

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Website**

*E-Tourism*, atau pariwisata elektronik, adalah istilah yang menggambarkan penggunaan teknologi internet sebagai sarana promosi pariwisata dengan menyediakan akses informasi yang mudah diakses kapan saja dan di mana saja [2]. *E-Tourism* juga dapat diartikan sebagai organisasi atau bisnis yang telah memanfaatkan teknologi informasi untuk mendukung promosi pariwisata, sehingga mempermudah akses informasi [5]. Penggunaannya dalam mendukung pengembangan pariwisata dapat meningkatkan minat pengunjung. Salah satu teknologi yang dimanfaatkan untuk promosi dalam teknologi informasi adalah *website*.

*Website* merupakan kumpulan halaman yang berada dalam suatu domain atau subdomain di *World Wide Web* (WWW) pada internet [16]. Halaman-halaman tersebut dapat diakses melalui URL utama yang disebut homepage. URL ini mengatur halaman-halaman web dalam suatu hierarki, meskipun hyperlink di dalam halaman tersebut membantu pengunjung memahami struktur keseluruhan dan bagaimana alur informasi berjalan.

#### **2.2. Hapi JS**

API atau *Application Programming Interface* adalah mekanisme yang memungkinkan layanan didistribusikan di berbagai domain dengan cara yang mudah, terjangkau, dan skalabel [17]. API mencakup serangkaian protokol, fungsi, mekanisme, alat, definisi, dan atribut yang digunakan untuk berbagi dan mengembangkan layanan baru di berbagai domain [17]. API menggunakan *protocol* HTTP untuk dapat mengirimkan data melalui media internet dengan mengirimkan format data *Javascript Object Notation* (JSON)[18]. Terdapat 2 jenis API yang tersedia yaitu RESTful API dan GraphQL [17]. Salah satu *framework* yang dirancang khusus untuk membangun API yang kuat dan fleksibel adalah Hapi JS.

Hapi JS (sering disebut sebagai Hapi) adalah singkatan dari HTTP API. Ini adalah kerangka kerja yang kaya untuk membangun aplikasi dan layanan [19]. Fitur utama dari Hapi adalah sebagai berikut:

- a. *Routing* yang kuat dan fleksibel: Hapi memungkinkan untuk mendefinisikan rute dengan berbagai cara, termasuk menggunakan metode HTTP, jalur, dan parameter. Menggunakan *middleware* untuk mencegat dan memproses permintaan sebelum mencapai handler mereka.
- b. Validasi *input*: Hapi menyediakan berbagai *validator* bawaan untuk memvalidasi *input* pengguna dan juga dapat membuat *validator* kustom sendiri.
- c. Otentikasi dan otorisasi: Hapi menyediakan dukungan bawaan untuk berbagai skema autentikasi, termasuk *OAuth*, *Basic Auth*, dan *Digest Auth*. Juga dapat menggunakan *middleware* untuk menerapkan otorisasi kustom.
- d. *Caching*: Hapi menyediakan dukungan bawaan untuk *caching*, memungkinkan untuk menyimpan data dalam memori untuk meningkatkan kinerja.
- e. Dukungan *WebSocket*: Hapi menyediakan dukungan bawaan untuk *WebSocket*, memungkinkan untuk membangun aplikasi *real-time*.

Hapi JS merupakan platform yang menyediakan alur yang baik,. Hapi JS juga membantu menjaga agar modul Node.js yang berantakan tetap terkendali [20].

Kelebihan Hapi JS [19]:

- a. Ekosistem yang kaya: *Plugin* yang tersedia hampir selalu mencakup semua kebutuhan desain aplikasi.
- b. Jaminan yang kuat: Misalnya, urutan konfigurasi komponen tidak menjadi masalah.
- c. *Plugin* yang aman: *Plugin* dapat dengan aman mengandalkan *plugin* lain, termasuk urutan eksekusinya.
- d. Abstraksi yang fleksibel: *Cache*, *plugin*, dekorator, dan metode server semuanya dilindungi dan tidak dapat ditimpa secara implisit - jadi bisa mengucapkan selamat tinggal pada *middleware hell*.

### 2.3. REST API

REST atau *Representational State Transfer* adalah sebuah gaya arsitektur perangkat lunak yang digunakan untuk mendistribusikan sistem hypermedia, seperti *World Wide Web (WWW)* [21], [22]. REST memungkinkan komunikasi antar mesin atau aplikasi melalui protokol HTTP yang sederhana [23]. REST API adalah aplikasi yang menyediakan komunikasi berbasis HTTP dengan menggunakan perintah data seperti *GET*, *PUT*, *POST*, dan *DELETE* untuk layanan web [23]. Sebuah API dapat disebut RESTful API jika mampu memenuhi aturan dan batasan yang ditetapkan oleh gaya arsitektur REST. Aturan ini dirancang untuk menstandarkan proses komunikasi antar platform sehingga lebih fleksibel. Berikut adalah beberapa prinsip utama arsitektur REST [22]:

a. *Addressability*

*Addressability* adalah konsep di mana setiap *resource* dalam suatu sistem dapat diakses menggunakan *unique identifier* tertentu.

b. *Constrained & Uniform Interface*

Dalam sistem REST, metode atau prosedur yang digunakan untuk layanan apa pun terbatas pada metode yang tersedia dalam HTTP. Prinsip *uniform interface* pada REST biasanya disebut *CRUD (Create, Read, Update, Delete)*.

c. *Stateless Communication*

Komunikasi dalam REST bersifat *stateless*, yang berarti tidak ada informasi sesi yang disimpan di sisi server antara permintaan satu dengan lainnya.

d. Format Pesan Pertukaran

Ada dua format pesan yang umum digunakan dalam *web service*, yaitu *XML* dan *JSON*. Dalam aplikasi berbasis *JavaScript*, penggunaan *JSON* sebagai format pertukaran pesan memberikan peningkatan performa yang signifikan dibandingkan penggunaan pustaka tambahan untuk membaca data dari *Document Object Model (DOM)*.

### 2.4. Unified Modelling Language

*Unified Modeling Language (UML)* adalah metode visual untuk memodelkan dan merancang sistem berorientasi objek. Fungsi UML digunakan untuk mendeskripsikan dan merancang sistem perangkat lunak, terutama yang dibuat

menggunakan pemrograman berorientasi objek.[24] UML muncul dari integrasi beberapa bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir tahun 1980-an dan awal tahun 1990-an[23]. Jenis diagram UML yang digunakan dalam desain ini adalah Use Case Diagram dan Activity Diagram.

*Diagram use case* adalah diagram UML yang menggambarkan hubungan antara aktor dan sistem [25]. Diagram ini juga mengidentifikasi interaksi antara sistem yang dirancang dan satu atau lebih aktor [25]. Diagram aktivitas adalah diagram yang memodelkan proses dan aliran dalam suatu sistem [25]. Fungsi utamanya adalah untuk menggambarkan alur kerja atau aktivitas yang terjadi dalam suatu sistem perangkat lunak. Diagram ini berfokus pada aktivitas sistem itu sendiri, bukan pada aktornya...

## 2.5. Frontend

*Frontend* adalah antarmuka pengguna yang menyajikan fungsionalitas dan tujuan situs web dan berinteraksi langsung dengan pengguna, dan juga terkait dengan pengalaman pengguna saat berinteraksi dengan situs web.[26] Antarmuka pengguna menyediakan jembatan yang secara efisien menghubungkan respons sistem dengan respons pengguna. Tampilan antarmuka pengguna yang menarik merupakan faktor kunci dalam meningkatkan pengalaman pengguna dan kualitas penggunaan aplikasi [27]. Bahasa pemrograman yang umum digunakan adalah HTML, CSS, dan JavaScript. Selain itu, kerangka kerja dan pustaka seperti React JS, Angular, dan Vue JS sering digunakan untuk memfasilitasi pengembangan antarmuka pengguna yang lebih kompleks dan dinamis.

React, juga dikenal sebagai ReactJS, adalah pustaka *front-end* berbasis JavaScript untuk membangun antarmuka pengguna atau komponen UI. React dikelola oleh Facebook dan didukung oleh komunitas pengembang dan perusahaan. ReactJS menawarkan beberapa keuntungan, termasuk kecepatan, kesederhanaan, dan peningkatan skalabilitas [28]. ReactJS memungkinkan pengembang situs web untuk membuat komponen UI yang lebih interaktif, stateful, dan dapat digunakan kembali [28].

Bootstrap adalah kerangka kerja *front-end* situs web yang digunakan untuk menyederhanakan dan mempercepat pengembangan antarmuka situs web [29]. Bootstrap menyediakan komponen antarmuka dalam bentuk HTML, CSS, dan JavaScript yang dapat digunakan secara langsung atau dikembangkan lebih lanjut oleh pengembang situs web [29]. Bootstrap adalah framework yang dapat digunakan untuk desain web responsif. Situs web yang menggunakan komponen Bootstrap dapat disesuaikan dengan ukuran layar browser yang digunakan, baik itu *smartphone*, tab, atau *desktop* [29].

## 2.6. Cloud Computing

Komputasi awan atau *cloud computing* adalah ketersediaan sumber daya komputasi (seperti penyimpanan dan infrastruktur) sesuai permintaan, sebagai layanan melalui internet[30]. Ini menghilangkan kebutuhan individu dan bisnis untuk mengelola sumber daya fisik sendiri, dan hanya membayar untuk apa yang mereka gunakan. Terdapat 3 layanan *cloud computing* yaitu, *Platform as a Service* (Paas), *Software as a Service* (SaaS) dan *Infrastructure as a Service* (IaaS) [31][32]:

- a. SaaS adalah layanan yang di-*hosting* secara terpusat dengan model berlangganan untuk perangkat lunak berlisensi. Pengguna dapat mengakses layanan yang disediakan oleh aplikasi melalui berbagai perangkat stasioner dan seluler menggunakan antarmuka klien tipis seperti web browser.
- b. PaaS menawarkan kemampuan untuk menerapkan aplikasi yang dibuat atau diperoleh konsumen menggunakan bahasa pemrograman dan alat yang didukung oleh penyedia. Pengguna tidak mengontrol atau mengelola infrastruktur *cloud* yang mendasarinya termasuk server, jaringan, sistem operasi, atau penyimpanan. Pengguna memiliki kontrol atas aplikasi yang diterapkan dan, mungkin, konfigurasi lingkungan *hosting* aplikasi.
- c. IaaS memiliki kemampuan untuk menyediakan penyimpanan, pemrosesan, jaringan, dan sumber daya komputasi fundamental lainnya. Konsumen dapat menerapkan dan menjalankan perangkat lunak *arbitrer*, yang dapat mencakup sistem operasi dan aplikasi. Konsumen tidak mengelola atau mengontrol infrastruktur *cloud* yang mendasarinya tetapi memiliki kontrol atas sistem

operasi, penyimpanan, aplikasi yang diterapkan, dan mungkin kontrol terbatas atas beberapa komponen jaringan, misalnya *firewall host*.

Tiga penyedia layanan cloud di dunia industry ialah Amazon Web Service (AWS), Microsoft Azure, dan Google Cloud Platform (GCP) [33] .

### 1.6.1. Google Cloud Platform

Google Cloud adalah platform komputasi awan yang ditawarkan oleh Google. Google platform cloud (GCP) menyediakan infrastruktur sebagai layanan, platform sebagai layanan, dan komputasi tanpa server lingkungan. Google Cloud menyediakan beberapa layanan seperti mesin komputasi yang merupakan infrastruktur sebagai layanan untuk dijalankan mesin virtual (VM) seperti microsoft windows dan linux, app mesin yang merupakan platform sebagai layanan dan komputasi awan platform untuk membuat dan menghosting mesin virtual seperti web aplikasi di pusat data yang dikelola Google [34]. Layanan yang ditawarkan GCP diantaranya adalah Cloud run dan App Engine.

Cloud run adalah layanan komputasi tanpa server yang memungkinkan untuk menjalankan aplikasi berbasis *container* di lingkungan yang dikelola sepenuhnya [33]. *Container* merupakan paket yang memiliki kode aplikasi dan dependensi yang mencakup lingkungan *runtime* aplikasi dan Pustaka *runtime* diperlukan untuk menjalankannya [33] . Cloud run dapat men-*deploy* kode aplikasi dalam berbagai bahasa pemrograman selama aplikasi tersebut dibangun dalam *container image* bahasa tersebut. Cloud run memiliki beberapa fitur layanan standar, yaitu *endpoint* HTTPS yang berbeda disetiap layanan, penskalaan otomatis berbasis permintaan yang cepat, sistem pengelolaan *traffic* bawaan, dan layanan pribadi dan *public*[35]. Cloud run menawarkan beberapa metode untuk *deploy*, cara pertama dengan menggunakan langkah-langkah berikut[36]:

- a. Kemas aplikasi kedalam *Docker file*.
- b. Membuat *image* dari *docker file* di *local machine*.
- c. *Push image* yang sudah dibuat ke GCR atau Google Container Registry.
- d. *Deploy* aplikasi dari GCR ke Cloud run

App engine adalah platform PaaS dari Google yang memungkinkan untuk membantu pengembang untuk *hosting* dan menskalakan aplikasi web dan mobile

dengan mudah [37]. App engine menawarkan dua pilihan *environment* untuk menjalankan aplikasi, yaitu *standar environment* dan *flexible environment*. *Standar environment* adalah lingkungan dimana pengembang tidak perlu memikirkan infrastruktur yang mendukungnya, dan membatasi opsi konfigurasi dan bahasa pemrograman tertentu [38]. Ditujukan untuk aplikasi yang perlu menangani peningkatan skala dengan cepat, dijalankan dengan biaya yang sangat rendah atau gratis jika tidak ada lalu lintas dan mengalami lonjakan lalu lintas yang tiba-tiba dan ekstrim, serta memerlukan penskalaan segera [38].

### **1.6.2. Amazon Web Services (AWS)**

AWS (Amazon Web Services) adalah platform komputasi awan yang komprehensif dan terus berkembang yang disediakan oleh Amazon. Platform ini mencakup gabungan dari Infrastruktur sebagai Layanan (IaaS), Platform sebagai Layanan (PaaS), dan Perangkat Lunak sebagai Layanan (SaaS)[39]. AWS menyediakan berbagai alat, termasuk daya komputasi, penyimpanan basis data, dan layanan pengiriman konten. Amazon Web Services meluncurkan layanan web pertamanya pada tahun 2002, memanfaatkan infrastruktur internal yang dibangunnya untuk menangani operasi ritel daring.[39] Perusahaan ini mulai menawarkan layanan IaaS-nya sendiri pada tahun 2006. AWS adalah salah satu perusahaan pertama yang memperkenalkan model komputasi awan berbayar sesuai penggunaan yang skalabel dan menyediakan komputasi, penyimpanan, dan throughput sesuai permintaan. AWS menyediakan berbagai alat dan produk untuk bisnis dan pengembang perangkat lunak di 245 negara dan wilayah. Instansi pemerintah, lembaga pendidikan, organisasi nirlaba, dan organisasi swasta menggunakan layanan AWS [39].

## **2.7. Kajian Pustaka**

Kajian Anggun, Nazruddin, Surya, dan Muhammad mendeskripsikan penggunaan LLM (*Large-Scale Language Model*) dan Langchain dalam membangun sistem tanya jawab *chatbot* Telegram. Langchain digunakan untuk memfasilitasi pengembangan *chatbots*, dan LLM digunakan sebagai mesin pembelajaran untuk membuat teks menyerupai ucapan manusia. Penggunaan LLM

dan Langchain dalam sistem chatbot akan menciptakan chatbot yang dapat berinteraksi dengan pengguna dan menjawab pertanyaan terkait UU no. Undang-Undang Kesehatan tanggal 17 Januari 2023 adalah sumber data yang digunakan berdasarkan Undang-undang ini [40].

Rifky dan Edy melanjutkan penelitiannya dan mendiskusikan pengembangan *chatbot* informasi pariwisata Prefektur Pati dengan menggunakan teknik *Natural Language Processing* (NLP) berbasis aliran dialog. Pada penelitian ini, kami merancang *chatbot* untuk layanan informasi pariwisata di Provinsi Pati untuk membantu petugas informasi dan wisatawan menghemat waktu dan mendapatkan informasi yang dibutuhkan dengan cepat dan instan tanpa terikat oleh batasan waktu atau zona waktu tertentu. *Chatbot* yang terlatih mampu menjawab 12 pertanyaan berdasarkan total 13 skenario pengujian [41].

Penelitian lanjutan tentang perancangan media informasi Pusat Wisata Kuliner Wonorejo Kota Surabaya berbasis *website* dengan metode Model View Controller oleh Iqbal Ramadhan. Mengembangkan media informasi berbasis *website* menggunakan teknik pemrograman PHP dan *database* MySQL serta mengimplementasikan metode Model-View-Controller dengan dukungan framework CodeIgniter. Tujuan dari *website* ini adalah untuk memberikan informasi kepada penduduk lokal dan wisatawan dari luar Surabaya tentang berbagai kuliner Surabaya yang dapat dinikmati di Pusat Wisata Kuliner Wonorejo. Hasil ujian yang dilakukan secara internal dan eksternal menunjukkan tingkat kelulusan sebesar 80,67% [42].

Penelitian lebih lanjut tentang pengembangan *chatbot* WhatsApp menggunakan Node JS dan model *Natural Language Processing* untuk layanan PPDB oleh Imam, Melisa dan Taufik. Pengujian sistem menunjukkan bahwa sistem bekerja dengan baik dalam membalas pesan yang dikirim melalui WhatsApp, baik untuk pesan teks yang sesuai dengan pola maupun yang tidak sesuai dengan pola, dengan akurasi sebesar 92% [11].

Penelitian lebih lanjut oleh Wiyanto, Salsa dan Arif tentang e-tourism sebagai media informasi pariwisata berbasis *website* Kabupaten Bandung. Penelitian ini menggunakan metode waterfall untuk mengembangkan sistem informasi pariwisata guna membantu wisatawan yang berkunjung dan mengelola promosi destinasi

wisata di Kabupaten Bandung. Hasil penelitian ini disajikan dalam bentuk sistem e-wisata sebagai media informasi pariwisata berbasis website yang memudahkan wisatawan mengetahui objek wisata di Kabupaten Bekasi.

