

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Penelitian yang dilakukan untuk mengamankan *web service* REST API menggunakan metode JWT telah banyak dilakukan. Tidak sedikit pula diantaranya mampu memberikan hasil pengamanan yang baik untuk pertukaran datanya. Dari banyaknya penelitian yang sudah dilakukan sebelumnya menunjukkan bahwa pengamanan *web service* dapat memberikan dampak yang baik dalam keamanan datanya. Berikut penelitian terdahulu yang menurut penulis mempunyai keterkaitan dengan penelitian yang akan dilakukan.

Pertama, penelitian yang berjudul "Pengamanan REST API menggunakan JWT untuk Aplikasi *Sales Order*" dilakukan pada tahun 2019 oleh Edy, Ferdiansyah, Wahyu Pramusinto, dan Sejati Waluyo. Studi ini menunjukkan bahwa penggunaan autentikasi *JSON Web Token* (JWT) dengan REST API dalam konteks aplikasi *Sales Order* memberikan hasil yang positif. Aplikasi *Sales Order* ini dapat bekerja dengan baik baik pada platform *web* maupun Android sehingga mempermudah proses pemesanan produk. Penggunaan JWT membuat aplikasi ini lebih aman, karena hanya pengguna yang memiliki token yang dapat mengaksesnya, mengurangi risiko akses yang tidak sah ke aplikasi tersebut karena hanya pengguna dengan token yang dapat mengakses aplikasi tersebut. Oleh karena itu, penelitian ini memiliki keterkaitan dengan penelitian yang akan dilakukan dalam menjamin keamanan layanan REST API *Web Services* [8].

Kedua, penelitian yang berjudul "Implementasi Fitur Keamanan dengan *JSON Web Token* dan Fitur *Geo-tagging* pada Aplikasi *Web Service Training From Home*" pada tahun 2020 oleh Aal Hibsy dan Arief Wibowo. Penelitian ini berfokus pada keamanan dan *geotagging* berbasis *JSON Web Token* (JWT) dan menyajikan hasil implementasi penting dalam pengembangan aplikasi layanan *web* berbasis *Android*. Hasil penelitian ini menunjukkan bahwa penerapan JWT pada *web service* mampu menghasilkan token sebagai sarana autentikasi

pengguna yang berhak mengakses aplikasi Presensi Atlet *Training From Home*. Ini memberikan keyakinan bahwa aspek keamanan dalam hak akses selama pertukaran data telah terjamin dan diimplementasikan dengan baik. Pengguna yang login akan menerima token pribadi yang memiliki batas waktu selama 24 jam, mengamankan akses aplikasi dari pihak yang tidak sah. Selain itu, evaluasi fitur geotagging menggunakan metode kompatibilitas *Google Maps* menunjukkan tingkat akurasi sebesar 90,9%, memastikan aplikasi berbasis Android ini memberikan informasi yang akurat kepada atlet peserta program *Training From Home* memberikan akurasi posisi dengan memberikan akurasi lokasi [9].

Ketiga, penelitian yang berjudul “Implementasi *JSON Web Token* Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita” dilakukan oleh Andi Setiawan dan Ade Irma Purnamasari pada tahun 2020. Penelitian ini mengungkapkan bahwa implementasi *JSON Web Token* (JWT) dengan algoritma SHA-512 pada aplikasi BatikKita memiliki dua manfaat utama. Pertama, implementasi ini berhasil mempercepat proses otentikasi untuk aplikasi tersebut. Kedua, penggunaan token pada saat otentikasi secara signifikan meningkatkan tingkat keamanan. Oleh karena itu, dapat disimpulkan bahwa penerapan *JSON Web Token* (JWT) dengan algoritma SHA-512 pada aplikasi BatikKita yang berbasis Android dan *web framework* Laravel merupakan langkah yang sangat tepat dan bermanfaat [10].

Keempat, penelitian yang berjudul “Pengamanan REST API Web Service Menggunakan *Json Web Token* (Studi Kasus: Aplikasi Siakadu Mobile Unesa)” dilakukan oleh Jihad Satrio Utama dan Aries Dwi Indriyati pada tahun 2023. Hasil penelitian ini menunjukkan bahwa kerentanan keamanan yang teridentifikasi pada permintaan data *loginless* berhasil diatasi dengan penerapan token web *JSON* dan algoritma HMAC SHA-512. Seiring dengan itu, pengguna harus melakukan *login* untuk menerima token yang kemudian digunakan dalam permintaan data, dan hasil pengujian menegaskan bahwa implementasi ini tidak hanya meningkatkan keamanan tetapi juga menjadikan proses permintaan data lebih aman dalam aplikasi tersebut [11].

Kelima, penelitian yang berjudul “Implementasi JSON Web Token Authentication pada Aplikasi Pembayaran Berbasis Mobile” dilakukan oleh Sopingi, Faulinda Ely Nastiti, dan Arvian Sofyan Majid pada tahun 2021. Hasil penelitian menunjukkan bahwa sistem pembayaran berbasis *mobile* mampu mengelola berbagai jenis data seperti detail tagihan, *invoice*, profil mahasiswa, dan riwayat pembayaran. Penggunaan *Web Service* dan autentikasi JWT memberikan keamanan data yang baik dan memudahkan akses ke *server database*. Selain itu, aplikasi pembayaran yang dikembangkan melalui penelitian ini telah diuji dan terbukti bebas dari kesalahan (*bug*) serta sesuai dengan kebutuhan fungsional yang diinginkan [12].

Tabel 2.1 Tabel Penelitian Sebelumnya

No	Judul, Penulis, Tahun	Studi kasus	Metode / Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
1	Pengamanan REST API <i>Web Service</i> Menggunakan <i>Json Web Token</i> , oleh Jihad Satrio Utama dan Aries Dwi Indriyati, Tahun 2023	Aplikasi Siakadu <i>Mobile</i> Unesa.	Metode Kualitatif.	Hasil pada penelitian ini yaitu dengan menerapkan JSON Web Token dan algoritma HMAC SHA-512, celah keamanan yang teridentifikasi dalam permintaan data tanpa <i>login</i> berhasil tertutup. Seiring dengan itu, pengguna harus melakukan <i>login</i> untuk menerima token yang kemudian digunakan dalam permintaan data.	Pada penelitian sebelumnya peneliti menggunakan metode autentikasi Login pada aplikasi mobile, sedangkan pada penelitian ini JSON Web Token digunakan sebagai autentikasi login pada web.
2	Implementasi <i>JSON Web Token Authentication</i> pada Aplikasi Pembayaran Berbasis Mobile, oleh Sopingi, Faulinda Ely	Universitas Duta Bangsa.	<i>Rapid Application Development</i> (RAD).	Hasil penelitian menunjukkan bahwa sistem pembayaran berbasis <i>mobile</i> mampu mengelola berbagai jenis data seperti detail tagihan, <i>invoice</i> , profil mahasiswa, dan riwayat pembayaran. Penggunaan <i>Web</i>	Penelitian sebelumnya berfokus pada pengamanan data dan autentikasi pada aplikasi mobile,

No	Judul, Penulis, Tahun	Studi kasus	Metode / Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
	Nastiti, dan Arvian Sofyan Majid, Tahun 2021.			<i>Service</i> serta JWT untuk autentikasi dapat memberikan keamanan data dan kemudahan akses ke <i>database</i> server. Selain itu, aplikasi pembayaran yang dikembangkan melalui penelitian ini telah diuji dan terbukti bebas dari kesalahan ( <i>bug</i> ) serta sesuai dengan kebutuhan fungsional yang diinginkan.	sedangkan pada penelitian ini berfokus pada pengamanan data dan autentikasi pada <i>web</i> .
3	Implementasi JSON Web Token Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita, oleh Andi Setiawan dan Ade Irma Purnamasari, Tahun 2020	Aplikasi BatikKita.	<i>Rapid Application Development</i> (RAD).	Penelitian ini memiliki dua manfaat utama. Pertama, implementasi ini berhasil mempercepat proses otentikasi dalam aplikasi. Kedua, penggunaan token pada saat otentikasi signifikan meningkatkan tingkat keamanan.	Pada penelitian sebelumnya peneliti berfokus pada kinerja <i>JSON Web Token</i> dengan algoritma SHA-512, sedangkan pada penelitian ini JSON Web Token digunakan untuk memenuhi kebutuhan autentikasi

No	Judul, Penulis, Tahun	Studi kasus	Metode / Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
					untuk pengamanan data.
4	Implementasi Fitur Keamanan dengan JSON <i>Web Token</i> dan Fitur <i>Geo-tagging</i> pada Aplikasi <i>Web Service Training From Home</i> , oleh Aal Hibsya dan Arief Wibowo, Tahun 2020	Aplikasi <i>Web Service Training From Home</i> .	Eksperimental.	Penerapan JWT pada <i>web service</i> mampu menghasilkan token sebagai alat autentikasi pengguna yang berhak mengakses aplikasi <i>Presensi Atlet Training From Home</i> . Pengguna yang login akan menerima token pribadi yang memiliki batas waktu selama 24 jam, mengamankan akses aplikasi dari pihak yang tidak sah.	Pada penelitian sebelumnya penelitian lebih berfokus pada autentikasi pengguna dan fitur <i>Geo-tagging</i> untuk memenuhi kebutuhan fungsional dari sistem presensi mahasiswa atlet, sedangkan pada penelitian ini lebih berfokus pada keamanan pertukaran data di REST API pada

No	Judul, Penulis, Tahun	Studi kasus	Metode / Algoritma	Hasil	Perbedaan dengan penelitian yang dilakukan
					<i>web service CV.</i> Marvelindo Utama.
5	Pengamanan REST API menggunakan JWT untuk Aplikasi Sales <i>Order</i> , oleh Edy, Ferdiansyah, Wahyu Pramusinto, dan Sejati Waluyo, Tahun 2019.	Aplikasi Sales <i>Order</i> .	<i>Waterfall.</i>	Aplikasi Sales <i>Order</i> ini dapat beroperasi dengan baik baik pada platform web maupun Android, yang mempermudah proses pemesanan produk. Penggunaan JWT membuat aplikasi ini lebih aman, karena hanya pengguna yang memiliki token yang dapat mengaksesnya, mengurangi risiko akses yang tidak sah ke aplikasi tersebut.	Pada penelitian sebelumnya JSON Web Token digunakan sebagai autentikasi untuk aplikasi android, sedangkan di penelitian ini digunakan sebagai autentikasi pada <i>web</i> .

## 2.2 Landasan Teori

### 2.2.1 *Supply Chain Management*

*Supply Chain Management*(SCM) merupakan serangkaian strategi yang dirancang untuk mengoptimalkan integrasi antara pemasok, produksi, penyimpanan, dan distribusi. Tujuannya adalah untuk memastikan bahwa produk disediakan dan disebar dalam jumlah dan lokasi yang tepat untuk mengurangi biaya serta meningkatkan kepuasan pelanggan [13]. SCM juga merupakan pengembangan lebih lanjut dari pengelolaan distribusi produk untuk memenuhi kebutuhan konsumen. Aktivitas dari *supplier* hingga ke konsumen disatukan dalam satu kesatuan sehingga mekanisme informasi antar elemen tersebut berlangsung secara jelas dan terbuka [14]. Tujuan utama dari SCM adalah menciptakan sistem yang mampu mengoordinasikan dan mengelola aliran produk secara optimal. Hal ini menjadi semakin penting mengingat dinamika pasar yang terus berubah, di mana kebutuhan pelanggan terus berkembang, sementara jumlah *retailer* juga terus bertambah dengan cepat. Dengan mengintegrasikan strategi, logistik, dan manajemen persediaan, *supply chain management* bertujuan untuk menjaga ketersediaan produk, meningkatkan efisiensi operasional, dan memberikan kepuasan pelanggan, sambil tetap mengoptimalkan biaya dan keuntungan [15].

### 2.2.2 REST API

REST API, atau *Representational State Transfer API*, adalah jenis arsitektur *Application Programming Interface* (API) yang digunakan dalam pengembangan layanan web. REST API memungkinkan berbagai sistem berkomunikasi satu sama lain dengan menggunakan pendekatan komunikasi yang disebut REST. REST API memetakan sumber daya (*resource*) yang ada dalam database sistem dengan menggunakan endpoint API. Dalam mengakses data, REST API menggunakan perintah HTTP *request* untuk mendapatkan sumber daya, dan format JSON sering digunakan untuk merepresentasikan sumber daya dalam REST. Dengan kata lain, REST API adalah pendekatan untuk memungkinkan komunikasi antara berbagai sistem



melalui protokol HTTP, dengan menggunakan representasi sumber daya yang mudah dipahami, seperti JSON [16].

### **2.2.3 JSON Web Token**

*JSON Web Token* (JWT) adalah sebuah *string* token yang digunakan untuk tujuan otentikasi dan pertukaran data. JWT terdiri dari tiga komponen utama, yaitu *header*, *payload*, dan *signature*. Terdapat dua jenis token, yakni token pembawa (*bearer token*) dan token pemegang kunci (*key token*), serta dua skema berdasarkan tujuan, yaitu token identitas dan token akses. Prinsip kerja JWT serupa dengan penggunaan password dalam proses otentikasi, dimana server menghasilkan token JWT setelah pengguna berhasil login, dan token tersebut disimpan di *local storage* atau *cookies* di *browser* pengguna. Token JWT digunakan untuk memberikan izin akses ke halaman tertentu, dan pengguna harus mengirim token tersebut sebagai bukti bahwa mereka telah berhasil *login*. Dengan demikian, JWT menjadi sebuah elemen penting dalam proses otentikasi dan pengaturan izin dalam aplikasi web [17].

### **2.2.4 Rapid Application Development**

*Rapid Application Development* (RAD) adalah sebuah model proses pengembangan perangkat lunak yang bersifat incremental, terutama digunakan dalam proyek-proyek dengan waktu pengerjaan yang singkat. Pendekatan ini menekankan peningkatan bertahap dalam pengembangan perangkat lunak [18].

Gambar 2. 1 Tahapan RAD [19].



RAD terbagi menjadi tiga tahapan yang saling berkaitan satu sama lain, yaitu:

1. *Requirements Planning* (Perencanaan Kebutuhan)

Pada fase ini, *user* dan *analyst* melakukan semacam pertemuan untuk melakukan identifikasi tujuan dari aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan. Yang terpenting pada tahap ini adalah partisipasi kedua belah pihak, serta persetujuan terhadap usulan yang diajukan. Untuk lebih jauh lagi, keterlibatan user bukan hanya dari satu tingkatan pada suatu organisasi, melainkan beberapa tingkatan organisasi sehingga informasi yang dibutuhkan untuk masingmasing *user* dapat terpenuhi dengan baik.

2. *Design Workshop* (Proses Perancangan)

Tahap ini melibatkan proses perancangan sistem dan perbaikan desain berdasarkan masukan dari pengguna. Interaksi aktif antara *user* dan *analyst* adalah kunci sukses pada tahap ini, sehingga dialog aktif antara pengguna dan analis adalah kunci keberhasilan.

3. *Implementation* (Penerapan)

Pada tahap ini, *programmer* mengembangkan program berdasarkan desain yang telah disetujui oleh *user* dan *analyst*. Setelah pengembangan selesai, program diuji untuk memastikan tidak ada

kesalahan sebelum diaplikasikan. Pada saat ini, pengguna memberikan tanggapan terhadap sistem yang telah dibuat dan menyetujui sistem tersebut [20].

### **2.2.5 PHP**

PHP adalah bahasa pemrograman untuk memanipulasi data pada website, khususnya dalam konteks Internet. PHP dapat diartikan sebagai singkatan dari *Hypertext Preprocessor*, yang merupakan sebuah bahasa pemrograman web *server-side* yang bersifat open source dan tersedia tanpa biaya. PHP berfungsi sebagai skrip yang terintegrasi dalam kode HTML dan dijalankan di sisi *server web* [21].

### **2.2.6 Laravel**

Laravel adalah suatu kerangka kerja yang digunakan untuk membangun aplikasi web dengan menggunakan PHP sebagai bahasa pemrograman. *Framework* ini memiliki popularitas tinggi dalam pengembangan aplikasi berbasis *back end* menggunakan PHP. Laravel menawarkan sejumlah fitur yang sangat mendukung percepatan proses pembuatan aplikasi dan sistem informasi berbasis web. Penggunaan Laravel bersifat fleksibel dan dapat digunakan sebagai backend, frontend, atau sekedar backend [22].

### **2.2.7 Visual Studio Code**

*Visual Studio Code* adalah sebuah aplikasi teks editor yang memiliki bobot ringan dan kinerja yang handal. Aplikasi ini dikembangkan oleh Microsoft dan tersedia untuk berbagai sistem operasi seperti *Linux*, *Mac*, dan *Windows*. *Visual Studio Code* mendukung bahasa pemrograman seperti *Javascript*, *Typescript*, dan *Node.js* secara langsung, dan juga memungkinkan pengguna untuk mengakses banyak bahasa pemrograman lainnya melalui penggunaan *plugin* yang dapat diunduh dan diinstal melalui

*marketplace Visual Studio Code*, termasuk C++, C#, Python, Go, Java, PHP, dan sejumlah bahasa lainnya [23].

### **2.2.8 Unit Testing**

*Unit testing* merupakan sebuah metode yang digunakan untuk melakukan pengujian pada kode program perangkat lunak dengan maksud memverifikasi keefektifan dan mendeteksi kesalahan. Dalam beberapa situasi dalam pengembangan perangkat lunak, seringkali terjadi bahwa kode program yang telah dibuat tidak efisien atau bahkan tidak digunakan sama sekali. Melalui *unit testing*, kita dapat mengidentifikasi kode program yang kurang efisien dan mengukur tingkat efektivitasnya dalam pengembangan perangkat lunak. Unit testing juga digunakan untuk mendeteksi kesalahan dan cacat yang mungkin tidak terlihat saat program dijalankan [24].

### **2.2.9 UML**



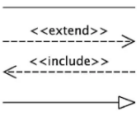
UML (Unified Modeling Language) adalah model atau alat yang digunakan dalam desain pengembangan perangkat lunak yang fokus pada pemrograman berorientasi objek. UML juga menyediakan panduan standar dalam membuat gambaran rinci tentang suatu sistem, yang mencakup aspek-aspek seperti konsep bisnis, spesifikasi kelas dalam bahasa pemrograman tertentu, desain skema *database*, serta komponen yang diperlukan dalam pengembangan perangkat lunak tersebut [25]. Berikut adalah jenis UML :

#### **2.2.9.1 Use Case Diagram**

Sebuah diagram yang menggambarkan interaksi umum antara pengguna dan suatu sistem adalah diagram kasus pengguna. Diagram ini menciptakan gambaran tentang bagaimana suatu sistem digunakan melalui narasi yang menggambarkan interaksi antara pengguna dan sistem tersebut. Diagram kasus pengguna ini terdiri dari elemen-elemen seperti aktor dan interaksi yang dilakukan oleh mereka. Aktor dapat berupa manusia,

perangkat keras, sistem lain, atau entitas lain yang terlibat dalam interaksi dengan sistem.

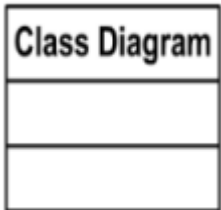
Tabel 2. 2 Komponen *Use Case Diagram* [26].



Simbol	Nama	Keterangan
	Aktor	Mewakili peran orang yang berkomunikasi dengan sistem
	<i>Use Case</i>	<i>Abstraksi</i> interaksi sistem dan aktor
	Relasi	Hubungan antar usecase

### 2.2.9.2 *Class Diagram*

*Class diagram* merupakan representasi visual dari struktur sistem yang berfokus pada pendefinisian kelas-kelas yang diperlukan untuk membangun sistem. Setiap kelas dalam diagram memiliki atribut dan metode atau operasi yang mendefinisikan perilaku dan sifat dari objek dalam sistem tersebut. Simbol-simbol yang digunakan dalam *Class Diagram* memberikan gambaran tentang struktur kelas, atribut, dan hubungan antar kelas dalam sistem secara intuitif dan sistematis.

Tabel 2. 3 Komponen *Class Diagram* [27].







Simbol	Nama	Keterangan
	<i>Object</i>	Blok untuk pembangun program, bagian atas <i>class</i> , bagian Tengah atribut dari <i>class</i> , dan bagian bawah adalah <i>method</i> dari sebuah <i>class</i>

	<i>Activition</i>	Hubungan antar <i>class</i>
	<i>Lifeline</i>	Ketergantungan satu <i>class</i> dengan yang lain

### 2.2.9.3 Sequence Diagram

*Sequence diagram* digunakan untuk mengilustrasikan interaksi antar objek di dalam dan di sekitar sistem dengan merepresentasikan pesan yang dijelaskan sepanjang waktu. Diagram urutan terdiri dari dimensi vertikal yang mencakup waktu dan dimensi horizontal yang mencakup objek-objek yang terlibat dalam interaksi tersebut [28].







Tabel 2. 4 Komponen *Sequence Diagram* [29].

Nama	Simbol	Fungsi
<i>Object</i>		Merepresentasikan sebuah <i>class</i> atau objek
<i>Activity Boxes</i>		Panjang waktu yang dibutuhkan objek
<i>Actors</i>		Pengguna yang berinteraksi dengan sistem
<i>Lifeline</i>		Kehidupan suatu objek
<i>Message</i>		Interaksi antar objek
<i>Message to Self</i>		Reaksi antar objek

### 2.2.9.4 Activity Diagram

*Activity diagram* adalah sebuah jenis diagram yang digunakan dalam untuk menggambarkan alur kerja dari suatu sistem atau proses. Diagram ini terdiri dari beberapa elemen utama yang diintegrasikan ke dalam node dan edge, yang masing-masing mewakili berbagai komponen dan aliran dalam sebuah aktivitas.

Tabel 2. 5 Komponen *Activity Diagram*[30]

Nama	Simbol	Fungsi
<i>Start</i>		Titik awal keadaan atau awal dari kegiatan
<i>Activity</i>		Mewakili kegiatan atau proses
<i>Control Flow</i>		Aliran kontrol dari aksi ke aksi lain
<i>Guard(kondisi)</i>		Kondisi jalur setelah <i>decision</i>
<i>Message</i>		Interaksi antar objek
<i>Activity Final Mode</i>		Akhir dari semua aliran kontrol