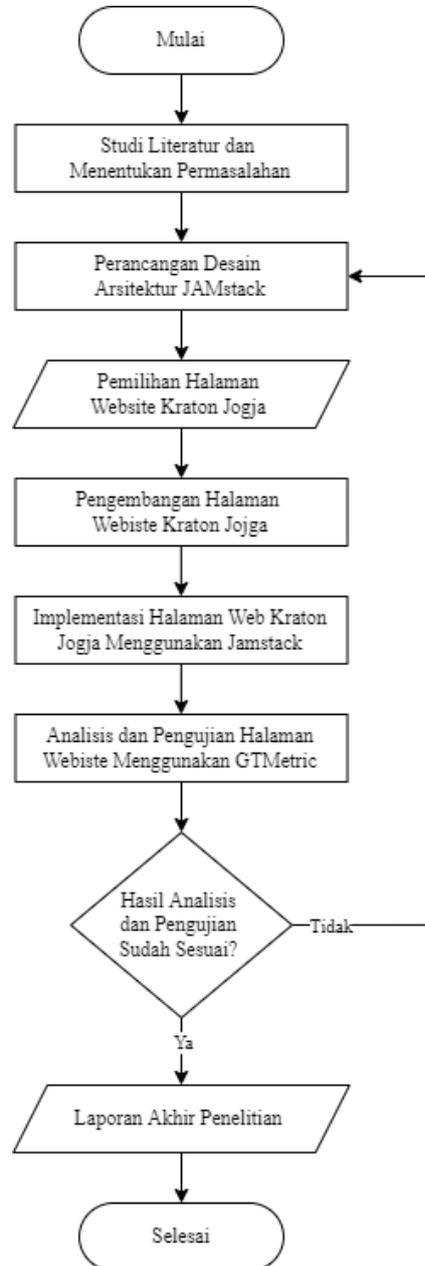


BAB III

METODE PENELITIAN

3.1 TAHAPAN PENELITIAN



Gambar 3.1 Alur Pengerjaan Penelitian

Penelitian ini akan melakukan desain arsitektur *website* Kraton Jogja dengan menggunakan teknologi Jamstack. Penelitian ini akan menentukan desain arsitektur *website* Kraton Jogja yang lebih modern dengan tujuan untuk meningkatkan efisiensi pengembangan dan memberikan pengalaman pengunjung

yang lebih baik. Penelitian diawali dengan melakukan studi literatur dan menentukan permasalahan. Permasalahan yang diambil adalah *website* Kraton Jogja yang merupakan *website* statis namun dinilai kurang efisien dan memberikan pengalaman pengguna yang kurang baik terutama pada saat *load time* di halaman yang memuat konten yang cukup banyak salah satu penyebab nya adalah penggunaan desain arsitektur *website* yang kurang efektif. Kemudian dilakukan desain arsitektur *website* Kraton Jogja dengan menggunakan Jamstack. Pada desain arsitektur tersebut akan terbagi menjadi dua bagian yaitu *front-office* dan *back-office*. Salah satu alasan dari pemisahan arsitektur tersebut adalah untuk meminimalisir terjadinya serangan *cyber* pada data-data *website* Kraton. Bagian *back-office* hanya akan dapat diakses oleh pengelola *website* dan tidak akan diberikan akses keluar dan pengunjung atau pihak luar hanya akan bisa mengakses sisi *front-office*. Untuk meningkatkan *security* juga akan dilakukan monitoring logs pada sisi *back-office* sehingga setiap aktivitas akan dapat dilihat melalui dashboard.

Pada penelitian ini akan mengambil salah satu halaman *website* Kraton Jogja yaitu pada halaman “peristiwa”, halaman tersebut dinilai cukup berat ketika dimuat pengunjung dan menjadi studi kasus yang tepat untuk membandingkan performansi saat setelah menggunakan Jamstack dan sebelum menggunakan Jamstack. Kemudian dilakukan pengembangan pada halaman tersebut dimana akan dibuat semirip mungkin dengan halaman “peristiwa” pada *website* Kraton Jogja sehingga pengujian akan lebih akurat. Kemudian akan dilakukan analisis dan pengujian menggunakan *software automation test* berbasis *website* menggunakan GTMetric dimana ini merupakan *software* yang umum digunakan untuk menguji performansi suatu halaman *website*. Tahap ini merupakan bagian terpenting karena akan mengukur hasil performansi *website* setelah menggunakan Jamstack dan akan dilihat perbandingannya dengan performansi *website* Kraton Jogja saat ini.

3.2 TOOLS YANG DIGUNAKAN

3.2.1. Directus

Pada penelitian ini menggunakan *Headless CMS* sebagai platform untuk mengelola konten *website* Kraton Jogja. Directus berfungsi untuk memberikan kemudahan khususnya bagi admin non-IT dalam mengelola konten *website*.

Directus memiliki *dashboard interface* yang *user friendly* sehingga mudah untuk dipahami dan digunakan. Directus berguna agar konten *website* menjadi dinamis sehingga ketika ada perubahan pada konten maka admin dapat dengan mudah melakukan perubahan. Konten akan dibagikan dan diolah ke *front-end* melalui API dengan URL `https://localhost/items/collection_name`. API ini akan memberikan respon berupa *JavaScript Object Notation* (JSON) sehingga pengembang *website* tinggal mengolah sesuai dengan kebutuhan yang ingin ditampilkan ke *website*. Untuk mengikatkan *security* pada konten *website* di Directus menyediakan *user*, *role* dan *permission*. *User* merupakan akun pribadi untuk melakukan autentikasi ke API dan aplikasi *website*, masing-masing *user* memiliki *role* atau jabatan dan diberikan *permission* khusus pada *user* tersebut. Selain itu di Directus dapat memberikan akses khusus sampai ke tingkat *collections* [24].

3.2.2. Eleventy

Eleventy merupakan salah satu SSG berbasis JavaScript yang cukup populer di kalangan developer. Eleventy fokus untuk menghasilkan asset statis dari data dan template yang sudah dibuat oleh pengembang. SSG ini murni JavaScript dan tidak mengadopsi *framework*. Proses *build* yang dilakukan Eleventy sangat cepat sehingga developer tidak perlu waktu lama untuk menunggu proses *build*. Salah satu yang membuat Eleventy berbeda dengan SSG lain adalah dapat mendukung dengan kurang lebih 10 bahasa dalam membuat template yaitu HTML, Markdown, JavaScript, Liquid, Nunjucks Handlebars, Mustache, EJS, Haml, dan Pug [25]. Pencampuran bahasa ini dapat dibuat dalam satu project yang sama atau antar *layout*. Kemampuan untuk mengkombinasikan 10 bahasa tersebut sangat *powerfull* karena dapat melakukan *build* dengan lebih fleksibel.

3.2.3. Elasticsearch, Logstash dan Kibana (ELK)

ELK berfungsi untuk melakukan penulisan aktivitas server atau *logs management system* dari Directus dan Eleventy. Penulisan *logs* sangat penting untuk mengetahui detail aktivitas yang terjadi pada suatu server. ELK juga dapat digunakan sebagai *monitoring* server sehingga ketika terjadi lonjakan traffic yang tidak wajar seperti serangan *Distributed Denial-of-Service* (DDoS) pada server admin *website* maka akan dapat langsung diketahui. Penggunaan ELK sebagai

bentuk antisipasi dan *monitoring* pada server tempat Directus dan Eleventy di *hosting*. Kelebihan dari ELK ini yaitu bersifat *open source* dan dapat mengubah format data log yang diterima sehingga dapat menampilkan log tersebut dalam bentuk visualisasi yang menarik.

3.2.4 Github

Github digunakan sebagai tempat *repository* atau penyimpanan kode file secara terpusat. Semua file assets statis termasuk gambar akan disimpan juga dalam *repository* ini. *Repository* ini akan digunakan sebagai tempat untuk menyimpan hasil halaman webiste yang telah di *build*. Selain *public repository* Github juga mendukung *private repository* sehingga *code* hanya akan bisa diakses oleh pemilik.

3.2.5. Netlify

Netlify merupakan *platform* yang digunakan untuk melakukan *deployment* atau hosting suatu *website*. Ini sangat cocok untuk tempat *hosting website static*. Netlify juga dapat menggunakan CI/CD untuk melakukan *deployment* sehingga ketika ada perubahan di Github maka Netlify akan ter trigger untuk melakukan *deployment*. Netlify juga menyediakan fitur untuk mengelola DNS, SSL dan lain-lain. Salah satu yang membuat menarik adalah dengan versi *starter* atau gratis maka seorang pengembang sudah dapat melakukan *deployment website* yang sudah tersebar ke berbagai server CDN yang dimiliki Netlify. Pemilihan Netlify adalah karena *platform* ini sudah terintegrasi dengan baik ke semua *tools* yang ada pada Jamstack sehingga developer dapat dengan mudah melakukan integrasi.

3.3 DESKRIPSI WEBSITE KRATON JOGJA

3.3.1 Pengertian Website Kraton

Website Kraton merupakan suatu *website* yang digunakan melayani kebutuhan masyarakat dan wistawan yang ingin mengetahui informasi tentang tentang Kraton secara online tanpa perlu datang secara langsung.

3.3.2. Deskripsi Website Kraton Jogja

Website Kraton Jogja merupakan suatu *website* yang digunakan keperluan informasi kepada masyarakat dan wistawan terkait Kraton Jogja. *Website* Kraton Jogja dapat diakses melalui <https://www.kratonjogja.id/>. *Website* ini diresmikan pada Selasa, 7 Maret 2017 yang diresmikan langsung oleh Raja Kraton Yogyakarta

sekaligus Gubernur DIY Sri Sultan Hamengku Buwono X di bangsal Sri Manganti. Dalam sambutan Sultan Hamengku Buwono X berharap *website* tersebut dapat membuka akses bagi masyarakat untuk memperoleh informasi dan layanan di Kraton Yogya agar Kraton. Sehingga tidak hanya warga tapi juga wisatwan yang ingin mengetahui tentang Kraton dapat membuka *website* tersebut.



Gambar 3.2 Tampilan *Website* Kraton Jogja

Website tersebut memiliki struktur sebagai berikut [26]:

- d. Sejarah: Bagian ini berisi tentang informasi sejarah Kraton Jogja dan raja-raja Kraton Jogja dari awal sampai saat ini. Terdiri dari dua bagian utama yaitu Cikal Bakal dan Raja-Raja.
- e. HB Ka 10: Berisi tentang biodata lengkap dari Raja Kraton Jogja saat ini yaitu Hamengku Buwono X (HB Ka 10). Bagian ini juga berisi tentang makna lambang Kraton Jogja. Terdiri dari dua bagian utama yaitu Biodata, Sebelum Bertahta, Lambang dan Regalia.
- f. Peristiwa: Berisi tentang semua kegiatan yang dilakukan di Kraton Jogja seperti seminar, *workshop* dan lain-lain.
- g. Tata Pemerintahan: Berisi struktural pemerintahan seperti Abdi Dalem dan Prajurit Kraton.
- h. Tata Rakiting Wewangunan: Berisi tentang lokasi-lokasi historis yang ada di Yogyakarta dan tempat yang biasa digunakan untuk acara Kraton Jogja.
- i. Hajad Dalem: Berisi tentang kegiatan di dalam Kraton Jogja seperti ulang tahun kenaikan tahta, Hari Besar Islam dan Siklus Hidup seperti upacara-upacara.

- j. Kagungan Dalem: Bagian ini berisi terkait kepunyaan Sultan yaitu mengenai budaya Kraton Jogja baik yang bersifat benda maupun tak benda.
- k. Kapustakan: Berisi aset-aset naskah-naskah dan obyek museum Keraton Yogyakarta. Terdapat tiga kategori besar merupakan koleksi dari Kawedanan Hageng Punakawan Nityabudaya, Widyabudaya, dan Kridhamardawa.
- l. Ragam: Berisi kegiatan kegiatan ragaman di Kraton Jogja seperti Tamanan yang berbentuk serial. Lestari Budaya dan lain-lain.
- m. Symposium: kegiatan konferensi budaya Jawa umumnya dalam bentuk webinar dan workshop.
- n. Aksara: merupakan versi dari *website* Kraton Jogja yang berbahasa Jawa.

3.4 SKENARIO PERANCANGAN ARSITEKTUR JAMSTACK

3.4.1 Skenario Desain Arsitektur

Arsitektur Jamstack akan terbagi menjadi dua bagian yaitu *Front-Office* dan *Back-Office*. Pemisahan ini bertujuan untuk membuat sebuah desain arsitektur yang efisien, handal dan aman terhadap serangan *cyber*.

a. *Front-Office*

Merupakan bagian yang akan dilihat oleh pengguna dan bisa diakses secara publik. Teknologi yang digunakan adalah Github sebagai tempat *repository* untuk menyimpan asset statis seperti kode, gambar dan lain-lain. Kemudian Netlify yang berguna sebagai tempat *deployment* aplikasi *website*. Pengguna akan mengakses *website* melalui protokol HTTP dan HTTPS untuk meningkatkan keamanan dalam mengakses *website*. Aplikasi yang akan di *deploy* ke Netlify adalah yang sudah di *build* dan di simpan di Github.

b. *Back-Office*

Bagian ini tidak akan dibuka akses secara publik dan hanya akan bisa diakses oleh pemilik saja, sehingga keamanan konten *website* akan terjaga. Teknologi yang akan digunakan adalah Directus sebagai konten manajemen *website* dan Eleventy digunakan untuk *generate* konten statis. Pada Directus akan dibuat sebuah *webhook* untuk melakukan aksi *build* secara otomatis ketika ada perubahan konten oleh seorang admin melalui Directus seperti menambah, mengubah dan mengedit konten.

3.4.2. Skenario Pengujian

Skenario pengujian performansi *website* yang telah di *deploy* pada *platform* Netlify dilakukan menggunakan GTMetric. Pengujian akan dilakukan pada 3 lokasi server yang berbeda yang telah disediakan oleh GTMetric pada versi gratis.



Gambar 3.3 Lokasi Pengujian Web Kraton Jogja

Masing-masing memiliki jumlah server yang berbeda dan dapat digunakan untuk melakukan pengujian seperti terlihat pada tabel 3.1.

Tabel 3.1 Lokasi Pengujian *Website* Pada GTMetric

No	Negara	Kota	Jumlah Server
1	Australia	Sydney	3
2	China	Hongkong	2
3	India	Mumbai	3

Pengujian di 3 lokasi berbeda sebagai bentuk simulasi jika pengunjung mengakses di lokasi berbeda-beda dengan tujuan untuk mendapatkan hasil pengujian performansi *website* yang lebih optimal dan akurat.

3.5 SIMULASI *WEBSITE* KRATON JOGJA

Pada penelitian ini akan mengambil studi kasus pada salah satu halaman *website* Kraton Jogja saat ini yang dinilai lama saat mengakses halaman tersebut. Hal ini karena cara kerja GTMetric adalah hanya akan menguji setiap halaman yang

tampil pada suatu URL yang dimasukkan. Studi kasus yang diambil adalah halaman *website* Kraton Jogja pada bagian “peristiwa”. Pada halaman tersebut tidak dibuat *pagination* sehingga *load time* untuk mengakses nya cukup lama sehingga cocok untuk dijadikan studi kasus untuk mengukur performansi.



Gambar 3.4 Halaman Peristiwa *Website* Kraton Jogja

Akan dilakukan pengembangan *website* pada laman tersebut dengan menggunakan teknologi Jamstack dan kemudian akan dilakukan pengukuran dengan menggunakan GTMetric lalu akan di analisis sesuai dengan parameter yang telah ditentukan. Pengembangan *website* pada halaman tersebut akan dibuat semirip mungkin pada setiap konten termasuk paragraf, gambar dan jumlah postingnya untuk meningkatkan akurasi hasil perbandingan sebelum dan setelah menggunakan teknologi Jamstack.