

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Di era modern saat ini, perkembangan teknologi sangatlah pesat membuat orang mengandalkan internet untuk beragam keperluan penting sehari-hari. Salah satunya adalah banyaknya akses website di internet untuk mencari sebuah informasi restoran menggunakan aplikasi web katalog restoran. Aplikasi web katalog restoran sendiri merupakan sebuah aplikasi web yang menampilkan sebuah daftar atau koleksi dari restoran-restoran yang tersedia dalam suatu wilayah atau negara. Informasi yang ditampilkan dalam katalog restoran meliputi nama restoran, alamat, jam buka, menu, harga, foto, ulasan, dan rating. Dalam perkembangannya masih banyak *web app* katalog restoran yang masih dikembangkan secara tradisional yang mempunyai performa yang rendah dan aksesibilitas yang terbatas.

Google, menyebut bahwa sekitar 53% pengguna melakukan *exit* dari *web page* yang *load*-nya lebih dari 8 detik. Padahal, rata-rata homepage memakan waktu 15 detik untuk bisa secara penuh memuat keseluruhan kontennya. Bisa dibayangkan pengguna harus menunggu 15 detik untuk membuka tiap-tiap halaman web yang pengguna akses. Perkembangan *web app* sendiri masih cenderung menggunakan pendekatan monolitik atau mengembangkan aplikasi yang menganggap seluruh aplikasi sebagai satu kesatuan tunggal sehingga akan membingungkan developer nantinya. Maka dari itu diperlukan sebuah struktur untuk mengembangkan *web app* yang dapat diakses, dimodifikasi, dan diatur oleh aplikasi.

Dari permasalahan di atas, dibutuhkan solusi untuk mengatasi masalah pengembangan web yang masih dikembangkan secara tradisional (biasa) dan bersifat monolitik. Maka dari itu *Progressive Web app* dan implementasi dari *Document Object Model* (DOM) menjadi solusi untuk permasalahan yang dimiliki oleh *web app* tradisional. Dengan adanya *Progressive Web app* dan DOM memungkinkan suatu aplikasi web mempunyai performa yang baik dan dapat berjalan lintas platform (*web browser* dan *mobile platform*) serta mempercepat pengembangan aplikasi web.

PWA (*Progressive Web app*) dirancang untuk memberikan pengalaman aplikasi seluler yang sama dengan aplikasi *native*. PWA dapat diinstal di perangkat seluler seperti aplikasi *native* dan dapat ditemukan di layar utama perangkat. PWA juga dapat digunakan *offline* dengan menyimpan konten di *cache*, memberikan pengalaman pengguna yang lebih baik dibanding aplikasi web tradisional.

*Document Object Model* DOM sendiri akan membuat halaman web menjadi sebuah objek yang dapat diakses melalui kode, sehingga membuat pengembangan dari *web app* dan PWA menjadi lebih mudah dan cepat.

Untuk itu pada penelitian kali ini, akan dilakukan sebuah pengujian terhadap solusi *Progressive Web app* untuk permasalahan dari web tradisional dengan cara melakukan pengujian aplikasi web yang telah dibangun sebelumnya secara tradisional (biasa) dengan aplikasi yang telah dibangun menggunakan *Progressive Web app* yang memiliki struktur DOM yang sama, menggunakan implementasi REST API, dan bersifat *responsive* pada implementasi *web app* Katalog restoran.

## 1.2. Rumusan Masalah

Berdasarkan dari latar belakang masalah di atas maka rumusan masalah pada penelitian ini yaitu bagaimana melakukan pengujian antara aplikasi web katalog restoran yang sebelumnya telah dibangun secara tradisional (biasa) dengan aplikasi web yang dibangun menggunakan *Progressive Web app* yang sama-sama memiliki struktur DOM, REST API, dan bersifat *responsive*.

## 1.3. Ruang Lingkup

Dari rumusan masalah di atas untuk memfokuskan penelitian pada pokok permasalahan, maka ruang lingkup dari penelitian akan dibatasi pada beberapa kategori:

1. Penelitian ini akan melakukan pengujian terhadap dua buah aplikasi web yaitu (1) aplikasi katalog restoran yang dibangun secara tradisional (biasa) dan (2) aplikasi katalog restoran yang menggunakan implementasi *Progressive Web apps* (PWA)
2. Pengujian di atas akan menghasilkan rekomendasi manakah dari dua aplikasi web katalog restoran yang lebih baik dan memenuhi syarat dari segi Performa, kemudahan penggunaan (Aksesibilitas), dan *Web Vitals*.
3. Kedua buah aplikasi web katalog restoran dibangun dengan struktur yang sama yakni (1) *Document Object Model* (DOM), (2) menggunakan implementasi REST API, dan (3) bersifat *responsive*.
4. Parameter Pengujian terdiri dari (1) *Web Vitals*, (2) Performa, dan (3) Aksesibilitas, dengan pengujian secara manual untuk pengujian aksesibilitas dan pengujian otomatis menggunakan tools sebagai berikut:

- a. Pengujian Performa menggunakan tools Lighthouse
  - b. Pengujian *Web Vitals* menggunakan tools WebPageTest
5. Kedua buah aplikasi dibuat menggunakan bahasa pemrograman JavaScript, HTML, dan CSS.
  6. Kedua buah aplikasi menggunakan *OPEN API* atau WEB API dari perusahaan Didocing.
  7. Pengujian implementasi *OPEN API* menggunakan tools Postman.

#### **1.4. Tujuan Penelitian**

Tujuan penelitian ini adalah untuk mengetahui hasil pengujian dari pengembangan aplikasi web katalog restoran yang dikembangkan secara tradisional (biasa) dan yang dikembangkan menggunakan *Progressive Web app*, apakah *Progressive Web app* lebih baik dari web tradisional (biasa) dari sisi *Web Vitals*, Performa, dan Aksesibilitas *web app*.

#### **1.5. Manfaat Penelitian**

Dari penjelasan di atas, terdapat beberapa manfaat yang didapat dari penelitian ini adalah mendapatkan rekomendasi mana yang lebih baik antara pengembangan *web app* secara tradisional dengan *web app* berbasis *Progressive Web app* pada pengembangan *web app* katalog restoran dengan struktur DOM, REST API, dan sifat *Responsive* yang sama.

#### **1.6. Sistematika Penulisan**

Sistematika penulisan skripsi dengan judul “Implementasi *Progressive Web app* (PWA) dan REST API dalam Membangun Aplikasi Web Katalog restoran”

disusun guna memberi gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan skripsi ini adalah sebagai berikut:

## **BAB I           PENDAHULUAN**

Permasalahan yang diangkat pada skripsi ini berawal dari adanya masalah pengembangan aplikasi web secara tradisional (biasa) dengan masalah web vital, performa, dan aksesibilitas dalam menggunakan aplikasi web. Pada bagian ini ditentukan juga batasan-batasan meliputi : hasil pengujian dari sebuah aplikasi web katalog restoran yang dibuat secara tradisional (biasa) dan dengan implementasi *Progressive Web apps* (PWA), ditentukan pula bahwa Aplikasi web Katalog restoran dibuat dengan struktur yang sama yakni *Document Object Model* (DOM), menggunakan implementasi REST API, dan bersifat *responsive* dan dibuat menggunakan bahasa pemrograman JavaScript, HTML, dan CSS. Pengujian dilakukan secara manual dan secara otomatis menggunakan tools Lighthouse dan WebPageTest. Ditentukan juga pengujian API menggunakan *OPEN API* atau WEB API dari perusahaan Dicoding dan menggunakan tools Postman. Selain itu aplikasi web ini hanya digunakan oleh pengunjung aplikasi web katalog restoran.

## **BAB II           TINJAUAN PUSTAKA DAN DASAR TEORI**

Berdasarkan literatur yang ditinjau, setidaknya ada 5 literatur yang dapat mendukung pemecahan solusi di atas yaitu Penerapan *Progressive Web apps* pada Aplikasi Lowongan Pekerjaan dengan Teknologi *Service Worker* (Studi Kasus Akakom Career Center), Implementasi *Progressive Web apps* pada Aplikasi Monitoring Service Laptop dengan Teknologi *Service Worker* (Studi Kasus Service Laptop Bengkel OS), *Progressive Web apps* (PWA) untuk Aplikasi Pelacakan

Pengiriman Barang (Studi Kasus Jasa Pengiriman PT.ORIFLAME), Analisis Performa Progressive *Web application* (PWA) pada Perangkat Mobile, Implementasi *Progressive Web apps* Pada Aplikasi Penjualan Bawang Putih, dan Pengembangan Aplikasi Progressive *Web application* (PWA) untuk Pembelajaran dan Evaluasi Kelas English Grammar Online Course. Dasar teori yang disertakan pada penelitian ini adalah teori tentang Proggresive *Web app*, Web Tradisional (1), JavaScript (2), *Service Worker* (2), REST API (3), NodeJS (4), Postman (5), Lighthouse (6), WebPageTest (7), *Web Vitals* (8)

### **BAB III METODE PENELITIAN**

Perancangan solusi (pengujian aplikasi web) akan dilakukan dengan menjabarkan bahan/data, peralatan (*tools*) yang diperlukan dalam pengujian, dan metode pengujian yang digunakan pada aplikasi web seperti Pengujian Web dengan Pengujian Manual, Pengujian otomatis dengan Lighthouse, dan WebPageTest serta Pengujian API dengan Postman.

### **BAB IV IMPLEMENTASI DAN PEMBAHASAN**

Pada bagian ini dibahas hasil pengujian penting untuk menguji kedua aplikasi web katalog web yang telah dibuat sebelumnya yaitu pengujian untuk hasil pengujian web dengan pengujian manual (akan menjadi sub bab 4.1), hasil pengujian web dengan Lighthouse (akan menjadi sub bab 4.2), hasil pengujian web dengan WebPageTest (akan menjadi sub bab 4.3)

Pada bagian ini juga akan dilakukan pengujian aplikasi web dengan menggunakan pengujian postman (akan menjadi sub bab 4.4) untuk menguji *OPEN*

*API*. Pada setiap pengujian akan disertakan analisa/pendapat berkaitan dengan hasil pengujian tersebut.

## **BAB V KESIMPULAN DAN SARAN**

Pada bagian ini akan disimpulkan apakah aplikasi web PWA yang telah diuji berhasil menjadi jawaban dari pengembangan aplikasi web secara tradisional (biasa) untuk memperbaiki masalah web vital, performa, dan aksesibilitas dari aplikasi web serta kenyamanan pengguna. Pada bagian ini akan diinventarisasi apa saja yang menjadi kelemahan-kelemahan sistem yang dibuat khususnya saat dilakukannya pengujian, untuk dijadikan saran perbaikan pengujian aplikasi web khususnya katalog restoran.