

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan tinjauan pustaka dari penelitian-penelitian yang telah dilakukan sebagai penunjang yang akan dilakukan oleh penulis. Beberapa tinjauan literatur dari penelitian yang telah dilakukan sebelumnya dapat dilihat pada Tabel 2.1 Daftar Literatur.

Tabel 2. 1 Daftar Literatur

No. 1	Rosyada (2021)
Judul	Sistem Keamanan Web Service menggunakan JSON Web Token untuk Mengukur Authorization dan Authentication dengan Algoritma SHA 512
Jurnal	Jurnal Universitas Sriwijaya
Volume dan Halaman	29
Tahun	2021
Penulis	Amrina Rosyada
Identifikasi Masalah	Permasalahan yang menjadi usulan penelitian ini adalah algoritma mana yang lebih unggul dalam mengamankan sebuah web service yang mencakup data serta seluruh komunikasi untuk melindungi kerahasiaan dan integrasi data
Metode/Tools	Menggunakan HMAC-SHA-512, yang mana <i>token string</i> yang dihasilkan pada HMAC- 512 lebih panjang
Tujuan dan Hasil Penelitian	Penelitian tersebut memberikan hasil dari perbandingan Antara algoritma HMAC SHA-256 dengan HMAC SHA-512. Dari paparan hasilnya didapat kinerja HMAC SHA-512 lebih baik dalam mengamankan Web Service yang dibangun

No. 2	Gunawan (2019)
Judul	JSON Web Token (JWT) untuk Autentikasi pada Interoperabilitas Arsitektur berbasis RESTful Web Service
Jurnal	Jurnal Edukasi dan Penelitian Informatika (JEPIN)
Volume dan Halaman	5. 1
Tahun	2019
Penulis	Rohmat Gunawan dan Alam Rahmatulloh
Identifikasi Masalah	Permasalahan yang menjadi usulan penelitian ini adalah bagaimana cara untuk menerapkan JWT untuk Authorisasi pada platform yang berbeda.
Metode/Tools	Menerapkan RESTful dengan JWT pada arsitektur sistem Interoperabilitas
Tujuan dan Hasil Penelitian	Penelitian tersebut berhasil membuat arsitektur multi-platform dengan memanfaatkan teknologi web service berbasis RESTful dan menerapkan JWT untuk otentikasi sistem yang dapat diakses dari platform yang berbeda
No. 3	Rahmatulloh, Sulastrri and Nugroho (2018)
Judul	Keamanan RESTful Web Service Menggunakan JWT (JSON Web Token) HMAC-SHA 512
Jurnal	JNTETI
Volume dan Halaman	7. 2
Tahun	2018
Penulis	Alam Rahmatulloh, Heni Sulastrri dan Rizal Nugroho

Identifikasi Masalah	Permasalahan yang menjadi usulan penelitian ini adalah bagaimana hasil perbandingan dari SHA-256 dan SHA 512.
Metode/Tools	Menerapkan penerapan algoritme SHA-256 dan SHA-512 pada arsitektur intel 64-bit
Tujuan dan Hasil Penelitian	Penelitian ini berhasil menerapkan algoritme SHA-256 dan SHA-512 pada arsitektur intel 64-bit menghasilkan kinerja SHA-512 50% lebih baik dibandingkan dengan SHA-256
No. 4	Rajagukguk (2018)
Judul	Penggunaan Kriptografi JWT dalam Mengimplementasi Keamanan API.
Jurnal	Jurnal STEI Institut Teknologi Bandung
Volume dan Halaman	5
Tahun	2018
Penulis	Rio Chandra Rajagukguk
Identifikasi Masalah	Permasalahan yang menjadi usulan penelitian ini adalah bagaimana algoritma hash dapat menjadi metode keamanan pada sebuah REST API.
Metode/Tools	Implementasi JWT pada server dengan skala eksperimen.
Tujuan dan Hasil Penelitian	Hasil penelitian yang telah dilakukan adalah memberikan penjelasan seperti apa bentuk JWT yang diimplementasi pada sebuah server dalam skala eksperimen
No. 5	Wijaya, Jacobus and Sambul (2021)

Judul	Penggunaan Kriptografi JWT dalam Mengimplementasi Keamanan API.
Jurnal	Jurnal Teknik Informatika UNSRAT
Volume dan Halaman	16. 4
Tahun	2021
Penulis	Febyana Wijaya, Agustinus Jacobus dan Alwin Sambul
Identifikasi Masalah	Permasalahan yang menjadi usulan penelitian ini adalah bagaimana perancangan serta implementasi web service pada sistem perpustakaan.
Metode/Tools	Menerapkan Web Service RESTful API dan menggunakan JWT untuk keamanannya.
Tujuan dan Hasil Penelitian	Penelitian tersebut berhasil membuat sebuah sistem perpustakaan yang mampu mengintegrasikan data dengan sistem yang lain dengan menerapkan Web Service RESTful API dan menggunakan JWT untuk keamanannya

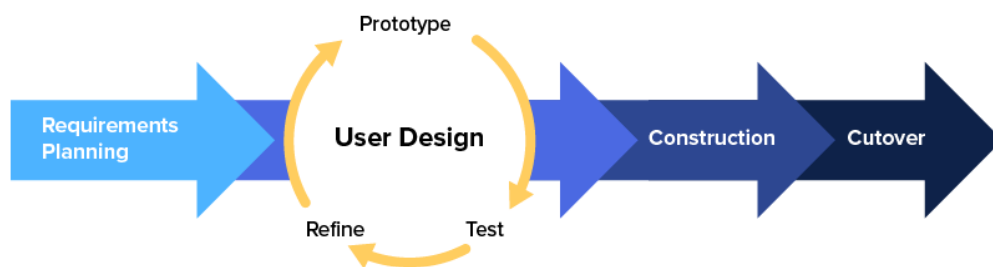
Dari beberapa tinjauan yang telah dijelaskan, dapat disimpulkan bahwa *JSON Web Token* terbukti bisa mengamankan sebuah *web service*. Untuk membedakan penelitian ini dengan penelitian yang sebelumnya, penelitian ini akan menerapkan *refreshing token authentication* pada JWT sebagai metode pengamanan yang lebih optimal (Dykstra, Altunay and Teheran, 2021).

2.2 Metode Pengembangan

2.2.1 *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah metode pengembangan perangkat lunak yang memiliki sifat *incremental*. RAD menekankan pada siklus pengembangan berdasarkan pembuatan *prototype*, iterasi (berulang) dan *feedback* yang berulang-ulang. Metode ini memungkinkan proses pengembangan yang cukup cepat. RAD lebih mengacu kepada apa yang diinginkan oleh penggunanya (Sagala, 2018).

Adapun ilustrasi dari metode pengembangan RAD yang ditunjukkan pada Gambar 2.1



Gambar 2. 1 Metode Pengembangan RAD (Sagala, 2018)

Penjelasan langkah-langkah dalam metode pengembangan ini adalah sebagai berikut :

1. *Requirements Planning*

Pada tahapan ini pengembang dan pengguna bertemu secara langsung untuk menentukan apa saja kriteria yang dibutuhkan untuk sistem yang akan dikembangkan. Kebutuhan yang ditentukan tidak perlu bersifat spesifik, hanya saja bersifat umum dan jumlahnya bisa banyak. Setelah kebutuhan

didapatkan, pengembang akan menentukan kebutuhan apa saja yang perlu diprioritaskan.

2. *User Design*

Tahap ini bertujuan untuk memberikan gambaran dan mengecek apakah sistem yang dikembangkan sudah sesuai dengan kebutuhan klien atau belum.

Tahap ini bisa dilakukan berulang-ulang dan melibatkan pengguna untuk melakukan uji coba dan memberikan *feedback*.

Pada tahap ini ada 3 objek yang menjadi fokus utamanya, diantaranya adalah sebagai berikut:

- a. *Prototype* : Rancangan atau gambaran dari sistem yang akan dikembangkan
- b. *Test* : Uji coba apakah rancangan sudah sesuai atau belum.
- c. *Refine* : Penambahan atau perubahan dari sistem sesuai dengan apa yang diinginkan pengguna.

3. *Construction*

Tahap ini melakukan proses pengembangan serta pengumpulan *feedback*. Setelah mengetahui seperti apa sistem yang ingin dikembangkan, pengembang mengubah *prototype* ke bentuk sistem versi beta sampai dengan status final.

4. *Cutover*

Tahapan ini merupakan langkah terakhir yang dilakukan sebelum menyerahkan sistem ke klien. Sebelum sistem diserahkan, pengembang

terus melakukan optimalisasi dalam memperbaiki interface, menguji stabilitas, hingga melakukan maintenance dan menyusun dokumentasi

2.3 Access Token

Access token merupakan sebuah gabungan string unik yang berupa angka, huruf dan simbol. *Access token* adalah sebuah kunci yang dibutuhkan oleh pengirim layanan yang harus digunakan untuk memverifikasi apakah pengirim memiliki hak akses dalam web service. Masing-masing pemilik akun pada layanan web service memiliki sebuah access token yang bersifat unik. Access token ini berfungsi untuk menambah tingkat keamanan pada web service, sehingga yang dapat mengaksesnya yang memiliki hak akses saja (Tedyyana, Fauzi and Ratnawati, 2021).

2.4 Refresh Token

Refresh token merupakan token yang digunakan sebagai validasi serta menjadi token yang akan dikirim untuk mendapatkan access token yang baru. Ketika masa aktif refresh token masih valid server akan terus mengubah access token dan memvalidasinya (Dykstra, Altunay and Teheran, 2021). Sama seperti access token bentuk token ini berupa string dengan 3 bagian yaitu header, payload dan signature key.

2.5 Django

Django adalah web framework Python yang didesain untuk membuat aplikasi web yang dinamis. Django membantu membuat website dengan berbagai fiturnya untuk mencegah para developer menulis kode python yang sama berulang kali. Django memiliki sistem ORM (*Object relational Model*) untuk mengolah

data, sistem *Views* mengatur logika dan sistem templating di bagian tampilannya (Wijaya, Jacobus and Sambul, 2021).

2.6 *JavaScript Object Notation (JSON)*

Javascript Object Notation (JSON) adalah sebuah format data yang digunakan untuk pertukaran dan penyimpanan data antara server dengan client. Filenya hanya memuatkan teks dan berekstesikan *.json*, format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan bahasa yang umum digunakan oleh programmer. Oleh karena itu, JSON ideal sebagai bahasa pertukaran data (Rosyada, 2021).

2.7 *JSON Web Token (JWT)*

JSON Web Token (JWT) adalah standar terbuka (RFC 7519) yang mendefinisikan cara yang ringkas dan mandiri untuk mentransmisikan informasi antar pihak secara aman sebagai objek JSON. Informasi ini dapat diverifikasi dan dipercaya karena ditandatangani secara digital. JWT dapat ditandatangani menggunakan *secret key* (dengan algoritma HMAC) atau pasangan kunci publik/pribadi menggunakan RSA atau ECDSA (Peyrott, 2022).

Dalam bentuknya yang ringkas, Token Web JSON terdiri dari tiga bagian yang dipisahkan oleh titik (*.*), yaitu *Header*, *Payload* dan *Signature*. Oleh karena itu, JWT biasanya terlihat dalam format *XXXX.YYYY.ZZZZ* yang berupa *rondom string*.

1. *Header*

Header biasanya terdiri dari dua bagian yakni tipe dari token misalnya JWT atau Hawk dan algoritma *signing* seperti HMAC SHA 512 atau RSA.

2. *Payload*

Bagian kedua dari token adalah *payload*, yang berisi klaim. Klaim adalah pernyataan tentang suatu entitas (biasanya data pengguna) dan data tambahan. Ada tiga jenis klaim yakni *registered*, *public*, dan *private claims*.

3. *Signature*

Signature merupakan gabungan dari bagian header yang di-*encode*, payload yang di-*encode*, *secret key*, algoritma yang ditentukan di *header*, dan semua bagian tersebut di enkripsikan.

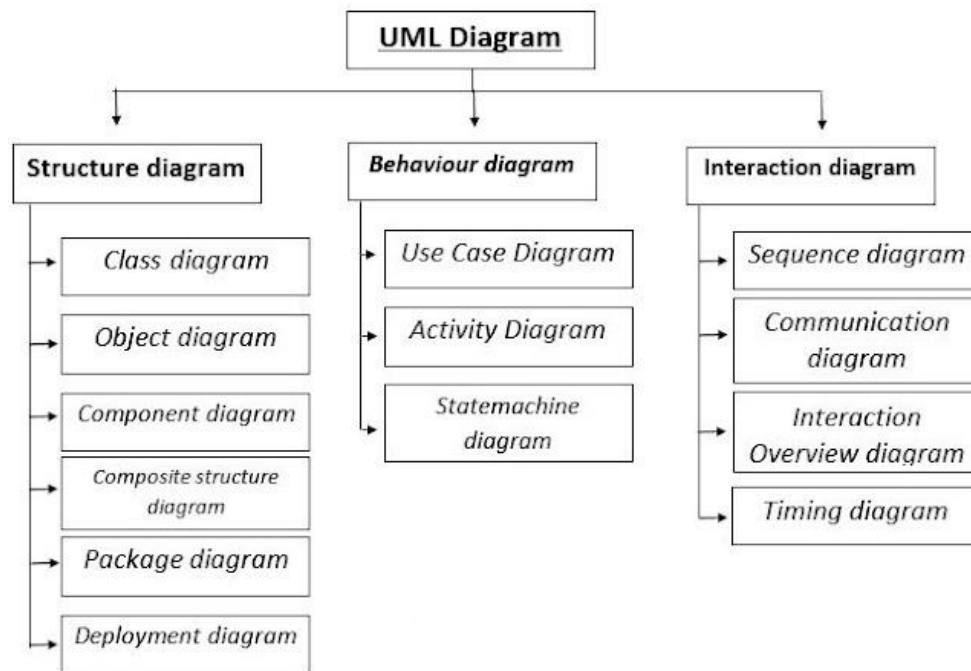
2.8 Uniform Resource Locator (URL)

URL merupakan rangkaian atau arahan dari sebuah web yang tertera yang berisikan nama folder, protocol bahasa dan yang lain sebagainya (Tedyyana, Fauzi and Ratnawati, 2021). URL ini digunakan untuk mengarahkan pengguna menuju ke sebuah website spesifik yang ingin dituju.

2.9 Visual Studio Code

Visual Studio Code merupakan salah satu *code editor* yang digunakan untuk mengembangkan serta *debugging* sebuah sistem web atau *cloud application*. Visual Studio Code ini telah mendukung pengembangan web yang menggunakan JavaScript, TypeScript dan Node.js. Bahasa pemrograman yang dapat digunakan juga telah mendukung C++, C#, Java, Python, PHP dan Go (Agustini and Kurniawan, 2019).

2.10 Unified Modelling Language (UML)



Gambar 2. 2 Jenis-jenis UML (Rosa and Shalahuddin, 2018)

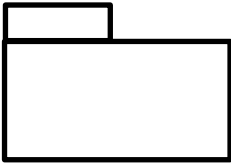
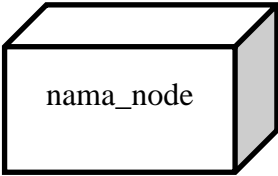


Unified Modelling Language (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML digambarkan sebagai *real time system* yang mana memiliki kepentingan lebih dalam membangun model konseptual yang kemudian akan berproses secara bertahap. UML pada dasarnya dikelompokkan ke dalam diagram struktural dari kelompok yang sering digunakan dalam merancang suatu sistem, diantaranya yaitu *Deployment Diagram*, *Activity Diagram* dan *Communication Diagram* (Rosa and Shalahuddin, 2018).

1. *Deployment Diagram*

Deployment diagram adalah proses penggambaran yang dilakukan untuk menunjukkan konfigurasi antara komponen dalam proses eksekusi

aplikasi. *Diagram* ini dapat digunakan untuk memodelkan sistem *server* ataupun *client* (Rosa and Shalahuddin, 2018). Adapun symbol-simbol dari *Deployment Diagram* ini dapat dilihat pada Tabel 2.2



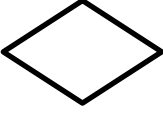


Tabel 2. 2 *Deployment Diagram*

No	Simbol	Nama	Keterangan
1		Package	Package merupakan sebuah bungkusan dari satu atau lebih node.
2		Node	Biasa mengacu pada perangkat keras atau perangkat lunak jika di dalam node disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikut sertakan harus sesuai dengan komponen yang telah didefinisikan pada diagram komponen.
3		Link	Relasi antar node.
4		Dependency	Kebergantungan antar node, arah node mengarah pada node yang dipakai.

2. Activity Diagram

Activity diagram merupakan diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertical.

Tabel 2. 3 *Activity Diagram*

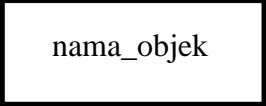


No	Simbol	Keterangan
1		Status awal aktivitas dari sistem.
2		Aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.
3		Percabangan (Decision) merupakan asosiasi percabangan dimana terdapat pilihan aktivitas yang lebih dari satu.
4		Penggabungan (Join) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir dari sistem.

3. Communication Diagram

Communication Diagram atau Diagram Komunikasi menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi mempresentasikan informasi yang diperoleh dari Class

Diagram, Sequence Diagram, dan Usecase Diagram untuk mendeskripsikan gabungan antara struktur statis atau tingkah laku dinamis dari suatu sistem.

Tabel 2. 4 *Communication Diagram*

No	Simbol	Keterangan
1	Objek 	Objek yang melakukan interaksi pesan
2	<i>Link</i> 	Relasi antar objek yang menggabungkan objek satu dengan yang lainnya atau dengan dirinya sendiri
3	Arah Pesan / <i>stimulus</i> 	Arah pesan yang terjadi, jika pada link ada dua arah pesan yang berbeda maka arah digambarkan dua arah pada dua sisi link