

# BAB I PENDAHULUAN

## 1.1 Latar Belakang Masalah

Saat ini, lebih dari 135 juta *website* terhubung melalui *hyperlink*, dan *website* dengan konsep *Electronic Commerce* dapat bersaing dengan perusahaan sejenis untuk meningkatkan kualitas pelayanan. Salah satu inovasi yang dapat dilakukan perusahaan untuk mencapai kesuksesan adalah dengan mengembangkan media *website* yang berkaitan dengan *Electronic Commerce* [1]. Pariwisata merupakan salah satu sektor yang memiliki potensi besar dalam perekonomian suatu negara. Saat ini, dengan adanya teknologi informasi, penggunaan aplikasi *travel booking online* semakin populer di kalangan masyarakat [2] [3].

Perusahaan *Travel Booking* merupakan salah satu perusahaan yang bekerja sama dengan PT Marka Kreasi Persada menyediakan layanan pemesanan perjalanan melalui *platform digital*. Sebelumnya perusahaan *travel booking* memiliki masalah yang sering ditemui yaitu bilamana membutuhkan data *customer* selalu membutuhkan waktu yang lama, hal ini dikarenakan data tersebut belum tersimpan pada *database* tetapi masih berupa dokumen arsip sehingga dapat menghambat kinerja perusahaan.

Menerapkan arsitektur *microservice* dapat memecah sistem menjadi komponen-komponen kecil yang terpisah. Memungkinkan penyimpanan dan akses data yang lebih efisien serta meningkatkan kinerja perusahaan.

Dengan adanya aplikasi *travel booking* perusahaan dapat memudahkan karyawan untuk manajemen data pemesanan *customer* dengan mudah dan cepat. Perusahaan mengharapkan layanan yang baik sehingga performa aplikasi menjadi faktor penting yang menentukan keberhasilan aplikasi.

PT Marka Kreasi Persada merupakan perusahaan Alterra Academy yang melayani kebutuhan bisnis spesialisasi dalam pengembangan, distribusi, dan pemeliharaan perangkat lunak.

Salah satu inisiatif dari Kampus Merdeka adalah Program Riset Independen Bersertifikat, yang memberikan peluang dan pengalaman kepada mahasiswa untuk belajar dan mengembangkan diri melalui kegiatan perkuliahan di luar kampus yang diakui sebagai bagian integral dari kurikulum. Program ini ditujukan untuk mahasiswa yang ingin membangun dan memperdalam keterampilan yang dimiliki, dengan mempelajari kompetensi khusus dan relevan yang dibutuhkan di dunia kerja [4].

Golang adalah sebuah bahasa pemrograman yang dibuat oleh Google dengan dasar bahasa C dan C++. Oleh karena itu, sintaks penulisan bahasa Golang sederhana dan terstruktur. Golang memiliki fitur *garbage collector* yang memastikan keamanan memori dan tipe data string di Golang adalah *UTF-8* yang di-*encode* secara *default*. Sebagai hasilnya, Golang mendukung konkurensi di *level* bahasa dan memungkinkan pengaplikasian konkurensi secara mudah. Selain itu, Golang juga mendukung pemrosesan data dengan banyak prosesor dalam waktu yang bersamaan (*parallel processing*). Kelebihan-kelebihan ini membuat Golang menjadi bahasa pemrograman yang ideal untuk pengembangan *REST API* [4].

*Back-end* merupakan fondasi utama dari suatu aplikasi. Tugas seorang *back-end engineer* adalah menyiapkan data yang akan ditampilkan pada *website* dan menjamin fungsionalitas aplikasi tersebut. Peran *back-end* mungkin tidak dikenal oleh banyak orang dan tidak terhubung langsung dengan pengguna, namun kesuksesan sebuah aplikasi sangat bergantung pada kemampuan *back-end* dalam membangun struktur data di balik *layer* [5].

*Representational State Transfer* (REST) adalah suatu arsitektur layanan web yang berasal dari berbagai gaya arsitektur jaringan yang sering diterapkan dalam layanan web. Arsitektur REST biasanya dioperasikan melalui HTTP (*Hypertext Transfer Protocol*), yang melibatkan proses membaca halaman web tertentu yang berisi file XML atau JSON. Tiap permintaan bersifat mandiri dan *server* tidak menyimpan informasi status permintaan apapun. API yang mengikuti gaya REST disebut RESTful API dan menggunakan URI (*Uniform Resource Identifier*) untuk

merepresentasikan sumber daya. Setiap sumber daya diidentifikasi menggunakan tautan URI. Dalam arsitektur REST, beberapa metode yang umumnya digunakan adalah GET untuk mendapatkan sumber daya, POST untuk membuat sumber daya baru, PUT untuk memperbarui sumber daya yang sudah ada, dan DELETE untuk menghapus sumber daya atau kumpulan sumber daya [6].

*Load Testing* adalah sebuah metode dalam *performance testing* yang bertujuan untuk mengukur *response* sistem dalam berbagai kondisi dan beban. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana perilaku *software* ketika beberapa pengguna mengaksesnya secara bersamaan. *Load Testing* digunakan untuk memeriksa kemampuan sistem yang sedang dikembangkan dalam menangani masalah atau beban yang diujikan dengan menggambarkan situasi yang sesuai dengan keadaan sebenarnya [7].

## 1.2 Perumusan Masalah

Berdasarkan uraian yang dijelaskan pada latar belakang, maka rumusan masalah yang ada di penelitian ini yaitu pengembangan pada *back end travel booking*:

### 1. Pengembangan dan Integrasi Sistem

Arsitektur *microservice* dapat diimplementasikan untuk meningkatkan efisiensi dan efektivitas dalam pengelolaan data *customer* yang sebelumnya masih berbasis dokumen arsip, sehingga dapat mempercepat proses dan meningkatkan kinerja *back end*.

### 2. Perancangan *Database*

Menghasilkan *back end* yang bagus perlu mendesain skema *database* yang optimal untuk aplikasi *travel booking* yang tidak hanya menjamin integritas data dan mengurangi redundansi tetapi juga mendukung efisiensi dalam penyimpanan dan pengambilan data.

### 3. Kinerja dan Skalabilitas

Pengujian beban (*load testing*) dan optimisasi dapat diterapkan untuk memastikan sistem *back end* mampu menangani lonjakan trafik dan operasi secara bersamaan tanpa mengurangi kualitas pelayanan.

#### 4. Pengembangan REST API

Proses yang melibatkan penggunaan bahasa pemrograman untuk memudahkan komunikasi atau pertukaran data antara sistem komputer di *internet*.

### 1.3 Pertanyaan Penelitian

Berdasarkan latar belakang dan perumusan masalah yang telah dijelaskan maka disusun pertanyaan penelitian sebagai berikut:

1. Bagaimana merancang aplikasi *Travel Booking* menerapkan *back-end* menggunakan bahasa pemrograman golang dengan arsitektur *microservice*?
2. Bagaimana hasil pengujian pada *back-end* dengan menggunakan *BlackBox Testing*?
3. Bagaimana hasil pengujian pada *back-end* dengan menggunakan *Load Testing*?

### 1.4 Batasan Masalah

Berdasarkan uraian diatas, maka batasan-batasan masalah dari penelitian sebagai berikut:

1. Aplikasi dikembangkan dengan menggunakan Golang sebagai *back-end*, namun penggunaan bahasa pemrograman *front-end* tidak dibahas dalam penelitian ini.
2. Layanan pada aplikasi ini menyediakan fitur-fitur yang memudahkan pengguna dalam melakukan pemesanan tiket dan reservasi hotel.

### 1.5 Tujuan Penelitian

Terdapat beberapa tujuan penelitian yang ingin dicapai sebagai berikut:

1. Menghasilkan *back end* pada aplikasi *Travel Booking* dengan menerapkan bahasa pemrograman *golang* dengan arsitektur *microservice*.
2. Mengetahui hasil pengujian *unit* pada *back-end* aplikasi *Travel Booking* dengan menggunakan *BlackBox Testing*.
3. Mengetahui hasil pengujian *unit* pada *back-end* aplikasi *Travel Booking* dengan menggunakan *Load Testing*.

## 1.6 Manfaat Penelitian

Berdasarkan tujuan penelitian, maka manfaat penelitian ini adalah menghasilkan aplikasi *back end travel booking online* sistem *booking* yang dibangun akan menjadi lebih efektif dan efisien, sehingga dapat meningkatkan produktivitas dan mengurangi biaya pengembangan aplikasi. Dengan menyediakan fitur-fitur yang memudahkan pengguna dalam melakukan pemesanan tiket dan reservasi hotel, *user experience* pada aplikasi akan meningkat, sehingga dapat meningkatkan kepuasan pengguna dan meningkatkan jumlah pelanggan.