

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan diuraikan pemahaman sebelumnya mengenai topik penelitian. Penjelasan ini mencakup beragam literatur, teori, hasil penelitian sebelumnya, serta konsep-konsep terkait yang menjadi dasar untuk perancangan, pelaksanaan, dan interpretasi penelitian yang sedang dilakukan. Penjelasan ini akan disajikan dengan merinci beberapa poin kunci sebagai berikut.

#### **2.1 Studi Literatur**

Fase studi literatur merupakan cara mengumpulkan referensi terkait dengan pemodelan simulasi. Literatur-literatur nantinya digunakan untuk mempelajari semua yang berhubungan dengan penelitian yang akan dilakukan seperti: rancang bangun *back end* pada sistem informasi, penggunaan Application Programming Interface pada sistem serta penggunaan metode pengembangan perangkat lunak Test Driven Development.

##### **2.1.1 Studi Pustaka**

Dalam melakukan tinjauan pustaka, dilakukan pengumpulan data melalui kajian, penelitian, dan telaah literatur, seperti buku, jurnal, tesis, dan artikel-artikel yang berkaitan dengan topik penelitian ini. Sebelum melanjutkan penelitian lebih lanjut, penulis juga membangun dasar teoritis, meningkatkan pemahaman mendalam tentang topik yang sedang diselidiki, dan mengidentifikasi kesenjangan pengetahuan yang dapat diisi oleh penelitian ini. Hal ini bertujuan untuk merumuskan pertanyaan yang relevan dan menetapkan tujuan penelitian secara jelas.

##### **a. Sistem Informasi Akademik (SIKAD)**

Sistem informasi akademik merupakan suatu sistem yang dibuat untuk mendukung administrasi dan pengelolaan data di lembaga pendidikan, seperti sekolah, universitas, atau institusi pendidikan lainnya. Sistem ini mencakup berbagai aspek yang berkaitan dengan kegiatan akademik, antara lain pendaftaran siswa, informasi kurikulum, dll. Sistem informasi akademik bertujuan untuk mempermudah pengaksesan informasi, meningkatkan efisiensi dan efektifitas dalam pengelolaan data dan proses akademik[9].

##### **b. Pengembangan *Back End***

*Back End* merupakan sistem perangkat lunak atau aplikasi yang berada di sisi server. *Back end* juga bertanggung jawab untuk menerima permintaan dari pengguna melalui user interface dan mengambil tindakan yang diperlukan untuk memenuhi permintaan tersebut. Beberapa dari bahasa pemrograman yang digunakan dalam pengembangan backend meliputi PHP, Javascript, C+, Python, dan sejumlah lainnya.

*Backend* adalah proses yang dijalankan sistem informasi atau aplikasi untuk menambah, mengubah, dan menghapus data. *Backend* biasanya menangani semua jenis proses yang tidak berhubungan langsung dengan pengguna, seperti server dan database. jika terdapat perintah yang diberikan memerlukan data dari basis data, backend akan mengambil data dari basis data, memproses data tersebut kembali, dan memberikan respond kepada sistem[10].

### **c. Application Programming Interface (API)**

Application Programming Interface adalah sekumpulan aturan dan protokol yang memungkinkan berbagai aplikasi untuk berkomunikasi dan saling berinteraksi satu sama lain. API dapat berfungsi sebagai perantara yang memungkinkan dua perangkat lunak atau aplikasi berinteraksi satu sama lain, meminta data, dan menyampaikan hasil secara efisien[11]. Penggunaan API dapat menyederhanakan proses pengembangan aplikasi, memberikan efisiensi baik dari segi waktu maupun biaya. Implementasi API dalam pembangunan sistem memberikan fleksibilitas dalam merinci desain, interaksi pengguna, tata kelola administratif, dan menciptakan banyak potensi inovatif.

Maka, API didefinisikan sebagai antarmuka sistem yang menetapkan cara interaksi antara beberapa perantara perangkat lunak. Di bawah ini terdapat berbagai fitur API yang sering dimanfaatkan oleh pengembang.:

- Jika seorang pengembang menginginkan penanaman peta lokasi untuk menampilkan koordinat suatu tempat, mereka dapat melakukannya tanpa perlu membuat tampilan gambar koordinat tempat tersebut dari awal pada tahap awal pengembangan aplikasi.
- Jika seorang pengembang ingin menyematkan situs web untuk menampilkan beberapa halaman situs web, mereka dapat melakukannya tanpa harus

mengonfigurasi tampilan browser situs web dari awal, hanya untuk aplikasi yang mereka kembangkan.

#### **d. Representational State Transfers (REST)**

Sistem Representational State Transfers (REST) adalah arsitektur perangkat lunak yang digunakan dalam pengembangan aplikasi web dan layanan web yang berfokus pada pertukaran data yang mudah dan efisien antara komponen perangkat lunak yang berbeda. Di dalam sistem REST, dapat melakukan metode HTTP request yaitu GET, POST, PUT dan DELETE[12]. Dalam kerangka kerja REST, sebuah sumber daya atau layanan dianggap sebagai entitas yang dapat diakses melalui URL (Uniform Resource Locator) yang bersifat unik untuk berinteraksi dengan layanan tersebut. Metode HTTP yang digunakan mencakup beberapa fungsi atau tugas dalam permintaan, seperti GET untuk mengambil data, POST untuk mengirim data baru, PUT untuk memperbarui data yang ada, dan DELETE untuk menghapus data.

Salah satu ciri khas dari REST adalah statelessness, yang berarti bahwa setiap permintaan yang dikirim dari klien ke server harus mengandung seluruh informasi yang diperlukan untuk memahami dan memproses permintaan tersebut. REST sangat sesuai untuk komunikasi antara sistem-sistem yang bekerja secara independen, seperti dalam konteks layanan mikro[13].

#### **e. Web Service**

Web service adalah teknologi yang membolehkan aplikasi dan sistem yang berbeda untuk berkomunikasi dan bertukar data melalui jaringan komputer atau internet.. Web service menggunakan sistem protokol standart seperti Representational State Transfers (REST) untuk mengirim dan menerima data antara aplikasi yang terhubung. Penggunaan konsep web service dapat menguntungkan pengguna untuk dapat lebih mudah dalam mengakses sebuah data[8].

Web service memiliki berbagai aplikasi dalam berbagai bidang, seperti e-commerce, aplikasi perbankan, sistem informasi, dan banyak lagi. Keamanan dan privasi adalah aspek penting dalam penggunaan web service, dan berbagai protokol dan teknik keamanan telah dikembangkan untuk melindungi data selama pertukaran. Tahapan pada penggunaan Web service dapat dilihat pada gambar dibawah ini.

#### **f. Test Driven Development (TDD)**

Test Driven Development (TDD) adalah suatu metodologi pengembangan perangkat lunak yang memfokuskan pada pengujian secara terstruktur dan otomatis. Pendekatan TDD mengharuskan pengembang untuk memulai dengan menulis tes sebelum kode fungsional[8]. Manfaat TDD yaitu peningkatan kualitas perangkat lunak dengan mengidentifikasi masalah sejak dini, mempermudah refaktorisasi kode, dan memberikan dokumentasi tentang bagaimana perangkat lunak berjalan. Serta melakukan eksperimen terhadap pengaruh TDD terhadap produktivitas, kode dan uji. Pendekatan Test-Driven Development (TDD) mendorong pengembang untuk memusatkan pada persyaratan fungsional perangkat lunak dan dapat mengurangi risiko penurunan kualitas saat kode diperbarui.

#### **g. Unit Testing**

Unit Testing adalah pengujian perangkat lunak yang memfokuskan pada pengujian unit-unit kecil dalam kode program secara terisolasi. Unit dalam konteks ini adalah bagian terkecil dari sebuah program yang dapat diuji secara terpisah, seperti fungsi, metode, atau kelas[14]. Tujuan utama dari unit testing adalah memastikan bahwa setiap unit kode berfungsi sesuai dengan apa yang diharapkan dan menghasilkan hasil yang benar.

Sebuah unit testing mempunyai struktur yang membentuknya. Mungkin terdapat beberapa perbedaan dari cara penggunaannya tergantung bahasa pemrograman. Proses unit testing melibatkan penulisan serangkaian tes kecil yang mengevaluasi perilaku setiap unit kode. Tes ini biasanya dikembangkan oleh pengembang dan menjalankan unit kode dengan berbagai masukan untuk memastikan bahwa ia menghasilkan kelaran yang diharapkan.

#### **h. JEST**

Jest adalah kerangka pengujian yang digunakan untuk menguji kode javascript, terutama kode yang ditulis dalam lingkungan React dan Node.js salah satu fitur utama jest adalah kemampuannya untuk menyediakan lingkungan pengujian yang lengkap, termasuk pembuatan dan pelaksanaan unit test, pengujian fungsional, pengujian integrasi, serta dukungan untuk mensimulasikan objek dan fungsi[14].

Jest juga mendukung konsep snapshot testing yang memungkinkan pengembang untuk mengambil snapshot dari output komponen React atau struktur data dan membandingkannya dengan snapshot yang ada untuk mendeteksi perubahan yang tidak diinginkan. Serta jest dapat menghasilkan laporan code coverage dimulai dari eksekusi pengujian.

### 2.1.2 Penelitian Terdahulu

Penelitian ini mengevaluasi informasi dan penelitian sebelumnya sebagai referensi, dengan mempertimbangkan penyelesaian masalah dan desain sistem yang telah dihasilkan dari penelitian sebelumnya tersebut.

Penelitian sebelumnya terkait perancangan *back end* serta penggunaan REST API. Jurnal referensi yang ditulis oleh Hendrik Fery Herdiyatomoko yang menjadi acuan dalam kajian penelitian ini adalah “BACK-END SYSTEM DESIGN BASED ON REST API”[6]. Penelitian ini dilakukan di SMA Xaverius 1 Palembang, membahas pengembangan back end berbasis REST API Aplikasi RESTful. Hal ini bertujuan untuk menyediakan data yang dapat diakses oleh REST Client melalui pertukaran data dalam format JSON menggunakan protokol HTTP. Framework yang digunakan dalam pembangunan ini adalah Laravel. Tujuan utama penelitian ini adalah membangun aplikasi berbasis REST Server yang dapat diakses oleh REST Client dari berbagai platform. Hasil penelitian ini mencakup pembangunan aplikasi REST Server berbasis Laravel 7 yang menyediakan endpoint dalam bentuk JSON dan kode status. Informasi ini dapat diakses oleh REST Client dari berbagai platform untuk mengelola data siswa.

Penelitian sebelumnya berkaitan dengan masalah pengembangan *back end* yang terintegrasi SIA sebagai pengelola aplikasi dan database yang dilakukan oleh Himawan Zidan Prayoga dengan judul “Development Of Final Study Management System Back End Integrated SIA In Faculty Of Engineering University Of General Soedirman ”[1]. Sistem *backend* ini berfungsi untuk mengelola semua aspek penelitian akhir, termasuk tugas akhir, hingga proses yudisium oleh lima pengguna yang berperan penting, yaitu Bapendik sebagai administrator, ketua jurusan, komisi tugas akhir, dosen, dan mahasiswa, masing-masing sesuai dengan perannya. Penelitian ini bertujuan untuk mengembangkan sistem backend penelitian studi akhir yang lebih efisien di Fakultas Teknik. Selain itu, penelitian ini juga bertujuan untuk meningkatkan efisiensi proses pembuatan berkas

penelitian akhir, sehingga seluruh bagian pada universitas dapat mengelola data hasil penelitian dengan lebih mudah, efisien, dan akurat.

Penelitian sebelumnya terkait pengujian unit Application Programming Interface menggunakan metode Test Driven Development (TDD) yang dilakukan oleh Sunardi, Imam Riadi dan Pradana Ananda Raharja dengan judul “Analisis Application Programming Interface Pada Mobile E-Voting Menggunakan Metode Test Driven Development”[7]. Pengujian perangkat lunak diperlukan sebagai langkah untuk menjamin kualitas dan kemampuan perangkat lunak. Penggunaan metode Test-Driven Development (TDD) dalam pengujian perangkat lunak dalam penelitian ini bertujuan untuk menilai dan meningkatkan kinerja perangkat lunak, serta memastikan kualitas kerja perangkat agar terhindar dari kegagalan atau kesalahan yang tidak diinginkan. Hasil dari pengujian dengan menggunakan metode TDD pada pembuatan API, dievaluasi melalui pengujian System Usability Scale dari pengalaman pengguna, dapat menunjukkan bahwa sistem ini dapat diterima dengan baik oleh pengguna.

Penelitian sebelumnya terkait pengujian sistem Point of Sale (POS) menggunakan menggunakan metode Test Driven Development (TDD) yang dilakukan oleh Egia Rosi Subhiyakto yang berjudul “Test-Driven Development (TDD) for Point of Sale System at Bicycle Shop”[8]. Penelitian ini bertujuan untuk mendukung penjualan di toko yang sebelumnya yang masih dilakukan secara konvensional. Tujuan penelitian ini yaitu pengembangan sebuah sistem POS dengan metode pengembangan berbasis tes (Test Driven Development/TDD) dimana sebuah tes ditulis sebelum fase pengkodean dengan tujuan agar kode yang dibuat telah lulus uji tes, mengurangi bug, dan menguji sistem. Hasil dari penelitian ini menunjukkan bahwa semua kode telah lulus uji tes. Kesimpulannya pembangunan sistem POS menggunakan metode TDD berhasil menghasilkan sistem yang berguna sesuai dengan persyaratan dan kualitas sistem yang diharapkan.

Dari keempat penelitian terdahulu terdapat beberapa persamaan seperti penggunaan pendekatan arsitektur REST API yang memungkinkan komunikasi antara berbagai komponen perangkat lunak dan digunakan secara luas dan penggunaan metode Test Driven Development sebagai upaya meningkatkan kualitas perangkat lunak dengan mengikuti siklus pengembangan yang berfokus pada pembuatan tes sebelum kode.

Adapun perbedaannya yaitu objek yang diteliti dan penarikan hasil dan saran penelitian. pada penelitian ini tujuan penelitian merupakan pemanfaatan sistem informasi akademik berbasis website pada SMK Muhammadiyah 7 Gondanglegi serta hasil yang akan diberikan berupa sistem yang dapat mengelola informasi akademik dari sekolah.

