

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Teknologi informasi saat ini berkembang sangat pesat sehingga membuat suatu informasi dapat dengan mudah tersebar ke belahan dunia. *Website* merupakan salah satu media informasi yang saat ini menjadi trend bagi institusi, perusahaan, organisasi maupun individu untuk dapat membagikan informasi secara global. Seiring dengan pesatnya perkembangan *website*, semua pengembang *website* berlomba-lomba membangun *website* yang dapat memberikan pengalaman terbaik bagi pengguna. Untuk memberikan pengalaman terbaik bagi pengguna banyak aspek yang dapat dipertimbangkan mulai dari tampilan, kemudahan akses, kecepatan akses dan lain-lain. Untuk dapat mewujudkan hal tersebut diperlukan suatu *website* yang handal dan kuat, dimana bagian yang penting adalah pemilihan arsitektur aplikasi *website* yang tepat. Dengan pemilihan arsitektur yang tepat dapat meningkatkan performansi *website* seperti *performance*, *availability* (ketersediaan), *complexity* dan *cost*. Saat ini arsitektur yang umum digunakan adalah arsitektur *multi-tier* atau *three-tier* yang terbilang tradisional, dimana terdiri dari tiga *layers* yaitu *presentation layers* (di sisi pengguna), *bussiness logic layers* (fungsionalitas dan pemrosesan data) dan *data layers* (penyimpanan data/database). Kelebihan dari arsitektur ini adalah meningkatkan *scalability* (mengatasi lonjakan trafik) namun kekurangannya adalah meningkatkan kompleksitas arsitektur dan membutuhkan banyak biaya.

Saat ini tengah berkembang pendekatan pengembangan web untuk meningkatkan efisiensi dalam proses pengembangan *website*. Jamstack terdiri dari komponen *JavaScript*, *API*, and *Markup* merupakan salah satu arsitektur pengembangan *website* modern untuk membuat *website* lebih cepat, lebih aman dan mudah untuk di skalakan. Terdapat banyak platform yang dapat digunakan pengembang *website* untuk meningkatkan produktivitas pembuatan *website*. Dengan menggunakan Jamstack performa aplikasi web akan lebih cepat karena user bisa menggunakan web seperti aplikasi *native*, keamanan lebih tinggi karena penggunaan API tidak terikat dengan *endpoint backend* dan aksesnya diatur oleh

client, kemudian secara pembiayaan akan lebih murah karena tidak membutuhkan *resource* besar serta dapat menggunakan *Content Delivery Network* (CDN) dalam melakukan *deployment* suatu *website*.

Website Kraton Jogja yang mengelola dan menampilkan konten statis (*static site*) berupa informasi-informasi terkait Kraton Jogja saat ini masih menggunakan desain arsitektur tradisional yaitu menggunakan *single server* sehingga dinilai tidak efektif saat menghadapi lonjakan *traffic* pengguna dan kurang aman karena sistem aplikasi secara tidak langsung terbuka secara publik. Perancangan arsitektur *website* menggunakan Jamstack pada skripsi ini akan diterapkan pada web Kraton Jogja. Desain arsitektur Jamstack pada web Kraton Jogja ini terdiri dari dua bagian yaitu *front-office* dan *back-office*. Pada sisi *front-office* menggunakan platform Netlify untuk deployment web dimana telah diimplementasikan CDN didalamnya, untuk deployment dibutuhkan repository yakni menggunakan Github. Pada sisi *back-office* menggunakan platform Eleventy untuk *Static Site Generator* (SSG) dan Directus untuk manajemen konten *website*. Kemudian akan ada *monitoring* sistem disisi *back-office* yang menggunakan platform *Elasticsearch*, *Logstash*, dan *Kibana* (ELK) sedangkan untuk pengukuran performansi disisi *front-office* menggunakan *software automation testing* yaitu GTMetric. Parameter performansi yang akan diukur adalah *Performance Score*, *Load Time*, *Speed Index*, *Time to Interactive (TTI)* dan *Time to First Byte (TTFB)*. Sehingga atas dasar tersebut, penulis mengambil topik skripsi dengan judul “**Perancangan Dan Analisis Performansi Arsitektur Jamstack Studi Kasus Web Kraton Jogja**”.

1.2 RUMUSAN MASALAH

Rumusan masalah dari penelitian ini adalah:

- 1) Bagaimana merancang arsitektur *website* menggunakan teknologi Jamstack pada *website* Kraton Jogja?
- 2) Bagaimana implementasi arsitektur Jamsatck dalam proses pengembangan *website* Kraton Jogja?
- 3) Bagaimana performansi *website* Kraton Jogja menggunakan teknologi Jamstack?

1.3 BATASAN MASALAH

Batasan masalah dari penelitian ini adalah:

- 1) Teknologi yang digunakan di sisi *front-office* adalah Netlify dan *repository* Github, kemudian di sisi *back-office* menggunakan Directus dan Eleventy.
- 2) Teknologi yang digunakan untuk mengukur performansi *website* disisi *front-office* adalah GTMetric sedangkan untuk *monitoring* disisi *back-office* menggunakan Elasticsearch, Logstash dan Kibana (ELK).
- 3) Parameter monitoring yang akan diambil pada sisi *back-office* adalah *Syslog*, *CPU Usage*, *Memory Usage* dan *Network Usage*.
- 4) Parameter performansi *website* yang diukur adalah *Perfomance Score*, *Load Time*, *Speed Index*, *Time to Interactive (TTI)* dan *Time to First Byte (TTFB)*.

1.4 TUJUAN

Tujuan dari penelitian ini adalah:

- 1) Merancang arsitektur *webiste* menggunakan teknologi Jamstack pada *website* Kraton Jogja yang saat ini masih bersifat tradisional.
- 2) Mengimplementasikan arsitektur Jamstack dalam pengembangan *website* Kraton Jogja.
- 3) Mengetahui performansi web Kraton Jogja setelah menerapkan teknologi Jamstack.

1.5 MANFAAT

Melalui penelitian ini, diharapkan dapat memberikan manfaat baik dari sisi teoritis dan praktik, yaitu:

- 1) Kegunaan teoritis: Penelitian ini diharapkan dapat menjadi tambahan pustaka dalam bidang akademik maupun industri.
- 2) Kegunaan praktik: Penelitian ini diharapkan menjadi referensi dan pertimbangan dalam pengembangan web di industri khususnya bagi pihak Kraton Jogja untuk meningkatkan efisiensi web Kraton Jogja dan meningkatkan keamanan pada sistem *website*.

1.6 SISTEMATIKA PENULISAN

Penelitian ini terbagi menjadi beberapa bab. Bab 1 berisi tentang latar belakang, rumusan masalah, manfaat dan tujuan penelitian, batasan masalah dan sistematika penulisan. Bab 2 membahas tentang konsep webiste, desain arsitektur *website*, perbedaan *static* dan *dynamic site*, Jamstack beserta arsitekturnya, *Content Delivery Network (CDN)*, *Static Site Generator (SSG)*, Tradisional CMS, *Headless CMS*, Elasticsearch, Logstash dan Kibana (ELK) dan parameter-parameter performansi website. Cara penelitian seperti alur penelitian, proses perancangan dibahas pada bab 3. Bab 4 membahas tentang hasil perancangan dan analisis sistem berdasarkan hasil imlpementasi dan pengukuran. Kesimpulan dan saran pengembangan skripsi untuk kedepannya dideskripsikan pada bab 5.