman, & Iwan Nurhidayat, A. (2016). Rancang Bangun Sistem Integrasi Antara Desktop Dan Web Aplikasi di Toko Parfum Confi. *Jurnal Manajemen Informatika*, 5(2), 37–41.