

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Putra (2020), telah melakukan penelitian tentang *RESTful API* Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices untuk ditampilkan di *frontend*. Aplikasi bertujuan untuk membangun sebuah sistem yang lebih dinamis dan berkembang dengan memanfaatkan teknologi Arsitektur Microservice. Dibuat dengan Bahasa pemrograman Python, menggunakan database PostgreSQL.

Doglio (2014) menuliskan Pro REST API Development with Node.js. Beliau menjelaskan bagaimana cara mengembangkan REST API pada sebuah aplikasi dengan menggunakan Bahasa pemrograman Node JS serta menggunakan MongoDB.

Dudhe dan Sharekar (2014), telah melakukan penelitian tentang Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment. Mereka menganalisis kinerja SOAP dan RESTful pada Mobile Web Service dengan Apache Tomcat Web Server, dari kedua pengujian diatas RESTful terbukti lebih stabil dan cepat waktu responnya.

Moroney (2017) menulis The definitive guide to Firebase. Beliau menjelaskan fitur-fitur yang ada pada Google Firebase, salah satunya yaitu Authentication dengan metode-metode yang disediakan oleh Firebase serta mengimplementasikan pada bahasa pemrograman JAVA.

Rahmatulloh, dkk. (2018) telah melakukan penelitian tentang Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. Mereka menganalisis seberapa cepat kinerja JSON Web Token serta keamanan pada pertukaran data RESTful Web Service.

Pembahasan yang dibuat kali ini tentang "IMPLEMENTASI KEAMANAN *RESTFUL API* DENGAN *FIREBASE AUTHENTICATION* PADA APLIKASI PRESENSI". Penelitian ini bertujuan untuk menghasilkan sebuah keamanan pada *RESTful API* dengan JSON Web Token yang dihasilkan oleh *Firebase Authentication* dan digunakan untuk pertukaran data yang sah pada aplikasi presensi C.V Amigo Mangesti Utomo. Dibuat dengan Bahasa pemrograman PHP, menggunakan database MySQL.

Tabel 2.1 tinjauan pustaka merupakan tabel yang dibuat untuk mendefinisikan penelitian yang sebelumnya hampir sama dilakukan dengan penelitian yang diajukan saat ini, adapun perbandingan yang menjadi tabel tinjauan pustaka penelitian yakni :

Table 2.1. Tinjauan Pustaka

Penulis	Judul	Bahasa Pemrograman	Teknologi	Database
Putra A (2019)	<i>RESTFUL API</i> Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices	Python, Javascript	REST API	PostgreSQL
Doglio, F (2014)	Pro REST API Development with Node.js	Javascript	REST API	NoSQL-MongoDB
Dudhe, A., dan Sharekar, S. S. (2014)	Performance Analysis of SOAP and RESTful Mobile Web Services in Cloud Environment	Java	REST API	Google Cloud SQL
Moroney, Laurence. (2017)	The definitive guide to Firebase	Java		Firebase Database
Rahmatuloh, Alam, Heni Sulastri, and Rizal Nugroho. (2018)	Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512.	Javascript Ajax	RESTful Web Service	Mysql

## 2.2 Dasar Teori

### 2.2.1 *RESTful API*

*RESTful API* didasarkan pada teknologi *state transfer* (representational state transfer / REST), gaya arsitektur dan pendekatan komunikasi yang sering digunakan dalam pengembangan layanan web, Gambar 2.1 adalah contoh arsitektur *RESTful API*.

Meskipun REST dapat digunakan di hampir semua protokol, tapi biasanya memanfaatkan HTTP ketika digunakan untuk Web API. Hal ini membantu pengembang web tidak perlu menginstal library atau perangkat lunak tambahan untuk memanfaatkan desain REST API. Design REST API pertama kali diperkenalkan oleh Dr. Roy Fielding dalam disertasi doktor tahun 2000-nya. REST API terkenal karena fleksibilitasnya yang luar biasa. Data tidak terikat dengan metode dan sumber daya, REST memiliki kemampuan untuk menangani beberapa jenis panggilan, mengembalikan format data yang berbeda dan bahkan mengubah secara struktural tentunya dengan implementasi yang benar.

REST yang digunakan oleh browser dapat dianggap sebagai bahasa internet. Dengan meningkatnya penggunaan cloud, API muncul untuk mengekspos layanan web. REST adalah pilihan logis untuk membangun API yang memungkinkan pengguna untuk terhubung dan berinteraksi dengan layanan cloud. API telah banyak digunakan oleh situs-situs seperti Amazon, Google, LinkedIn dan Twitter (Yudana, 2019).



**Gambar 2.1** *Arsitektur RESTful API*

### 2.2.2 *Firebase Authentication*

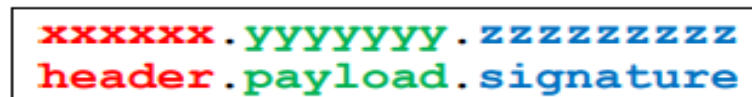
*Firebase Authentication* adalah salah satu layanan back-end, fitur Android dan iOS, SDK yang mudah digunakan, dan tampilan interfaces yang siap pakai untuk mengautentikasi pengguna ke aplikasi yang akan dibuat. *Firebase Authentication* mendukung autentikasi menggunakan nomor telepon, sandi, penyedia identitas gabungan populer seperti Google, Facebook, dan sebagainya (Dicoding, 2020).

Tidak semua pengguna aplikasi akan memberikan kredensial atau data diri secara sukarela, ini berpotensi menyebabkan pengguna aplikasi ragu dan menolak melanjutkan penggunaan aplikasi tersebut. Maka dari itu *Firebase Authentication* merangkum semua agar pengguna dapat lebih mudah melakukan sign-in dengan beberapa provider yang disediakan oleh Firebase (Moroney, 2017).

### 2.2.3 **JSON Web Token**

JSON Web Token (JWT) adalah sebuah token berbentuk string JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Karena bentuknya kecil, token JWT dapat dikirim melalui URL, parameter HTTP POST atau di dalam

Header HTTP, dan juga karena ukurannya yang kecil maka dapat ditransmisikan dengan lebih cepat. Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu query ke basis data lebih dari satu kali. Token tersebut dapat diverifikasi dan dipercaya karena sudah di-sign secara digital. Token JWT dapat di-sign dengan menggunakan secret (algoritma HMAC) atau pasangan public/private key (algoritma RSA). Proses login yang dilakukan tidak seperti aplikasi website biasa, tetapi menggunakan session untuk mengingat yang sedang melakukan proses login. JWT tidak bergantung pada bahasa program tertentu. Struktur JWT terdiri atas tiga bagian yang dipisahkan oleh titik (“.”) seperti Gambar 2.2. Tiga bagian tersebut antara lain header, payload, dan signature (Rahmatulloh, *dkk.*, 2018).



**Gambar 2.2** Struktur JSON Web Token

#### 2.2.4 MySQL

Menurut Arief (2011) “MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya”.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP.

### 2.2.5 Middleware

Middleware adalah salah satu fitur dari Laravel yang menyediakan mekanisme penyaringan HTTP request yang masuk ke aplikasi anda atau dengan kata lain setiap kali ada request yang masuk maka akan difilter oleh Middleware. Sebagai contoh, Laravel telah menyediakan sebuah middleware untuk memverifikasi setiap user yang melakukan autentikasi pada aplikasi. Jika user tersebut gagal melakukan *Authentication* maka akan dialihkan ke halaman login. Namun jika user berhasil melakukan *Authentication*, maka middleware akan mengizinkan untuk melakukan request selanjutnya pada aplikasi tersebut (Codepolitan, 2017).

### 2.2.6 Postman

Postman adalah sebuah aplikasi yang berfungsi sebagai *REST CLIENT* untuk uji coba *REST API*. Postman biasa digunakan oleh developer pembuat *API* sebagai tools untuk menguji *API* yang telah mereka buat (Antares, 2020). Postman sendiri memiliki fitur yang dapat digunakan secara individu ataupun berkelompok (team), dapat juga digunakan secara gratis ataupun berbayar. Postman juga dapat digunakan untuk mengumpulkan *API* yang dapat dibuat menjadi sebuah dokumentasi utuh untuk satu projek tertentu. Jika dokumentasi *API* dibuat lengkap dengan memanfaatkan Postman akan mempermudah dalam proses pengembangan projek, karena setiap developer bisa memiliki acuan yang jelas untuk penggunaan setiap *API* (Refactory, 2021).