

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian ini sebelumnya pernah dilakukan oleh Adi Darmawan (2009), dalam penelitian ini Adi Darmawan membahas tentang sistem informasi pada Kedai Kopi Kupas berbasis web menggunakan PHP dan MySQL. Dalam penelitiannya, Adi Darmawan menyajikan informasi Kedai Kopi Kupas berupa profil kedai kopi, produk, pemesanan tempat, pembatalan pemesanan tempat dan forum diskusi kedai.

Sebelumnya, penelitian ini juga pernah dibuat oleh Abdi Dwi Pranata (2010), dalam penelitian ini Abdi Dwi Pranata membahas tentang analisis dan perancangan sistem informasi pada *coffee shop* Kopitiam berbasis *client server*. Penelitian ini membahas tentang informasi data penjualan, data presensi karyawan, dan data penggajian.

Penelitian ini sebelumnya pernah dibuat oleh Kristina Windesi (2013) dalam penelitian ini Kristin Windesi membahas tentang sistem informasi wisata kuliner di Wamena yang berbasis Web. Sistem yang dibahas oleh Kristin Windesi berupa informasi wisata kuliner yang memberikan tampilan informasi makanan, informasi souvenir, dan informasi rumah makan di Wamena yang berbasis web.

Penelitian sebelumnya juga pernah dibuat oleh Muh Zainuddin Saiful Majidi (2019), dalam penelitian ini Muh Zainuddin Saiful Majidi membahas tentang aplikasi pemesanan makanan menggunakan *Framework* Laravel. Aplikasi ini digunakan untuk memesan makanan di Lesehan Elen, Lombok Timur yang saat itu masih manual akhirnya diganti menggunakan aplikasi.

Selain itu, penelitian ini juga pernah dilakukan oleh Daulat Alexander Sibarani (2019), dalam penelitian ini Daulat Alexander Sibarani membuat sistem informasi makanan menggunakan CMS (*Content Management System*) Wordpress yang menampilkan resep makanan, lokasi rekomendasi untuk makan keluarga, dan artikel tentang referensi berbagai tempat makan sebelum berkunjung ke tempat makan tersebut.

Tabel 2.1 Data Tinjauan Pustaka

Penulis	Objek	Teknologi	Hasil
Adi Darmawan (2009)	Kedai Kopi Kupas	PHP, MySql	Aplikasi sistem informasi berupa profil kedai kopi, produk, pemesanan tempat, pembatalan pemesanan tempat dan forum diskusi kedai
Abdi Dwi Pranata (2010)	Coffeeshop Kopitiam	Visual Basic 6.0, Microsoft SQL	Aplikasi sistem informasi berisi data penjualan, data presensi karyawan dan

			data penggajian
Kristina Windesi (2013)	Wisata Kuliner Wamena	PHP, MySql	Aplikasi berisi sistem informasi makanan, souvenir, maupun rumah makan di Wamena
Muh Zainuddin Saiful Majidi (2019)	Lesehan Elen Lombok	Framework Laravel	Aplikasi pemesanan makanan online
Daulat Alexander Sibarani (2019)	Sistem Informasi Makanan	CMS Wordpress	Website rekomendasi resep makanan, lokasi rekomendasi makan keluarga
Usulan penelitian Yanuar (2019)	Kedai Kopi	Typescript, ReactJS	Aplikasi sistem informasi berupa profil kedai kopi, menu kedai, harga menu, promo kedai kopi dan pemberian rating

2.2 Dasar Teori

2.2.1 Kedai Kopi

Menurut Nurazizi (2013), kedai kopi merupakan tempat yang menyediakan kopi beserta produk turunannya sebagai minuman utama dan berbagai jenis minuman (sampingan) lainnya seperti teh dan coklat, selain

menyediakan jenis makanan ringan yang juga ditawarkan sebagai kudapan pendamping minum kopi. Kedai kopi juga merupakan tempat berkumpulnya orang-orang yang sekedar bersantai atau melakukan aktivitas (ringan) lainnya seperti diskusi atau obrolan, membaca media cetak, online atau buku, menyelesaikan beberapa tugas akademik atau non-akademik hingga bersenang-senang dengan hiburan yang ditawarkan.

Secara umum kedai kopi adalah tempat yang menyediakan dan menjual minuman olahan dari biji kopi untuk dikonsumsi oleh masyarakat. Kedai kopi adalah bangunan yang digunakan sebagai tempat berjualan makanan dan minuman. Kedai kopi terbentuk untuk memfasilitasi kebutuhan produsen dalam melangsungkan hidup dengan menjual minuman atau produk berupa kopi (juga makanan) selain didukung dan dibentuk oleh faktor lain seperti budaya masyarakat yang menyukai kopi dan menjadikan kedai kopi sebagai salah satu tempat untuk berinteraksi dengan sesama masyarakat (Badan Pengembangan dan Pembinaan Bahasa, 2015).

2.2.2 Antarmuka (*user interface*)

User interface adalah cara program dan pengguna untuk berinteraksi. Istilah *user interface* terkadang digunakan sebagai pengganti istilah *Human Computer Interaction* (HCI) dimana semua aspek dari interaksi pengguna dan komputer. Semua yang terlihat di layar, membaca dalam dokumentasi dan dimanipulasikan dengan *keyboard* atau *mouse* juga merupakan bagian dari user interface (Lastiansah, 2012).

User interface memiliki fungsi untuk menghubungkan atau menerjemahkan informasi antara pengguna dengan sistem operasi, sehingga komputer dapat digunakan. Dengan demikian, *user interface* bisa juga diartikan sebagai mekanisme inter-relasi atau integrasi total dari perangkat keras dan lunak dalam membentuk pengalaman berkomputer. Konsep *user interface* memiliki banyak aspek yang perlu diperhatikan, karena akan mengacu pada beragam aplikasi teknologi seperti *electronic display*, aplikasi web, aplikasi *mobile*, dan lain sebagainya.

Menurut Sridevi (2014), *user interface* merupakan salah satu hal yang paling penting pada sistem atau produk yang berbasis komputer. Jika *user interface* dirancang dengan kurang baik, kemampuan *user* untuk memanfaatkan kekuatan komputasi dari suatu aplikasi mungkin akan sangat berkurang. Kenyataannya, *interface* yang lemah dapat mengakibatkan aplikasi yang telah dirancang dengan baik dan telah diimplementasikan menjadi gagal.

2.2.3 Typescript

Typescript adalah bahasa pemrograman berbasis JavaScript yang menambahkan fitur *strong-typing* dan konsep pemrograman OOP klasik (*class*, *interface*). Di dalam dokumentasinya, *typescript* disebut sebagai *superset* dari JavaScript, artinya semua kode *JavaScript* adalah kode *TypeScript* juga (Aep, 2016). Bahasa pemrograman ini menawarkan *class*, *module*, dan *interface* yang membuat *developer* bisa mengembangkan aplikasi kompleks dengan lebih mudah. Hal inilah yang membedakan *typescript* dengan *JavaScript*.

Keunggulan dan fitur-fitur dari *Typescript* antara lain: (1) *Support Class* dan *Module*, (2) *Static Type-Checking*, (3) *Support ES6 Feature*, (4) *Clear Library API Definition*, (5) *Build-in Support* untuk *JavaScript Packing*, (6) Kesamaan *Syntax* untuk *Backend*, dan (7) *Superset* dari *JavaScript*. (Anggie Bratadinata, 2019)

2.2.4 IO-ts

IO-ts secara teori adalah sistem tipe *runtime* untuk *IO decoding/encoding*. Dalam praktiknya, *io-ts* adalah solusi untuk masalah yang sangat rumit (Michal Zalacki, 2019). *IO-ts* adalah *library Javascript* yang digunakan untuk pengecekan struktur objek. Keuntungan menggunakan *io-ts* untuk mendefinisikan tipe *runtime* adalah dapat memvalidasi tipe data saat *runtime* dan juga dapat mengekstraksi tipe statis yang sesuai, sehingga tidak perlu mendefinisikannya dua kali.

2.2.5 REST API

Menurut Yusa Indermana (2017), *Application Programming Interface* (API) adalah sebuah teknologi untuk memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. API adalah antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama.

REST (*Representational State Transfer*) adalah satu implementasi dari *web service* yaitu sebuah standar yang digunakan untuk pertukaran data antar aplikasi atau sistem. REST adalah suatu arsitektur metode komunikasi yang

menggunakan protokol HTTP untuk pertukaran data. Dimana tujuannya untuk menjadikan sistem yang memiliki performa yang baik, cepat, dan mudah untuk dikembangkan (*scale*) terutama dalam pertukaran dan komunikasi data.

RESTful API merupakan implementasi dari API. RESTful juga disebut sebagai protokol untuk melakukan REST. RESTful API memiliki 4 komponen penting didalamnya, diantaranya adalah:

- a. *URL Design*, RESTful API diakses menggunakan protokol HTTP. penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti *developer*.
- b. *HTTP verbs*, setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang diminta client, diantaranya adalah GET, POST, PUT, DELETE.
- c. *HTTP Response Code*, adalah kode standarisasi dalam menginformasikan hasil *request* kepada *client*. Secara umum dibagi 3 kelompok yaitu 2xx (*response code* bahwa request berhasil), 4xx (*response code* bahwa request mengalami kesalahan pada sisi *client*) dan 5xx (*response code* bahwa *request* mengalami kesalahan pada sisi server).
- d. *Format Response*, setiap *request* yang dilakukan *client* akan menerima data *respons* dari server, response tersebut biasanya berupa data XML atau JSON (Sepry Haryadi, 2016).

2.2.6 ReactJS

Menurut Kumar dan Singh (2016) ReactJS merupakan bagian *view* dari konsep MVC (*model-view-controller*) yang berarti ReactJS hanya mengurus bagian tampilan antarmuka dengan pengguna saja, tanpa mengurus bagaimana cara mendapatkan data ataupun hubungan ke basis data.

ReactJS merupakan *JavaScript library* yang dikembangkan oleh Facebook untuk memfasilitasi pembuatan komponen antarmuka yang interaktif, *statefull*, dan mudah untuk digunakan ulang. ReactJS sangat cocok digunakan untuk *rendering* antarmuka yang kompleks dengan performa tinggi (Kumar dan Singh, 2016)

Dalam pengembangannya, banyak sekali hal yang telah dapat dijalankan oleh ReactJS seperti menganimasikan suatu objek dengan efek transisi, menjalankan permainan di web browser yang sepenuhnya diprogram dengan menggunakan ReactJS ataupun validasi *form* yang berjalan secara *real-time* sembari pengguna mengisikan data pada form tersebut. Pemakaian ReactJS dalam sebuah situs dapat dilihat menggunakan alat tambahan pada Google Chrome yang bernama *React Developer Tools* yang dapat mendeteksi keberadaan atau penggunaan ReactJS dalam suatu web (M dan Sonpatki, 2016).

Hal ini membuktikan bahwa ReactJS merupakan salah satu *library JavaScript* yang sangat berkembang, banyak digunakan, serta sangat handal dalam melaksanakan tugasnya sebagai *UI-rendering JavaScript library*.