

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Papon.id adalah *startup* usaha sosial yang mengembangkan permainan untuk memaksimalkan potensi dan promosi pariwisata daerah di Indonesia. Selain mengembangkan permainan, Papon.id juga mengembangkan situs web dan aplikasi perangkat bergerak (*mobile*) sebagai sarana penyebaran informasi tempat wisata, UMKM lokal beserta produknya, acara di lokasi setempat, dan informasi pendukung lainnya (Papon.id, 2021).

Dalam pengembangan sistemnya, Papon.id mengadopsi gaya arsitektur *Representational State Transfer* (REST). Gaya arsitektur ini memiliki enam batasan formal, dimana lima diantaranya adalah batasan wajib, dan satu lainnya adalah batasan opsional. Batasan wajib itu meliputi penggunaan arsitektur klien-server, sifat *statelessness*, *cacheability*, sistem berlapis, dan antarmuka seragam. Sedangkan batasan opsional dari REST adalah *code on demand* (Fielding, 2000). Papon.id sendiri mengadopsi REST dikarenakan sifatnya yang mudah diskalakan, upaya memisahkan perhatian antar komponen dalam sistem untuk membagi kerja tim pengembang, dan sifat portabilitas yang memudahkan dalam pengembangan sistem Papon.id.

Karena mempergunakan arsitektur REST, maka Papon.id memiliki klien dan server yang terdedikasi, dan saling berkomunikasi menggunakan standar protokol HTTP yang telah ditetapkan. Klien yang dimaksudkan disini adalah semua perangkat atau aplikasi yang mengonsumsi sumber daya dari server, sedangkan server adalah penyedia layanan untuk pihak yang memerlukannya.

Untuk kasus yang terjadi di Papon.id, kliennya adalah laman web dan aplikasi perangkat bergerak (*mobile*); sedangkan servernya merupakan sebuah perangkat lunak, yang mengekspos layanan antarmuka pemrograman aplikasi/*application programming interface* (API). Aplikasi server ini bertugas sebagai gerbang pertukaran data dari dan ke laman web; serta dari dan ke aplikasi *mobile*. Selain itu, server juga bertugas sebagai pusat logika bisnis aplikasi, dan antarmuka data dari dan menuju basis data.

Untuk mengembangkan laman web, pengembang Papon.id menggunakan *framework* Laravel sebagai mesin utama yang menyediakan halaman web; baik untuk menampilkan data ke pengguna, serta sebagai antarmuka bagi administrator untuk menambahkan, mengedit, memperbarui, dan menghapus/*create, read, update, delete* (CRUD) data terkait yang relevan.

Oleh karena Papon.id menerapkan arsitektur klien-server, maka laman web yang menjadi antarmuka utama Papon.id terhadap pengguna akhir dan antarmuka CRUD data bagi administrator tidak terhubung secara langsung ke basis data melainkan harus melewati layanan API yang telah dipersiapkan tim pengembang; sebagai upaya pemisahan perhatian personel tim, dan menjaga konsistensi serta keamanan dalam pemrosesan data dari dan ke basis data *startup* Papon.id.

Layanan API yang dimiliki oleh Papon.id juga didesain dengan memisahkan rute antara permintaan HTTP yang terautentikasi, dengan permintaan yang tidak membutuhkan autentikasi, dimana perbedaan utama dari kedua jenis permintaan ini adalah permintaan terautentikasi membutuhkan *access token* yang valid, sedangkan permintaan yang tidak terautentikasi tidak perlu menambahkan *access token*. *Access token* ini diperoleh dari server API Papon.id ketika pengguna berhasil melakukan autentikasi. Rute yang membutuhkan autentikasi ini disebut juga rute *private*, dan rute yang tidak butuh autentikasi disebut dengan rute *public*.

Masalah mulai muncul ketika administrator dari Papon.id butuh untuk membuat permintaan HTTP ke rute *public* pada layanan API di server dari klien web untuk kebutuhan autentikasi sesi pengguna di web saat proses *login*, dan kemudian dengan bekal *access token* yang didapatkan dari proses *login*, membuat permintaan ke rute *private* untuk melakukan CRUD data pada server. Walaupun Laravel dibekali dengan fitur autentikasi yang sangat canggih, lengkap, dan terstruktur, tetapi fitur tersebut secara *default* hanya menangani autentikasi pengguna menggunakan basis data, berkat *Object-Relational Mapper* (ORM) bawaan Laravel sebagai *user provider*. Selain itu, *user provider* yang tersedia dari pihak ketiga dirasa belum memenuhi kebutuhan untuk dipergunakan dalam proses autentikasi pengguna dan mengakses rute terproteksi alias rute *private* yang mempergunakan *access token* terhadap layanan eksternal, dalam hal ini adalah layanan API Papon.id, dikarenakan desain pustakanya yang justru memerlukan

*access token* terlebih dahulu untuk mendapatkan data pengguna (Mirmahmoudi, 2021), dan justru tidak mendukung mekanisme login yang diperlukan oleh pengguna web Papon.id.

Untungnya, pihak pengembang dari *framework* Laravel sudah merancang arsitektur *framework* dengan sangat baik, sehingga alih-alih menulis ulang keseluruhan logika modul autentikasi yang sudah ada, *framework* Laravel menawarkan solusi yang lebih mudah diimplementasikan, yaitu membuat *custom user provider* yang dapat meniru kinerja *user provider* bawaan, dimana *user provider* yang akan dibuat ini tidak mengautentikasi pengguna ke basis data secara langsung, melainkan ke medium atau perantara penyimpanan data lainnya yang sesuai dengan kebutuhan pengguna *framework*; dalam kasus Papon.id yaitu mengautentikasi pengguna ke server Papon.id melalui kumpulan antarmuka pemrograman aplikasi bergaya arsitektur REST.

Karena desain sistem Papon.id mengharuskan penggunaan *access token* untuk mengakses rute *private*, maka data hasil autentikasi pengguna yang sudah berhasil didapatkan dari proses autentikasi sebelumnya, harus disimpan juga kedalam penyimpanan data sementara yang dimiliki oleh Laravel, dan ketika pengguna butuh untuk melakukan permintaan terautentikasi dari laman web, *framework* Laravel dapat mengikutsertakan *access token* dari pengguna terkait ke permintaan HTTP yang menuju rute *private*.

Berangkat dari permasalahan ini, maka penulis memandang perlunya melakukan riset dan pengembangan sebuah pustaka (*library*) *user provider* berbasis REST API untuk melengkapi *middleware* autentikasi pada *framework* Laravel, dimana *user provider* yang akan dikembangkan ini diharapkan dapat mengatasi permasalahan terkait untuk melakukan autentikasi pengguna di halaman web yang dibangun diatas *framework* Laravel, dan juga dapat melakukan permintaan terautentikasi terhadap layanan web eksternal – dalam kasus ini adalah server Papon.id – tanpa harus membuat ulang unit atau *middleware* autentikasi khusus pada *framework* Laravel.

## 1.2 Perumusan Masalah

Autentikasi merupakan salah satu hal wajib yang diperlukan oleh layanan web sebagai penanda bahwa entitas yang melakukan perubahan data terhadap sistem, adalah benar merupakan entitas yang terdaftar pada sistem, dan diperbolehkan mengakses sumber daya yang disediakan oleh sistem. Jika ada entitas tak terdaftar yang mencoba mengakses sumber daya terlindungi, maka sistem akan menutup akses untuk entitas tersebut, dan mengembalikan galat yang sesuai.

Papon.id sudah cukup berhasil menggunakan *framework* Laravel untuk membantu pengembangan laman web yang tidak memerlukan pengguna untuk masuk ke sistem, alias hanya dipergunakan untuk mengakses rute *public*, tetapi terkendala dalam menyediakan fitur autentikasi terhadap layanan API perusahaan pada laman web; mengakibatkan administrator tidak dapat mengakses sumber daya terlindungi untuk melakukan fungsi CRUD data.

*Middleware* autentikasi pada *framework* Laravel secara *default* hanya melakukan autentikasi pengguna menggunakan *user provider* yang mengecek ke basis data, dan tidak dapat melakukan autentikasi pengguna ke layanan web eksternal, tetapi *middleware* autentikasi pada *framework* Laravel memungkinkan untuk mengembangkan *custom user provider* yang sesuai dengan kebutuhan pengguna, tanpa harus menulis ulang logika autentikasi yang mungkin mempergunakan *middleware* khusus atau cara lainnya, dan belum tentu teruji waktu (*battle tested*) seperti *middleware* autentikasi yang sudah ada didalam kode sumber *framework*.

Berkaca dari latar belakang yang telah dijelaskan di bagian sebelumnya, dan uraian permasalahan singkat, permasalahan yang dihadapi dapat dirumuskan sebagai berikut :

1. Bagaimana alur kerja autentikasi pengguna pada *framework* Laravel?
2. Apa yang menyebabkan pengembang halaman web Papon.id memerlukan *user provider* berbasis REST API pada *middleware* autentikasi Laravel?

3. Apa saja langkah implementasi yang perlu dilakukan untuk membuat serta mengintegrasikan *user provider* berbasis REST API terhadap *middleware* autentikasi bawaan dari *framework* Laravel?
4. Bagaimana cara melakukan pengujian autentikasi pengguna dari laman web berbasis Laravel yang telah mengintegrasikan *user provider* berbasis REST API terhadap layanan web eksternal, dalam hal ini adalah laman web Papon.id terhadap layanan API Papon.id?

### 1.3 Tujuan Penelitian

Adapun tujuan umum yang ingin dicapai dari penelitian ini adalah terciptanya sebuah rancangan paket pustaka generik *user provider* untuk *middleware* autentikasi *framework* Laravel; yang dapat dipergunakan oleh situs web Papon.id, atau situs web apapun berbasis Laravel dengan versi Laravel 8.x dan PHP yang didukung adalah versi 8.0.x, dimana situs web tersebut membutuhkan mekanisme autentikasi pengguna terhadap layanan web eksternal melalui kumpulan antarmuka pemrograman aplikasi/*application programming interface* (API) bergaya arsitektur REST, oleh karena basis data layanan web tersebut tidak dapat diakses secara langsung oleh situs web berbasis Laravel. *User provider* yang dikembangkan ini harus dapat memperluas fitur autentikasi pada Laravel dengan cara hanya mengubah pengaturan untuk memilih *user provider* yang akan dipergunakan pada berkas konfigurasi *framework*, tanpa perlu membangun ulang fitur yang telah ada, misalnya membuat *middleware* autentikasi baru yang belum teruji untuk menggantikan *middleware* autentikasi Laravel yang kokoh dan teruji.

### 1.4 Pembatasan Masalah

Untuk mencegah pembahasan keluar dari koridor yang semestinya, maka penulis memandang perlu untuk menerapkan pembatasan masalah, yang didefinisikan dalam daftar berikut ini:

1. Penelitian ini berfokus kepada bagaimana proses autentikasi pengguna yang terjadi di dalam *framework* Laravel.
2. Hanya menangani autentikasi pengguna berbasis kata sandi yang akan ditukar dengan *access token* dan *refresh token*; dengan kemungkinan

untuk me-*refresh access token* kedaluwarsa menggunakan *refresh token* yang disediakan.

3. Metode interkoneksi layanan web akan menggunakan arsitektur REST (*Representational State Transfer*), dan tidak akan membahas jenis arsitektur interkoneksi sistem lainnya.
4. Penelitian ini tidak akan membahas proses otorisasi pengguna, misalnya: bagaimana sumber daya tertentu hanya dapat diakses oleh pengguna dengan peran/*roles* dan izin/*permission* tertentu.
5. Permintaan HTTP terbatas pada ruang lingkup autentikasi, yaitu mengirim data kredensial pengguna ke server, menerima respons yang sesuai ketika data pengguna berhasil diautentikasi, melakukan *refresh* terhadap *access token*, dan mengakses sumber daya yang dilindungi; terlepas dari apapun peran pengguna yang sudah berhasil diautentikasi.
6. Penelitian ini tidak membahas aspek keamanan transportasi data yang terjadi dalam permintaan HTTP; semua permintaan HTTP yang dilakukan dalam proses autentikasi yang terlibat diasumsikan terjadi dalam keadaan aman.
7. Penelitian ini tidak akan membahas mengenai proses pengembangan sistem ataupun aplikasi terkait, yang memanfaatkan *framework* Laravel dan *middleware* autentikasinya.
8. Paket pustaka user provider yang akan dibangun ini akan mempergunakan *framework* Laravel versi 8.x dengan versi PHP 8.0.x sebagai acuan teknis pengembangan dan pengujian. Paket pustaka ini mungkin tidak bekerja dengan baik diluar dari versi Laravel dan PHP yang dipergunakan.

## 1.5 Sistematika Penulisan

Sebagai upaya untuk membuat penelitian yang terstruktur dan runtut, maka penulis menerapkan beberapa aturan dalam sistematika penulisan skripsi. Adapun sistematika dari penulisan tugas akhir ini disusun dalam lima bab yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian, Bab IV Hasil dan Analisis, yang kemudian akan dirangkum secara umum oleh Bab V Penutup.

**Bab I Pendahuluan** adalah bab yang berisi latar belakang, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan.

**Bab II Tinjauan Pustaka** adalah bab yang berisi definisi dari berbagai istilah yang akan digunakan dalam penelitian ini, yaitu definisi mengenai autentikasi, *Uniform Resource Locator* (URL), *Hypertext Transfer Protocol* (HTTP), autentikasi permintaan HTTP, PHP, *JavaScript Object Notation* (JSON), Composer, Git, Laravel, *middleware*, *user provider*, rekayasa terbalik (*reverse engineering*), *Unified Modelling Language* (UML), dan pengujian perangkat lunak.

**Bab III Metodologi Penelitian** adalah bab yang berisi tentang bahan penelitian, perangkat penelitian, alat bantu penelitian, serta langkah penelitian.

**Bab IV Hasil dan Analisis** adalah bab yang berisi hasil dari penelitian yang sudah didefinisikan di bab sebelumnya. Bab ini akan berisi hasil persiapan, hasil observasi, hasil perancangan, hasil pengembangan, hasil pengujian, dan hasil publikasi.

**Bab V Penutup** adalah bab yang berisi kesimpulan dari penelitian yang telah dilakukan dan saran/rekomendasi untuk perbaikan, pengembangan atau kesempurnaan/kelengkapan penelitian yang telah dilakukan.