

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Website

Website dapat diartikan sebagai Kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, suara atau gabungan semuanya yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang terkait. Web juga dikenal dengan sebutan *World Wide Web* atau WWW, yang dapat digunakan ketika terhubung internet yang menyediakan informasi dari sekedar informasi “sampah” atau informasi yang tidak berguna sama sekali sampai informasi yang komersial (Andi Prayudi, 2023).

2.2 Front-End Website

Front end sering disebut client-side, front end adalah bagian dari website yang berbentuk pemrograman untuk mengembangkan tampilan sebuah aplikasi atau website. Front end berkaitan dengan hal-hal visual, berfokus memperhatikan kegunaannya dimana rekayasa yang mengubah desain menjadi situs web interaktif yang lebih hidup (Andi Prayudi, 2023).

2.3 Application Programming Interface (API)

API atau *Application Programming Interface* (Antar Pemrograman Aplikasi). Api adalah mekanisme yang memungkinkan dua komponen perangkat lunak saling berkomunikasi menggunakan serangkaian protocol dan definisi. Arsitektur Api biasanya berkaitan dengan klien dan server (amazon.com, n.d.).

2.4 Postman

Postman adalah sebuah aplikasi yang berfungsi sebagai REST Client untuk uji coba REST API. Postman dapat mengakomodasi berbagai jenis request HTTP, menyimpan lingkungan untuk penggunaan selanjutnya, hingga mengubah API menjadi kode dengan berbagai bahasa pemrograman, seperti JavaScript dan Python. Postman juga biasa digunakan oleh

developer pembuat API sebagai tools untuk menguji API yang telah dibuat (Sekarputri, 2022).

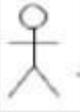
2.5 React JS

React JS dikembangkan oleh meta atau pada awal pengembangannya bernama Facebook. karena itu keluarga dari Meta juga menggunakan React JS misalnya seperti Instagram dan WhatsApp (biznetgio.com, n.d.). React JS adalah sebuah pustaka atau library javascript yang bersifat open source untuk membangun user interface, React JS hanya mengurus semua hal yang berkaitan dengan tampilan dan logika di sekitarnya (HANRY, 2019).

2.6 Use Case Diagram

Use Case diagram digunakan untuk merepresentasikan perilaku dinamis suatu sistem. Use case diagram merangkum fungsionalitas *system* dengan menggabungkan khusus pengguna, akktor, dan hubungan mereka membentuk sebuah model tugas, layanan, dan fungsi yang dibutuhkan oleh *system/* subsistem aplikasi. Tujuan utama dari use case diagram adalah untuk menggambarkan aspek dinamis dari sebuah system, mengakumulasikan persyaratan *system* mencakup pengaruh internal maupun eksternal (javatpoint.com, n.d.).

Tabel 2. 1 Simbol Use Case Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili User
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
	<i>Generalization</i>	Hubungan dimana objek anak (Descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
	<i>Include</i>	Menspesifikasikan bahwa Use Casesumber secara eksplisit
	<i>Extend</i>	Merupakan perluasan dari Use Case lain jika kondisi atau Syarat terpenuhi