Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 6 (2020) 1036 – 1045 ISSN Media Elektronik: 2580-0760

Implementasi *JSON Web Token* Berbasis Algoritma *SHA-512* untuk Otentikasi Aplikasi BatikKita

Andi Setiawan[1], Ade Irma Purnamasari[2]
[1,2]Teknik Informatika, STMIK IKMI Cirebon
[1]42andisetiawan@gmail.com*, [2]irma2974@yahoo.com

Abstract

BatikKita is an application built using an android platform and a web framework, with a web service architecture as communication between the two BatikKita applications with different platforms, while for authentication between the two BatikKita application platforms, it uses the concept of asymmetric cryptography with the Keyed-Hash Message Authentication Code encryption method (HMAC) the first generation which is the weakness of the BatikKita application. The purpose of this study is to implement the JSON Web Token (JWT) with the HMAC SHA-512 algorithm based on a web service architecture with the type of RESTful web service in the BatikKita application. The algorithm used in this research is JSON Web Token (JWT) with HMAC SHA-512, while the software development model used is Rapid Application Development. Tests were carried out on two web service architectures, namely Simple Object Access Protocol (SOAP) and Representational State Transfer (REST), using JSON Web Token (JWT) with the HMAC SHA-512 algorithm as authentication in the BatikKita application which is built on the android and web platforms. framework. As a comparison in testing this research is the use of JSON Web Token (JWT) with the HMAC SHA-256 algorithm and JSON Web Token (JWT) with the HMAC SHA-384 algorithm against two web service architectures, namely Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The results obtained from testing the implementation of JSON Web Token with the SHA-512 algorithm in the BatikKita application are for an average speed increase between 138.8 milliseconds for SOAP and 122.7 milliseconds for REST compared to the HMAC SHA-256 algorithm and the HMAC SHA-algorithm. 384. While the required token size is greater between 2.13 kb for SOAP and 2.11 kb for REST compared to the HMAC SHA-256 algorithm and the HMAC SHA-384 algorithm.

Keywords: JSON Web Token, HMAC SHA-512, SOAP Web Service, REST Web Service, Rapid Application Development.

Abstrak

BatikKita adalah aplikasi yang dibangun menggunakan platform android dan web framework, dengan arsitektur *web service* sebagai komunikasi diantara kedua aplikasi BatikKita yang berbeda platform, sedangkan untuk otentikasi diantara kedua platform aplikasi BatikKita, menggunakan konsep kriptografi asimetris dengan metode enkripsi *Keyed-Hash Message Authentication Code* (*HMAC*) generasi pertama yang menjadi kelemahan dari aplikasi BatikKita. Tujuan dari penelitian ini adalah untuk mengimplementasikan *JSON Web Token (JWT)* dengan algoritma *HMAC SHA-512* berdasarkan arsitektur *web service* dengan jenis *RESTful web service* pada aplikasi BatikKita. Algoritma yang digunakan dalam penelitian ini adalah *JSON Web Token (JWT)* dengan *HMAC SHA-512*, sedangkan model pengembangan perangkat lunak yang digunakan adalah *Rapid Application Development*. Pengujian dilakukan terhadap dua buah arsitektur *web service*, yaitu *Simple Object Access Protocol* (*SOAP*) dan R*epresentational State Transfer* (*REST*), menggunakan *JSON Web Token (JWT)* dengan algoritma *HMAC SHA-512* sebagai otentikasi dalam aplikasi BatikKita yang dibangun berdasarkan platform android dan web framework. Sebagai perbandingan dalam pengujian dipenelitian ini adalah penggunaan *JSON Web Token (JWT)* dengan algoritma *HMAC SHA-256* dan *JSON Web Token (JWT)* dengan algoritma *HMAC SHA-384* terhadap dua buah arsitektur *web service*, yaitu *Simple Object Access Protocol* (*SOAP*) dan R*epresentational State Transfer* (*REST*). Hasil yang diperoleh dari pengujian implementasi *JSON Web Token* dengan algoritma *SHA-512* pada aplikasi BatikKita adalah untuk kecepatan meningkat rata-rata antara **138,8 milisecond** untuk *SOAP* dan **122,7 milisecond** untuk *REST* dibandingkan dengan algoritma *HMAC SHA-256* dan algoritma *HMAC SHA-384*. Sedangkan ukuran token yang dibutuhkan lebih besar antara **2,13 kb** untuk *SOAP* dan **2,11 kb** untuk *REST* dibandingkan dengan algoritma *HMAC SHA-256* dan algoritma *HMAC SHA-384*.

Kata kunci: *JSON Web Token, HMAC SHA-512, SOAP Web Service, REST Web Service, Rapid Application Development.*

Diterima Redaksi : 04-10-2020 | Selesai Revisi : 20-11-2020 | Diterbitkan Online : 20-12-2020

1. Pendahuluan

BatikKita adalah aplikasi yang dikembangkan dengan menggunakan platform Android sebagai *design frontend* dan Web Framework sebagai *design backend*. Untuk mekanisme pertukaran data dari aplikasi BatikKita, menggunakan *Restful Web Service*, *library retrofit* dan *library volley* sebagai *web application programming interface* (API) yang memungkinkan kedua aplikasi yang berbeda platform dapat saling berkomunikasi menggunakan sumber data yang sama termasuk pada proses otentikasi. Kelemahan dari aplikasi BatikKita adalah penggunaan kriptografi asimetris dengan metode enkripsi *Keyed-Hash Message Authentication Code* (*HMAC*) generasi pertama sebagai proses otentikasi datanya dan tidak menggunakan token dalam proses verifikasinya. Permasalahan yang diangkat dalam penelitian ini adalah masalah kecepatan pada saat proses otentikasi data dibagian *frontend* sering dikeluhkan oleh pengguna aplikasi BatikKita terutama penjual dan pembeli batik Trusmi Cirebon.

Sedangkan tujuan dari penelitian ini adalah untuk mengimplementasikan *JSON Web Token* (JWT) dengan algoritma *Keyed-Hash Message Authentication Code* (*HMAC*) SHA-512 pada aplikasi BatikKita yang berbasis arsitektur *Restful Web Service* (*RWS)*.

Dari beberapa literatur menyebutkan bahwa permasalah-an keamanan merupakan poin penting ketika aplikasi BatikKita diimplementasikan pada sebuah web server, hal ini disebabkan karena aplikasi yang dibangun melalui platform web service, memiliki kerentanan tertinggi terhadap keamanan dan minim perlindungan ketika diimplementasikan pada sebuah *Web Server* [1]. Selain rentan terhadap masalah keamanan, web service juga memiliki permasalahan keamanan pada saat otentifikasi, hal ini disebabkan akibat arsitektur dari web service menggunakan *Representational State Transfer* (REST) dan dijalankan melalui *Hypertext Transfer Protocol* (HTTP) yang sangat membutuhkan *JSON Web Token* (JWT) pada *Web Service* dan *Backend System* BatikKita [2]. Implementasi JSON Web Token pada aplikasi BatikKita berbasis RESTful API akan meningkatkan keamanan, hal ini disebabkan karena pada saat otentifikasi maka aplikasi akan sulit diakses tanpa adanya token [3]. Merujuk dari penelitian sebelumnya, penelitian ini menerapkan algoritma *SHA-512* yang diimplementasikan pada SOAP dan RESTful untuk memperoleh kecepatan token dan kecepatan pada saat otentikasi [4]. Otentikasi token atau otentikasi berbasis token memiliki keunggulan dibandingkan dengan otentikasi secara tradisional karena tidak memiliki batasan tertentu (*stateless*) yang memungkin-kan untuk mengganti peranan *cookie* dan *session* pada otentikasi secara tradisional sekaligus mengurangi beban kerja dari server, dengan menggantikan peranan *session* pada layanan web service secara tradisional [5]. Implementasi *JSON Web Token* (JWT) juga dapat digunakan untuk pengamanan komunikasi antara agen pada suatu perusahaan berdasarkan algoritma yang terenkripsi secara asimetris [6]. Kemudian JSON Web Token dapat memberikan kinerja yang sangat signifikan dalam sebuah sistem terdistribusi berskala besar dan terdesentralisasi [7]. Disamping berbagai keunggulan yang dikemukan sebelumnya, keunggulan lainnya dalam penyimpanan kedalam database. File JSON terdiri dari data JSON dan data biner, sehingga pada saat pemisahan data biner akan disimpan sebagai gambar dan data semi terstruktur diubah menjadi JSON dokumen [8]. Selain penerapannya pada web service JSON Web Token juga dapat diimplementasikan pada MQTT dan NodeMCU, karena JSON Web Token dapat melakukan otentikasi pada token yang telah kadaluarsa [9].

Sedangkan BatikKita merupakan aplikasi berbasis *restful web service (RWS)*, yang menerapkan *JSON Web Token* (*JWT*) dengan arsitektur algoritma *SHA-512* untuk otentikasi penggunanya. Pada sebuah percobaan pembentukan DNA buatan, algoritma SHA-512 diguna-kan untuk mengotentikasi gambar, data, teks, dan video selama pengujian biologi ilustrasi pembentukan DNA buatan [10]. Algoritma SHA-512 cukup handal dan aman ketika mengenkripsi gambar, sensitivitas gambar, dan tahan terhadap serangan kriptanalitik [11]. *Secure Hash Algorithm* (*SHA-1*) bahkan digunakan untuk mengenkripsi tanda tangan digital *Digital Signature Algorithm* (DSA) [12]. Selain digunakan untuk mengenkripsi tanda tangan digital atau DSA, *Secure Hash Algoritma* (SHA-3) dapat digunakan untuk arsitektur perangkat keras yaitu processor Nios II pada FPGA Arria 10 GX [13]. *Secure Hash Algorithm* (*SHA*) digunakan pula struktur dua fungsi hash dengan *Chaotic Neural Network* (*CNN*) [14]. Selain itu fungsi dari *secure hash* digunakan untuk mensimulasikan serangan secara kriptanalitik terhadap MNF-256 [15]. Secure Hash juga digunakan untuk mengotentikasi gambar menggunakan metode pengenalan iris berbasis *Low Density Parity Check* (LDPC) dan *Secure Hash Algorithm* (SHA) dengan *watermarking scheme* yang dapat dibalik [16].

Untuk metode pengembangan perangkat lunak yang digunakan dalam merancang aplikasi BatikKita berbasis android dan web framework laravel adalah *Rapid Application Development* (*RAD*). *Rapid Applica-tion Development* (*RAD*), memiliki tahapan yang paling singkat dibandingkan dengan metode pengembangan perangkat lunak lainnya, terdiri dari tiga tahap diantaranya adalah *requirement planning*, *design and testing system*, dan *implementation*. Hal ini dibutuhkan untuk mempersingkat tahap pengembangan aplikasi BatikKita [17]–[21].

Kemudian dalam implementasinya aplikasi BatikKita.-online menerapkan web service berbasis framework model, view, dan controller (MVC) untuk memudahkan pembuatan aplikasi BatikKita pada saat pengkode-an aplikasi. Model, View, dan Controller termasuk arsitektur berbasis web service untuk menghasilkan

aplikasi secara modular, yang dapat mengurangi kompleksitas saat mendesign aplikasi BatikKita, meningkatkan fleksibilitas aplikasi BatikKita, dan memiliki kemampuan untuk penggunaan kembali atau *reusability* terutama pada tahap pengembangan aplikasi BatikKita [22]–[30]. Sehingga penerapan model, view, dan controller (MVC) di aplikasi BatikKita sangat tepat.

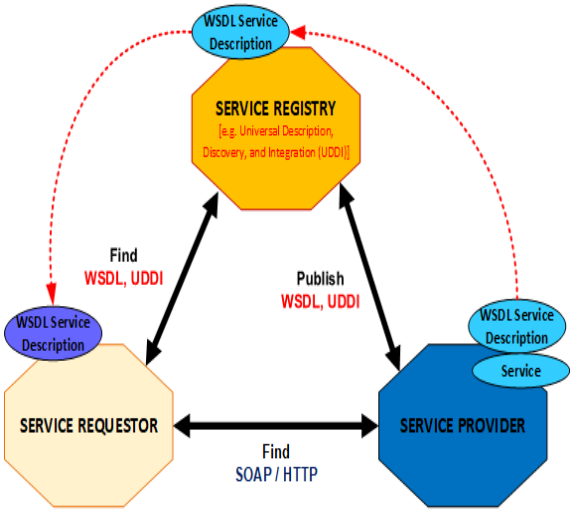
Sedangkan alasan dilakukannya penelitian ini adalah untuk memudahkan transaksi antara penjual atau pengrajin batik Trusmi kabupaten Cirebon dengan pembeli terutama pada saat pandemi saat ini yang membatasi transaksi pembelian secara langsung, sekaligus membantu pengrajin batik Trusmi untuk memasarkan produksinya kepada konsumen secara langsung, karena aplikasi BatikKita menggunakan konsep bussiness to customer (B2C) dalam implementasinya [31]–[37].

Berikut pembahasan dari penelitian ini mengenai teknis untuk menghubungkan antara aplikasi BatikKita dengan platform android dan web framework melalui arsitektur *restfull web service* dengan *JSON Web Token* (JWT) dengan algoritma *HMAC SHA-512* untuk otentikasi datanya.

2. Metode Penelitian

2.1 Web Service

Layanan web service merupakan sebuah sistem yang didesign untuk mendukung interaksi antara aplikasi BatikKita dengan sebuah jaringan. Layanan web service memberikan layanan berupa informasi antar sistem sehingga antara bagian didalam sistem layanan web service dapat saling berinteraksi. Layanan web service dapat juga diartikan sebagai antarmuka yang mengillustrasikan beberapa operasi yang dapat diakses melalui jaringan [4]. Contoh arsitektur layanan web termasuk Simple Object Access Protokol (SOAP) dan Representational State Transfer (REST) terutama pada aplikasi BatikKita. Arsitektur web service aplikasi BatikKita dapat dilihat pada gambar 1 berikut ini.



Gambar 1. Arsitektur Web Service

2.2 *Representation State Transfer* (REST)

REST bukan sebuah bahasa pemrograman, melainkan arsitektur dari *web service* yang diturunkan dari berbagai gaya arsitektur *hybrid* berbasis jaringan yang sering diterapkan dalam layanan berbasis web. REST sebagai salah satu dari arsitektur *web service*, secara umum dijalankan melalui *Hypertext Transfer Protocol* (HTTP) yang melibatkan pembacaan sebuah halaman berbasis web yang berisi file XML atau JSON. *Application Programming Interface* (API) yang mengikuti pola dari REST dinamakan RESTful API, menggunakan *Uniform Resource Inditifiers* (URI) untuk penggunaan sumber daya *web service*. Metode *GET* digunakan untuk mendapatkan sumber daya dan metode *POST* digunakan untuk membuat sumber daya baru, kemudian metode *PUT* digunakan untuk memperbaha-rui sumber daya, sedangkan metode *DELETE* digunakan untuk menghapus sumber daya atau kumpulan sumber daya, seperti yang diilustrasikan pada tabel berikut [5].

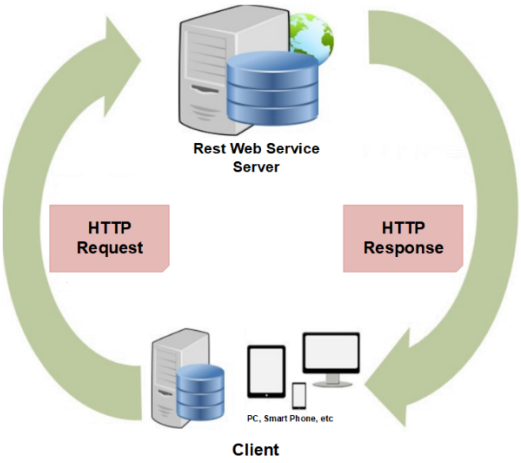
Tabel 1. Contoh RESTful API

Resource	Method			
	Get	Post	Put	Delete
/api/ customer	Get a list of all customer	Create a new list of customer, Treat as a collection.	Update a list of customer	Delete all customer
/api/ customer/ 123	Get a customer by customer's ID	Create a new customer in it.	If a customer exists, update the customer. If a customer not exits. Create a new customer	Delete the customer

Batasan dari konsep REST [4], diantaranya adalah :

a) *Resource Identification*, artinya web bergantung pada *Uniform Resource Identifier* (URI) untuk mengidentifikasi sumber daya.
b) *Connectedness*, artinya clien dari *RESTFul Service* harus mengetahui tautan untuk menemukan sumber daya agar dapat berinteraksi dengan layanan.
c) *Uniform Interface*, artinya sumber daya harus tersedia melalui antarmuka yang seragam dengan semantik yang mendefinisikan interaksi.
d) *Self-Describing Messages*, artinya *web service* mengekspos sumber daya yang ada, RESTFul menggunakan lebih dari satu format data (XML, JSON, RDF, dll.) dibandingkan dengan SOAP (XML).
e) *Stateless Interactions*, artinya setiap permintaan dari clien sudah lengkap dan memenuhi kebutuhan akan permintaan.

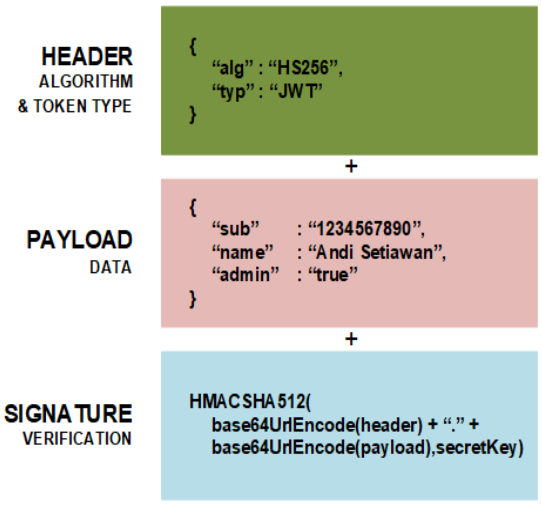
Siklus *Representation State Transfer* (REST), dimulai dari *request* kemudian dilanjutkan dengan *response* digambarkan pada gambar berikut ini.



Gambar 2. Siklus Representation State Transfer (REST)

2.3 *JSON Web Token* (JWT)

JSON Web Token (JWT) merupakan token yang berbentuk String JSON yang padat dari sisi ukuran, informasi bersifat mandiri yang dapat digunakan untuk melakukan otentikasi dan pertukaran informasi pada aplikasi BatikKita. Terdiri dari tiga bagian utama yang dipisahkan oleh titik-titik (".") yaitu header, payload dan signature [4]. Secara teknis, cara kerja JWT sangat mirip ketika pengguna mengisi data berupa password. Apabila pengguna berhasil login, server memberikan token yang tersimpan didalam *cookies browser* atau *local storage*. Fungsi dari token pada aplikasi BatikKita untuk otentikasi dan pertukaran informasi, kemudian pengguna akan mengirim balik token tersebut sekaligus membuktikan bahwa pengguna sudah berhasil login pada aplikasi BatikKita [5]. Penggabungan bagian utama yang terdiri dari *header*, *payload* dan *signature* untuk menghasilkan *JSON Web Token* digambarkan pada gambar 2 berikut ini.



Gambar 3. Siklus *JSON Web Token* (JWT)

Pada bagian Header dari contoh pada gambar 3, terdiri dari dua bagian utama yaitu algoritma dan jenis token yang digunakan, kemudian JSON dikodekan berdasarkan Base64. Bagian payload berisi merupakan tambahan data dan data pengguna yang muatannya sama dengan bagian header yang dienkode dengan Base64. Sedangkan bagian ketiga adalah signature yang sudah disandikan dan telah ditentukan header dan tandanya [5]. Algoritma kriptografi yang biasa digunakan oleh JSON Web Token (JWT) adalah sebagai berikut.

Tabel 2. Algoritma JSON Web Token (JWT) [5].

"alg" parameter value	Digital Signature or MAC Algorithm	Implementation Requirements
HS256	HMAC using SHA-256	Required
HS384	HMAC using SHA-384	Optional
HS512	HMAC using SHA-512	Optional
RS256	RSASSA-PKCS1-v1_5 using SHA-256	Recommended
RS384	RSASSA-PKCS1-v1_5 using SHA-384	Optional
RS512	RSASSA-PKCS1-v1_5 using SHA-512	Optional
ES256	ECDSA using P-256 and SHA-256	Recommended
ES384	ECDSA using 384 and SHA-384	Optional
ES512	ECDSA using P-512 and SHA-512	Optional
PS256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256	Optional
PS384	RSASSA-PSS using SHA-384 and MGF1 with SHA-384	Optional
PS512	RSASSA-PSS using SHA-512 and MGF1 with SHA-512	Optional
none	No digital signature or MAC performed	Optional

Sedangkan untuk rumus *JSON Web Token* (JWT), diformulasikan pada formula berikut ini [5].

$$\boldsymbol{Token} = f(Base64Encode)\sum_{n=\propto,\beta}^{\infty}(header.payload.signature) \quad (1)$$

Formulasi *JSON Web Token* (JWT) dibangun dari fungsi *Base64Encode* dengan parameternya *header*, *payload*, dan *signature*. Berikut adalah contoh dari token yang digunakan oleh JWT [5].

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.EkN-DOsnsuRjRO6BxXemmJDm3HbxrbRzXglbN2S4sOkopdU4IsDxTI8jO19W_A4K8ZPJijNLis4EZsHeY559a4DFOd50_OqgHGuERTqYZyuhtF39yxJPAjUESwxk2J5k_4zM3O-vtd1Ghyo4IbqKKSy6J9mTniYJPenn5-HIirE
```

Gambar 4. Contoh dari JWT Token

Dari contoh JWT Token tersebut, terdapat 3 bagian utama dari JWT Token yaitu Header (algoritma dan jenis token), Payload (data), dan Signature (verify signature). Untuk bagian Header, Payload, dan Signature dapat dijelaskan sebagai berikut.

1. Header

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9
```

2. Payload

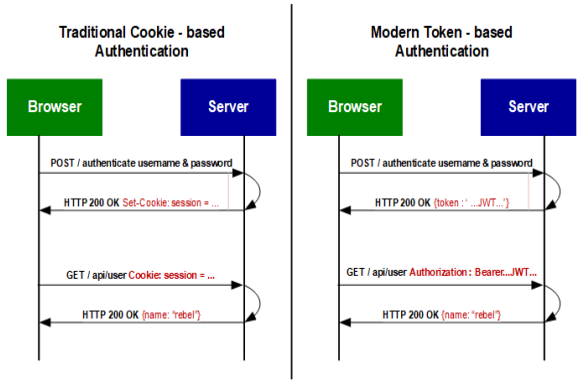
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9

3. Signature

EkN-DOsnsuRjRO6BxXemmJDm3HbxrbRzXglbN2S4sOkopdU4IsDxTI8jO19W_A4K8ZPJijNLis4EZsHeY559a4DFOd50_OqgHGuERTqYZyuhtF39yxJPAjUESwxk2J5k_4zM3O-vtd1Ghyo4IbqKKSy6J9mTniYJPenn5-HIirE

Gambar 5. Tiga Bagian Utama JWT Token

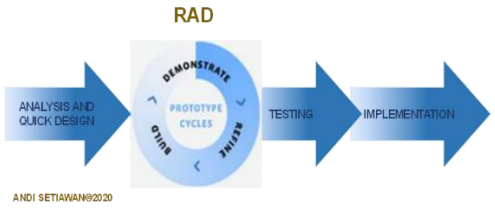
Otentikasi menggunakan token akan lebih aman, terukur, dan dapat digunakan dengan mudah oleh pengguna tanpa membebani kinerja dari server. Mekanisme otentifikasi secara tradisional akan menyimpan banyak ID session diserver yang berisi identitas pengguna dari cookie session di server, sedangkan otentikasi secara modern akan menyimpan informasi penting pada sebuah token, seperti pada gambar berikut ini [5].



Gambar 6. Perbedaan Tradisional Cookie dan Modern Token

2.4 *Rapid Application Development* (RAD)

Pemodelan pengembangan sistem yang digunakan untuk membangun aplikasi BatikKita adalah *Rapid Application Development* (RAD). Keunggulan dari *Rapid Application Development* (RAD) adalah siklus pengembangan sistem yang sangat singkat dibandingkan dengan pemodelan sistem lainnya [19]. Sehingga sangat tepat bila digunakan sebagai tahap pengembangan dalam membangun aplikasi BatikKita. Pada tahap *Analy-sis And Quick Design*, pengembangan sistem difokuskan untuk menganalisa kebutuhan aplikasi berdasarkan masukan dari para pengrajin dan konsumen batik Trusmi Cirebon. Kemudian pada tahap *prototype cycles*, penelitian difokuskan pada pengembangan aplikasi melalui server web lokal. Berikutnya pada tahap testing atau pengujian, penelitian difokuskan pada pengujian server web, sekaligus untuk menguji kehandalan dari *JSON Web Token* (JWT) yang diimplementasikan pada aplikasi BatikKita pada saat otentikasi. Kemudian pada tahap *implementation*, penelitian difokuskan kepada respon dari pengguna terutama para pengrajin batik dan konsumen batik Trusmi Cirebon. Tahap pengembangan sistem BatikKita, digambarkan pada gambar berikut ini.

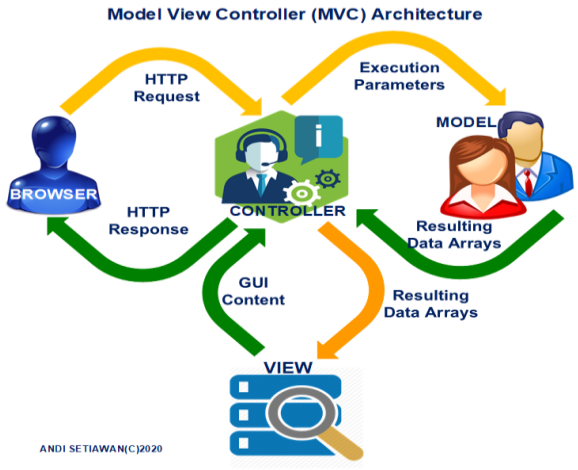


Gambar 7. *Rapid Application Development* (RAD)

2.5 Arsitektur *Model View Controller* (MVC)

Aplikasi BatikKita dibangun berdasarkan arsitek-tur framework dengan memisahkan *design pattern*-nya, yaitu *model*, *view*, dan *controller*. Dengan memisahkan logika pembuatan kode dengan tampilan halaman website dengan menggunakan pola *design pattern* berupa *model*, *view*, dan *controller*, akan menjadikan aplikasi BatikKita menjadi lebih terstruktur dan sederhana untuk mempermudah pada saat mendesign kode program dari aplikasi BatikKita [19].

Secara teknis arsitektur MVC laravel aplikasi BatikKita. online diilustrasikan sebagai berikut, pada bagian tampilan depan atau *frontend* dikembangkan interface berbasis android untuk memberikan input melalui *HTTP Request* ke *design pattern controller* pada bagian *backend* sistem aplikasi BatikKita. Selanjutnya *design pattern model* akan mengeksekusi parameter yang diberikan oleh *design pattern controller* dan memberikan hasil berupa data array ke *design pattern controller*. Untuk *design pattern view*, *design pattern controller* memberikan hasil berupa data array yang sudah diproses sebelumnya dari *design pattern model* oleh *design pattern controller*. Kemudian *design pattern view* memberikan hasil berupa konten berbasis GUI ke *design pattern controller*. Bagian terakhir *design pattern controller* akan menampilkannya pada browser berupa *HTTP Response*, seperti pada gambar berikut ini.



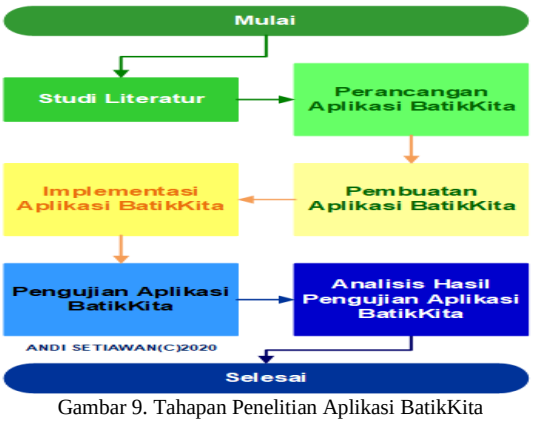
Gambar 8. Arsitektur *Model, View, Dan Controller* (MVC)

3. Hasil dan Pembahasan

Hasil dan pembahasan dari penelitian implementasi *JSON Web Token* (JWT) dengan algoritma *HMAC SHA-512* untuk aplikasi BatikKita, sebagai berikut.

3.1 Analisis dan Design

Pada tahap analisis dan design, penelitian dimulai dari studi literatur yang digunakan dalam penelitian ini, studi literatur dibutuhkan untuk membandingkan hasil dari penelitian sebelumnya terutama dari sisi keunggulan, ketercapaian hasil, efektifitas dan sebagainya terutama kesesuaian topik yang dibahas dalam penelitian ini, yaitu *JSON Web Token* (JWT) dengan algoritma *HMAC SHA-512*. Setelah pengumpulan literatur dan studi literatur, penelitian dilanjutkan dengan perancangan aplikasi BatikKita. Pada tahap ini, penelitian difokuskan pada modifikasi database dan aplikasi yang sudah ada dengan menambahkan android untuk pengembangan aplikasi BatikKita. Kemudian tahap pengembangan aplikasi BatikKita, penelitian difokuskan pada perancangan *web service* yang akan diimplementasikan. Hal ini sangat dibutuhkan untuk menghubungkan dua aplikasi yang berbeda platform yaitu android dan web framework. Setelah dilakukan pengujian, maka *web service* yang digunakan pada aplikasi BatikKita adalah *Restfull Web Service (RWS)*, pemilihan jenis *restfull web service* karena penggunaan *JSON Web Token* (JWT) dengan algoritma *HMAC SHA-512* untuk otentikasi datanya. Setelah tahap pengembangan dari aplikasi BatikKita, penelitian dilanjutkan ke tahap implementasi. Pada tahap ini penelitian difokuskan pada implementasi aplikasi BatikKita pada sebuah web server. Pada tahap ini selain implementasi pada web server, juga dilakukan pengujian terhadap *JSON Web Token* (JWT) dengan algoritma *HMAC SHA-512* menggunakan *Postman Tools* untuk mengetahui kecepatan dan ukuran token yang digunakan dalam pengujian. Sedangkan pada tahap analisis hasil pengujian, penelitian difokuskan pada hasil pengujian *JSON Web Token (JWT)* dengan algoritma *HMAC SHA-256* dan algoritma *HMAC SHA-384* terhadap dua buah arsitektur *web service*, yaitu *Simple Object Access Protocol* (*SOAP*) dan R*epresentational State Transfer* (*REST*), seperti pada gambar berikut ini.



Gambar 9. Tahapan Penelitian Aplikasi BatikKita

Sedangkan alur flowchart dari aplikasi BatikKita digambarkan sebagai berikut.

Gambar 10. Alur Flowchart Aplikasi BatikKita

Penjelasan dari alur flowchart aplikasi BatikKita adalah sebagai berikut. 1) Pengguna dalam hal ini pengrajin dan konsumen batik Trusmi mengakses aplikasi BatikKita dengan memberikan nama pengguna dan kata sandi yang diperoleh pada saat pendaftaran melalui platform android dan platform web framework. 2) Nama pengguna serta kata sandi kemudian diotentikasi menggunakan *Java Script Object Notation (JSON) Web Token* atau disingkat menjadi JWT. 3) Nama pengguna dan kata sandi yang sudah diotentikasi kemudian diverifikasi oleh sistem dan menunggu proses selanjutnya. 4) Pengguna kemudian diverifikasi oleh sistem untuk dapat mengakses lebih lanjut aplikasi BatikKita. 5) Pengguna dapat mengakses foto jenis batik dari aplikasi BatikKita yang tersimpan dalam database ataupun harga serta ongkos kirim yang harus dibayarkan oleh pengguna. 6) Admin sistem mengecek halaman *backend system* dengan memberikan nama pengguna serta kata sandi. 7) Nama pengguna serta kata sandi kemudian diotentikasi menggunakan *Java Script Object Notation (JSON) Web Token*. 8) Nama pengguna dan kata sandi yang sudah diotentikasi kemudian diverifikasi oleh sistem dan menunggu proses selanjutnya. 9) Admin Sistem kemudian diverifikasi oleh sistem untuk dapat mengakses lebih lanjut aplikasi BatikKita.

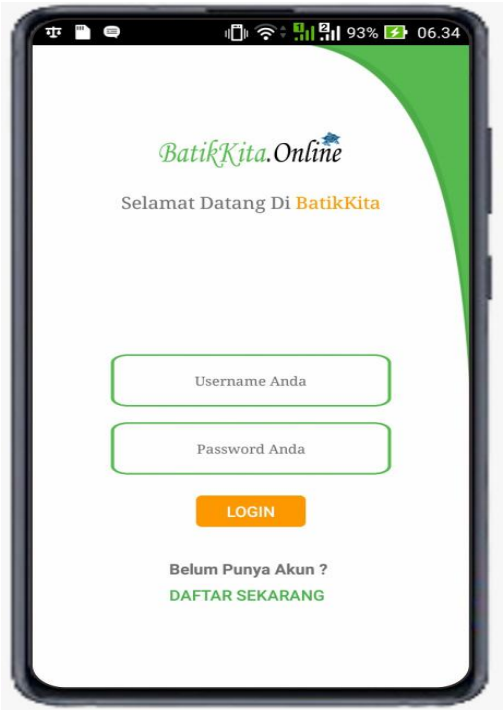
3.2 Testing Dan Implementasi

Pada tahap ini dilakukan pengujian dan implementasi aplikasi BatikKita. Aplikasi BatikKita berbasis android untuk bagian *frontend* dan web framework laravel untuk bagian *backend*. Bagian frontend digunakan untuk ber-

interaksi antara pengrajin dan konsumen batik Trusmi dengan aplikasi BatikKita. Halaman pertama akan memunculkan *splash screen* aplikasi BatikKita seperti pada gambar berikut ini.

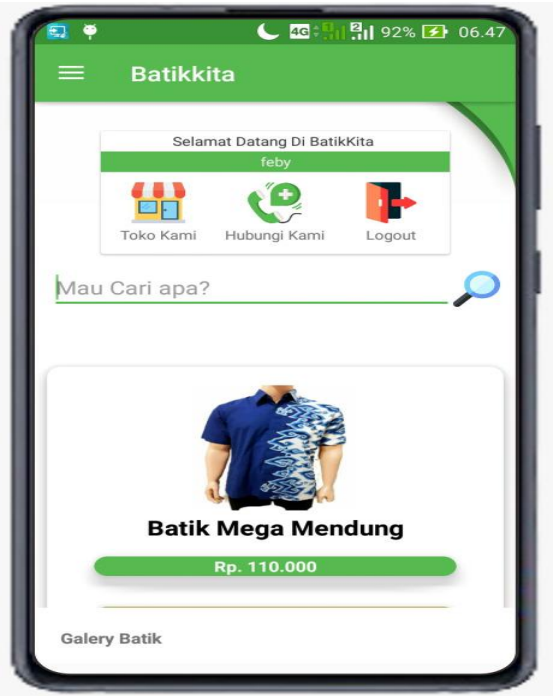
Gambar 11. Splash Screen Aplikasi BatikKita

Berikutnya aplikasi BatikKita, akan menampilkan halam-an login untuk otentikasi data pengguna baik pengrajin atau konsumen batik Trusmi pada aplikasi BatikKita yang sudah diimplementasikan dengan *JSON Web Token* (JWT) dengan algoritma *SHA-512*, seperti pada gambar berikut ini.



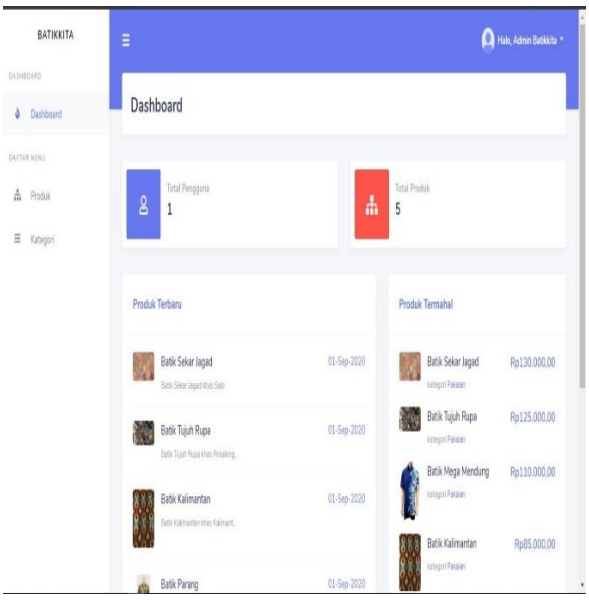
Gambar 12. Halaman Login Aplikasi BatikKita

Apabila proses otentikasi berhasil dilalui oleh pengguna aplikasi BatikKita baik pengrajin ataupun konsumen, pengguna diarahkan ke halaman utama aplikasi batik kita seperti pada gambar berikut ini.



Gambar 13. Halaman Utama Aplikasi BatikKita

Sedangkan untuk halaman backend digunakan oleh pengelola aplikasi BatikKita untuk mengatur aktifitas penjualan batik Trusmi antara pengrajin dan konsumen dengan aplikasi BatikKita, seperti pada gambar berikut ini.



Gambar 14. Halaman Utama BackEnd Aplikasi BatikKita

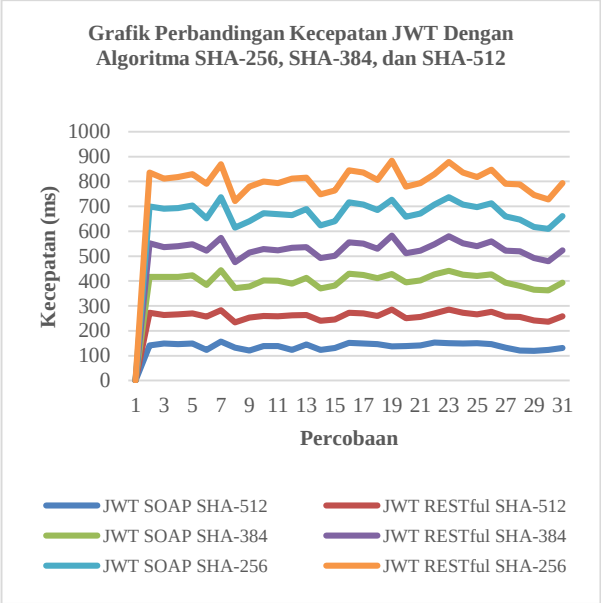
3.3 Pengujian

Untuk mengetahui kinerja *JSON Web Token* (JWT) dengan algoritma *SHA-512* dilakukan pengujian dari sisi kecepatan dengan tool *Postman*. Pengujian dilakukan dengan membandingkan kecepatan antara arsitektur layanan web *Simple Object Access Protokol* (SOAP) dengan *Representational State Transfer* (REST) pada aplikasi BatikKita. Pengujian dilakukan sebanyak 30 kali pengujian untuk mengetahui kecepatan *JSON Web Token* (JWT) dengan algoritma *SHA-512*. Sebagai perbandingan dilakukan juga pengujian terhadap algoritma *SHA*-256 dan algoritma *SHA*-384. Berikut hasil pengujian terhadap *JSON Web Token* (JWT) dengan algoritma *SHA*-256, *SHA*-384, dan *SHA-512* dari sisi kecepatan antara arsitektur layanan web *Simple Object Access Protokol* (SOAP) dengan *Representational State Transfer* (REST).

Tabel 3. Perbandingan Kecepatan *JSON Web* Token (JWT)

Percobaan	Kecepatan (ms)					
	SHA-256		SHA-384		SHA-512	
	SOAP	REST	SOAP	REST	SOAP	REST
1	147	137	145	135	141	131
2	155	121	153	119	149	115
3	153	125	151	123	147	119
4	155	127	153	125	149	121
5	130	139	128	137	124	133
6	163	132	161	130	157	126
7	139	107	137	105	133	101
8	127	138	125	136	121	132
9	145	127	143	125	139	121
10	145	125	143	123	139	119
11	131	147	128	144	123	139
12	153	126	150	123	145	118
13	132	125	129	122	124	117
14	139	123	136	120	131	115
15	160	129	157	126	152	121
16	157	129	154	126	149	121
17	155	121	152	118	147	113
18	145	157	142	154	137	149
19	147	120	144	117	139	112
20	149	123	146	120	141	115
21	159	123	157	121	153	117
22	157	141	155	139	151	135
23	155	129	153	127	149	123
24	157	121	155	119	151	115
25	153	135	151	133	147	129
26	138	131	136	129	132	125
27	127	141	125	139	121	135
28	125	129	123	127	119	123
29	129	119	127	117	123	113
30	137	133	135	131	131	127
Average	**145,5**	**129,3**	**143,1**	**127,0**	**138,8**	**122,7**

Dari tabel 3 diperoleh rata-rata dari kecepatan *JSON Web Token* (JWT) dengan algoritma *SHA*-256, *SHA*-384, dan *SHA-512*. Untuk rata-rata kecepatan arsitektur layanan web *Simple Object Access Protokol* (SOAP) dengan algoritma *SHA-256* adalah 145,5 milisecond, algoritma *SHA-384* lebih cepat dari algoritma *SHA-256* dengan rata-rata kecepatan 143,1 milisecond, dan algoritma *SHA-512* lebih cepat dibandingkan deng-an algoritma *SHA-384* dan algoritma algoritma *SHA-256* dengan rata-rata kecepatan 138,8 milisecond. Sedang-kan rata-rata kecepatan arsitektur untuk layanan web *Representational State Transfer* (REST) dengan algorit-ma *SHA-256* adalah 129,3 milisecond, algoritma *SHA-384* lebih cepat dibandingkan algoritma *SHA-256* dengan rata-rata kecepatan 127,0 milisecond, dan algoritma *SHA-512* lebih cepat dibandingkan dengan algoritma *SHA-384* dan algoritma algoritma *SHA-256* rata-rata kecepatan 122,7 milisecond. Hasil perbanding-an rata-rata dari kecepatan *JSON Web Token* (JWT) digambarkan dengan grafik perbandingan berikut ini.



Gambar 15. Grafik Perbandingan Kecepatan JWT Aplikasi BatikKita

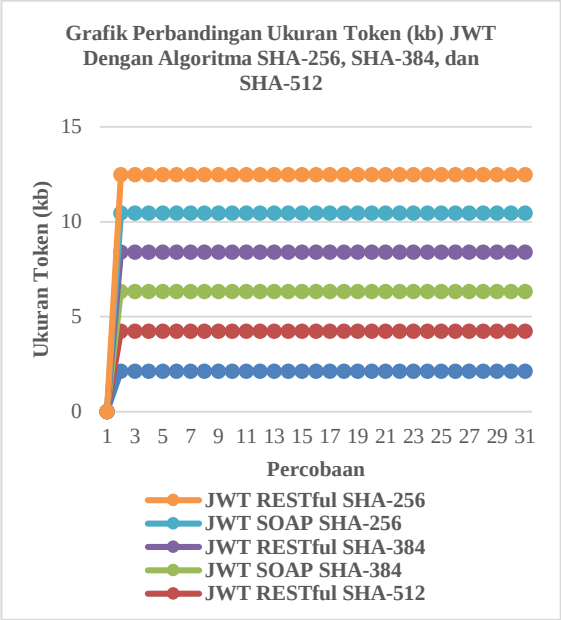
Pengujian berikutnya dilakukan untuk mengetahui ukuran dari masing-masing *JSON Web Token* (JWT) dengan algoritma *SHA-512*, algoritma *SHA*-256 dan algoritma *SHA*-384. Berikut hasil pengujian ter-hadap *JSON Web Token* (JWT) dengan algoritma *SHA*-256, *SHA*-384, dan *SHA-512* dari sisi ukuran antara arsitektur layanan web *Simple Object Access Protokol* dengan *Representational State Transfer*.

Tabel 4. Perbandingan Ukuran *JSON Web* Token (JWT)

Percobaan	Ukuran (kb)					
	SHA-256		SHA-384		SHA-512	
	SOAP	REST	SOAP	REST	SOAP	REST
1	2,05	2,03	2,09	2,07	2,13	2,11
2	2,05	2,03	2,09	2,07	2,13	2,11
3	2,05	2,03	2,09	2,07	2,13	2,11

4	2,05	2,03	2,09	2,07	2,13	2,11
5	2,05	2,03	2,09	2,07	2,13	2,11
6	2,05	2,03	2,09	2,07	2,13	2,11
7	2,05	2,03	2,09	2,07	2,13	2,11
8	2,05	2,03	2,09	2,07	2,13	2,11
9	2,05	2,03	2,09	2,07	2,13	2,11
10	2,05	2,03	2,09	2,07	2,13	2,11
11	2,05	2,03	2,09	2,07	2,13	2,11
12	2,05	2,03	2,09	2,07	2,13	2,11
13	2,05	2,03	2,09	2,07	2,13	2,11
14	2,05	2,03	2,09	2,07	2,13	2,11
15	2,05	2,03	2,09	2,07	2,13	2,11
16	2,05	2,03	2,09	2,07	2,13	2,11
17	2,05	2,03	2,09	2,07	2,13	2,11
18	2,05	2,03	2,09	2,07	2,13	2,11
19	2,05	2,03	2,09	2,07	2,13	2,11
20	2,05	2,03	2,09	2,07	2,13	2,11
21	2,05	2,03	2,09	2,07	2,13	2,11
22	2,05	2,03	2,09	2,07	2,13	2,11
23	2,05	2,03	2,09	2,07	2,13	2,11
24	2,05	2,03	2,09	2,07	2,13	2,11
25	2,05	2,03	2,09	2,07	2,13	2,11
26	2,05	2,03	2,09	2,07	2,13	2,11
27	2,05	2,03	2,09	2,07	2,13	2,11
28	2,05	2,03	2,09	2,07	2,13	2,11
29	2,05	2,03	2,09	2,07	2,13	2,11
30	2,05	2,03	2,09	2,07	2,13	2,11
Average	**2,05**	**2,03**	**2,09**	**2,07**	**2,13**	**2,11**

Dari tabel 4, diperoleh rata-rata perbandingan dari ukuran token *JSON Web Token* (JWT) dengan algoritma *SHA*-256, *SHA*-384, dan *SHA-512*. Token yang dihasilkan oleh algoritma *SHA-256* berukuran 256 bit sehingga ukuran yang dibutuhkan oleh token dengan *SHA-256* sangat kecil bila dibandingkan dengan algoritma *SHA-384* berukuran 384 bit dan algoritma *SHA-512* yang berukuran 512 bit. Ukuran token *SHA-512* lebih panjang dibandingkan dengan *SHA-384* dan *SHA-256* dikarenakan berbedaan bit yang digunakan oleh masing-masing algoritma. Untuk rata-rata ukuran token arsitektur layanan web *Simple Object Access Protokol* (SOAP) dengan algoritma *SHA-256* ukuran token yang digunakan sebesar 2,05 kb, algoritma *SHA-384* lebih panjang dari algoritma *SHA-256* dengan rata-rata ukuran token 2,09 kb, dan algoritma *SHA-512* lebih panjang ukuran token yang digunakan dibandingkan dengan algoritma *SHA-384* dan algoritma algoritma *SHA-256* dengan rata-rata ukuran token yang digunakan sebesar 2,13 kb. Sedangkan rata-rata ukuran token arsitektur untuk layanan web *Representational State Transfer* (REST) dengan algoritma *SHA-256* adalah 2,03 kb, algoritma *SHA-384* lebih panjang ukuran token yang digunakan dibandingkan algoritma *SHA-256* dengan ukuran token sebesar 2,07 kb, dan algoritma *SHA-512* lebih panjang dibandingkan dengan algoritma *SHA-384* dan algoritma algoritma *SHA-256* rata-rata ukuran token yang digunakan sebesar 2,11 kb. Hasil perbandingan rata-rata dari kecepatan *JSON Web Token* (JWT) digambarkan dengan grafik perbandingan berikut ini.



Gambar 16. Grafik Perbandingan Ukuran Token JWT Aplikasi BatikKita

Dari hasil pengujian implementasi *JSON Web Token* (JWT) dengan algoritma *SHA-512* untuk membandingkan kecepatan dan ukuran token yang dibutuhkan antara arsitektur layanan web *Simple Object Access Protokol* dengan *Representational State Transfer* pada aplikasi BatikKita, dilakukan sebanyak 30 (tiga puluh) kali pengujian, diperoleh hasil berupa kecepatan dapat meningkat dengan rata-rata antara **138,8 milisecond** untuk SOAP dan **122,7 milisecond** untuk REST. Sedangkan ukuran token yang dibutuhkan antara **2,13 kb** untuk SOAP dan **2,11 kb** untuk REST.

4. Kesimpulan

Kesimpulan yang diperoleh dari hasil pengujian yang dilakukan pada penelitian ini adalah implementasi *JSON Web Token* (JWT) dengan algoritma *SHA-512* pada aplikasi BatikKita dapat mempercepat pada saat proses otentikasi serta dapat meningkatkan keamanan karena penggunaan token pada saat otentikasi. Sehingga implementasi *JSON Web Token* (JWT) dengan algoritma *SHA-512* pada aplikasi BatikKita berbasis android dan web framework laravel sangat tepat diimplementasikan.

Ucapan Terima Kasih

Direktorat Riset dan Pengabdian Masyarakat (DRPM) Kementrian Ristekdikti.

Daftar Rujukan

[1] A. Rahmatulloh, H. Sulastri, dan R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *JNTETI*, vol. 7, no. 2, hal. 131–137, 2018.

[2] R. Gunawan dan A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *JEPIN (Jurnal Edukasi dan Penelit. Inform.*, vol. 5, no. 1, hal. 74–79, 2019.

[3] Edy, Ferdiansyah, W. Pramusinto, dan S. Waluyo, "Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order," *J. RESTI*, vol. 3, no. 2, hal. 106–112, 2019.
[4] A. P. Aldya, A. Rahmatulloh, dan M. N. Arifin, "Stateless Authentication with JSON Web Tokens using RSA-512 Algorithm," *J. INFOTEL*, vol. 11, no. 2, hal. 36–42, 2019.
[5] A. Rahmatulloh, R Gunawan, dan F. M. S. Nursuwars, "Performance comparison of signed algorithms on JSON Web Token Performance comparison of signed algorithms on JSON Web Token," in *SICIR*, 2019, hal. 1–9.
[6] B. E. Sabir, M. Youssfi, O. Bouattane, dan H. Allali, "Authentication and load balancing scheme based on JSON Token for Multi-Agent Systems," *Procedia Comput. Sci.*, vol. 148, hal. 562–570, 2019.
[7] L. V. Jánoky, J. Levendovszky, dan P. Ekler, "An analysis on the revoking mechanisms for JSON Web Tokens," *Int. J. Distrib. Sens. Networks*, vol. 14, no. 9, hal. 1–10, 2018.
[8] Z. Da, W. Yang, P. Ran, dan Y. Huo, "Program Design of JSON to Structured Data Conversion," in *MATEC Web of Conferences*, 2017, vol. 139, hal. 1–4.
[9] A. W. P. Putra, A. Bhawiyuga, dan M. Data, "Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 2, hal. 584–593, 2018.
[10] D. I. Nassr, "Secure Hash Algorithm-2 formed on DNA," *J. Egypt. Math. Soc.*, vol. 27, no. 1, hal. 1–20, 2019.
[11] M. Ahmad, E. Al Solami, X. Y. Wang, M. N. Doja, M. M. Sufyan Beg, dan A. A. Alzaidi, "Cryptanalysis of an image encryption algorithm based on combined chaos for a BAN system, and improved scheme using SHA-512 and hyperchaos," *Symmetry (Basel).*, vol. 10, no. 7, hal. 1–18, 2018.
[12] M. A. Nazal, R. Pulungan, dan M. Riasetiawan, "Data Integrity and Security using Keccak and Digital Signature Algorithm (DSA)," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 3, hal. 273–282, 2019.
[13] A. Sideris, T. Sanida, dan M. Dasygenis, "High Throughput Implementation of the Keccak Hash Function Using the Nios-II Processor," *Technologies*, vol. 8, no. 1, hal. 15, 2020.
[14] N. Abdoun, S. El Assad, T. M. Hoang, O. Deforges, R. Assaf, dan M. Khalil, "Designing Two Secure Keyed Hash Functions Based on Sponge Construction and the Chaotic Neural Network," *Entropy J.*, vol. 2, no. 2, hal. 1–32, 2020.
[15] H. Tiwari dan K. Asawa, "A secure and efficient cryptographic hash function based on NewFORK-256," *Egypt. Informatics J.*, vol. 13, no. 3, hal. 199–208, 2012.
[16] K. Seetharaman dan R. Ragupathy, "LDPC and SHA based iris recognition for image authentication," *Egypt. Informatics J.*, vol. 13, no. 3, hal. 217–224, 2012.
[17] Andri, "Penerapan Algoritma Pencarian Binary Search dan QuickSort pada Aplikasi Kamus Bahasa Palembang Berbasis Web," *J. Inform. J. Pengemb. IT*, vol. 04, no. 01, hal. 70–74, 2019.
[18] A. P. Atmaja dan A. Azis, "Sistem Informasi Terintegrasi Evaluasi Kegiatan Mengajar Dosen Sebagai Implementasi Sistem Penjaminan Mutu Internal," *J. Matrix*, vol. 9, no. 1, hal. 1–6, 2019.
[19] R. R. Sani dan D. Kurniawan, "Rancang Bangun Sistem Try Out Berbasis Paperless Untuk Evaluasi Hasil Belajar Siswa Dengan MVC," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 3, hal. 277–286, 2019.
[20] A. P. Atmaja dan S. V. Yulianto, "Integrated Student portal Menggunakan Metode Pengembangan Siklus Pendek," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 03, no. 01, hal. 24–31, 2018.
[21] A. H. Faqih, T. G. Laksana, dan A. Febriati, "Sistem informasi reporting curiculum vitae karyawan menggunakan metode rapid application development berbasis website di PT. PINS Indonesia," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 03, no. 01, hal. 69–75, 2018.
[22] S. Mcdonald *et al.*, "Environmental Modelling & Software Web-based decision support system tools : The Soil and Water Assessment Tool Online visualization and analyses (SWATOnline) and NASA earth observation data downloading and reformatting tool (NASAaccess)," *Environ. Model. Softw.*, vol. 120, no. August, hal. 104499, 2019.
[23] O. Dagdeviren, V. K. Akram, dan A. Farzan, "A Distributed Evolutionary Algorithm for Detecting Minimum Vertex Cuts for Wireless Ad hoc and Sensor Network," *J. Netw. Comput. Appl.*, vol. X, hal. 1–39, 2018.
[24] H. Wu, C. Bailey, P. Rasoulinejad, dan S. Li, "Automated Comprehensive Adolescent Idiopathic Scoliosis," *Med. Image Anal.*, hal. 1–31, 2018.
[25] J. S. Jeong dan Á. Ramírez-Gómez, "Development of a web graphic model with Fuzzy-DEcision-MAking Trial and Evaluation Laboratory/Multi-Criteria-Spatial Decision Support System (F-DEMATEL/MC-SDSS) for sustainable planning and construction of rural housings," *J. Clean. Prod.*, hal. 1–31, 2018.
[26] A. Sunardi dan Suharjito, "ScienceDirect ScienceDirect MVC Architecture : A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based," *Procedia Comput. Sci.*, vol. 157, hal. 134–141, 2019.
[27] Y. Rhazali, Y. Hadi, dan A. Mouloudi, "Model Transformation with ATL into MDA from CIM to PIM Structured through MVC," *Procedia - Procedia Comput. Sci.*, vol. 83, no. Fams, hal. 1096–1101, 2016.
[28] N. Prokofyeva dan V. Boltunova, "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems," *Procedia - Procedia Comput. Sci.*, vol. 104, no. December 2016, hal. 51–56, 2017.
[29] D. Pop dan A. Altar, "Designing an MVC Model for Rapid Web Application Development," *Procedia Eng.*, vol. 69, hal. 1172–1179, 2014.
[30] J. Gracia dan E. Bayo, "An effective and user-friendly web application for the collaborative analysis of steel joints," *Adv. Eng. Softw.*, vol. 119, no. March 2017, hal. 60–67, 2018.
[31] A. Zaif dan A. E. Cerchia, "Integrating Online Marketing Strategies in B2B Companies," *Ovidius Univ. Ann. Econ. Sci. Ser.*, vol. XIX, no. 2, hal. 614–620, 2019.
[32] P. Jasek, L. Vrana, L. Sperkova, Z. Smutny, dan M. Kobulsky, "Comparative analysis of selected probabilistic customer lifetime value models in online shopping," *J. Bus. Econ. Manag.*, vol. 20, no. 3, hal. 398–423, 2019.
[33] J. Zhang, L. Li, dan Y. Qian, "A Study of Online Review Promptness in a B2C System," *Discret. Dyn. Nat. Soc.*, vol. 2016, hal. 1–10, 2016.
[34] N. B. Puspitasari, S. N. W. P, D. N. Amyhorsea, dan A. Susanty, "Consumer ' s Buying Decision -Making Process in E-Commerce," in *E3S Web of Conferences*, 2018, vol. 11003, no. September 2016, hal. 1–6.
[35] N. AlMajed, L. A. Maglaras, F. Siewe, H. Janicke, dan P. Bagheri Zadeh, "Prevention of crime in B2C E-Commerce: How E-Retailers/Banks protect themselves from Criminal Sctivities," *ICST Trans. Secur. Saf.*, vol. 3, no. 7, hal. 1–15, 2016.
[36] W. Xu dan B. Li, "The third party logistics partner selection of B2C E-commerce enterprise," *MATEC Web Conf.*, vol. 100, hal. 1–10, 2017.
[37] R. Li dan T. Sun, "Assessing factors for designing a successful B2C E-Commerce website using fuzzy AHP and TOPSIS-Grey methodology," *Symmetry (Basel).*, vol. 12, no. 3, hal. 1–26, 2020.