

BAB 2

DASAR TEORI DAN TINJAUAN PUSTAKA

2.1 Dasar Teori (Pengembangan API RESTful)

Pengembangan *API RESTful* aplikasi testing ini berfokus pada *backend*, menggunakan arsitektur model *Architecture Monolith REST*. Dalam arsitektur monolith, seluruh sistem atau aplikasi dikembangkan sebagai satu kesatuan yang terdiri dari berbagai komponen yang terintegrasi. Semua fitur dan logika bisnis aplikasi ditempatkan dalam monolit ini. Dalam konteks pengembangan API, monolit berperan sebagai server yang menerima permintaan dari klien dan memberikan respons yang sesuai.

Monolit menyediakan *API RESTful* yang mengikuti prinsip-prinsip *REST*. *API* ini memungkinkan klien eksternal berinteraksi dengan sistem menggunakan metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE*. Permintaan dari klien dikirim melalui *HTTP* ke *endpoint* yang relevan pada *API* monolit, dan monolit merespons dengan data dalam format yang sesuai (misalnya, *JSON* atau *XML*). *API RESTful* ini memungkinkan klien untuk mengakses dan memanipulasi data dalam monolit.

2.1.1 API (Application Programming Interface)

API adalah sekumpulan aturan, protokol, dan alat yang digunakan untuk memungkinkan komunikasi dan interaksi antara perangkat lunak atau aplikasi yang berbeda. *API* dapat digunakan untuk mengakses, mengirim, dan menerima data dalam berbagai format, seperti *JSON* atau *XML*.

API juga menyediakan fungsi-fungsi atau layanan tertentu yang dapat diakses dan digunakan oleh pengembang aplikasi. *API* dapat digunakan dalam berbagai konteks, termasuk pengembangan aplikasi web, aplikasi mobile, integrasi sistem, dan pengembangan aplikasi berbasis *cloud*. *API* memungkinkan pengembang untuk memanfaatkan kekuatan dan fungsionalitas aplikasi atau sistem lain tanpa perlu memahami secara rinci bagaimana implementasinya dilakukan.

2.1.2 API RESTful

API RESTful (*Representational State Transfer*) adalah suatu gaya arsitektur yang digunakan dalam pengembangan *API* (*Application Programming Interface*) yang berbasis web.

API RESTful memungkinkan komunikasi dan interaksi antara klien (seperti aplikasi web atau aplikasi mobile) dan server melalui protokol HTTP (*Hypertext Transfer Protocol*).

API RESTful memungkinkan pengembang untuk memanfaatkan kekuatan protokol *HTTP*, membangun API yang *scalable*, dan memungkinkan integrasi yang mudah dengan aplikasi lain. Dengan mengikuti prinsip-prinsip *RESTful*, API dapat dirancang dengan antarmuka yang intuitif, mudah digunakan, dan mudah dimengerti oleh pengembang dan konsumen API.

2.1.3 Ruby On Rails

Ruby on Rails, atau yang sering disebut *Rails*, adalah kerangka kerja (*framework*) *open-source* yang digunakan dalam pengembangan aplikasi web. Dibangun dengan menggunakan bahasa pemrograman Ruby, *Rails* menyediakan berbagai fitur dan konvensi yang dirancang untuk mempercepat pengembangan aplikasi web secara efisien.

1. *Rails* mengikuti arsitektur *Model-View-Controller* (MVC), yang memisahkan logika bisnis (model), tampilan (view), dan pengendalian interaksi (controller).
2. *Rails* menyediakan abstraksi tingkat tinggi untuk akses basis data menggunakan *ActiveRecord*.
3. *Rails* memiliki ekosistem yang kaya dengan ribuan gems (paket pustaka) yang dapat digunakan untuk menambahkan fungsionalitas tambahan ke aplikasi.
4. *Rails* mendukung skalabilitas aplikasi melalui penggunaan mekanisme *caching*, pembagian beban (*load balancing*), dan skalabilitas horizontal (*horizontal scaling*) dengan mudah.
5. *Ruby on Rails* telah digunakan dalam pengembangan berbagai aplikasi web yang sukses dan memiliki komunitas yang aktif dan solid. Dengan kemudahan penggunaan, konvensi yang jelas, dan fokus pada produktivitas. *Ruby on Rails* menjadi pilihan populer bagi pengembang dalam membangun aplikasi web yang elegan dan efisien.

2.1.4 Schema Database

Schema database mengacu pada struktur logis atau pemodelan data dari sebuah *database*. Secara sederhana, *schema database* adalah cara untuk mengorganisir dan mendefinisikan tabel, kolom, hubungan, dan aturan lainnya yang terkait dengan penyimpanan dan pengelolaan data dalam sebuah database.

Schema database berperan penting dalam pengembangan dan pengelolaan basis data. Dengan mendefinisikan struktur dan hubungan antar tabel serta menerapkan *constraint* yang sesuai, *schema database* membantu memastikan integritas data.

2.1.5 Pengujian API

Pengujian API (*Application Programming Interface*) adalah proses memeriksa dan menguji fungsionalitas, kinerja, keamanan, dan interaksi antara berbagai komponen perangkat lunak melalui antarmuka yang telah ditentukan. API memungkinkan berbagai aplikasi atau layanan berkomunikasi dan berinteraksi satu sama lain, sehingga pengujian yang cermat sangat penting untuk memastikan bahwa API berfungsi sesuai harapan dan tidak mengakibatkan masalah yang tidak diinginkan.

2.1.6 Dokumentasi API

Dokumentasi API adalah kumpulan informasi dan petunjuk yang menjelaskan cara penggunaan dan interaksi dengan suatu API (*Application Programming Interface*). Dokumentasi ini berfungsi sebagai panduan bagi pengembang atau pengguna lain yang ingin menggunakan API tersebut.

Dokumentasi harus menjelaskan jenis respons yang diharapkan dari setiap operasi API. Ini meliputi format respons (seperti JSON atau XML), status kode *HTTP* yang mungkin diterima, dan struktur data yang akan dikembalikan oleh API. Jika ada kode kesalahan atau pesan kesalahan yang mungkin muncul, juga perlu didokumentasikan.

2.1.7 Postman

Postman adalah platform kolaborasi yang populer untuk pengembangan API (*Application Programming Interface*). Platform ini menyediakan antarmuka yang ramah pengguna yang memungkinkan para pengembang untuk merancang, menguji, dan mendokumentasikan API. Berikut adalah beberapa fitur dan aspek kunci dari Postman :

- a. Pengujian API
- b. Pembuatan Permintaan
- c. Kumpulan Permintaan(Collection)
- d. Lingkungan (Environment)
- e. Otomatisasi
- f. Mock ServerPemantauan(Monitoring)
- g. Dokumentasi

h. Kolaborasi

Secara keseluruhan, Postman adalah alat yang kuat yang menyederhanakan proses pengembangan dan pengujian API, menyederhanakan kolaborasi, dan mengurangi waktu yang diperlukan untuk mengembangkan dan menjaga API.

2.2 Tinjauan Pustaka

Dalam pembuatan sistem yang akan dibangun mengacu dari beberapa penelitian yang sudah ada, acuan yang digunakan dijabarkan sebagai berikut :

Jayadi dan Atletiko(2010) pada penelitiannya membahas pengembangan Sistem Informasi Geografis(SIG) untuk periklanan menggunakan *Google Maps* pada *platform Ruby On Rails* dan *Flex*. Studi ini berfokus pada tim periklanan di Surabaya, Indonesia, yang kewalahan dengan jumlah besar titik periklanan di kota tersebut.

Herlina dan Hidayatullah(2017) pada penelitiannya melakukan implementasi sistem absensi berbasis QR Code pada SMP Negeri 11 Kota Sukabumi. Sistem ini dapat membantu proses pencatatn data kehadiran siswa secara efektif dan teliti. Selain itu sistem ini juga dapat menghasilkan laporan harian, bulanan dan tahunan untuk absensi siswa. *Framework* yang digunakan pada pengembangan sistem ini yaitu *Rails* menggunakan Bahasa pemrograman Ruby.

Lamuel(2019) pada penelitiannya membahas pengembangan aplikasi Mitra Bukalapak berbasis Web-API menggunakan *framework Ruby on Rails*. Tujuannya adalah mengembangkan fitur *Product Bundling* dan *BRILink Grosir* pada aplikasi Mitra Bukalapak. Metodologi yang digunakan meliputi studi literature, analisis dan perancangan, implementasi, pengujian, dan sistematika penulisan.

Aklani dan Yang (2022) pada penelitiannya membahas pengembangan dan implementasi system informasi studi tracer berbasis web untuk SMK Multi Studi Hight School(SMK MHS) di Batam. Tujuan dari Penelitian ini adalah memberikan informasi yang berguna untuk mengevaluasi hasil institusi pendidikan dan meningkatkan kualitas pendidikan. Pada pengembangan ini menggunakan *Ruby On Rails* sebagai *Backend*, *Devise* sebagai *authentical solution*, *Bootstrap* dengan template SB Admin 2 sebagai *frontend*, *PostgreSQL* sebagai *relational database*.

Pradana(2022) pada penelitiannya melakukan perancangan sistem *backend* untuk aplikasi HIMASKOM UNDIP, pada pengembangan ini menggunakan salah satu metode pengembangan aplikasi yang disebut agile. Model yang digunakan dari metode ini adalah *Test Driven*

Development(TDD). Strategi pengembangan perangkat lunak dengan menggunakan TDD adalah dengan membuat *test case* terlebih dahulu, kemudian dilakukan pembuatan kode serta refactoring.

Berikut merupakan tabel tinjauan pustaka yang digunakan sebagai acuan serta referensi dari pembangunan penelitian. Dapat dilihat pada table 2.1, 2.2 dan 2.3

Tabel 2.1 Acuan Tinjauan Pustaka

Penulis	Obyek	Bahasa Pemrograman	Keterangan
Jayadi dan Atletiko(2010)	Pembuatan Sistem Informasi Geografis Reklame Menggunakan Teknologi Google Maps pada Platform Ruby On Rails dan Flex	Ruby, Flex	Sistem periklanan berbasis web yang ada tidak memiliki kemampuan pemetaan, sehingga para peneliti mengembangkan aplikasi SIG menggunakan Google Maps, Flex, dan Ruby On Rails. Sistem ini mampu menampilkan peta digital dan satelit serta berhasil terintegrasi dengan sistem periklanan yang sudah ada. Studi ini menyimpulkan bahwa SIG berbasis web dapat dibuat menggunakan teknologi-teknologi ini.
Herlina dan Hidayatullah(2017)	Penerapan QR Code Untuk Sistem Absensi Siswa SMP Berbasis Web	Ruby	Pada penelitian ini menghasilkan implementasi sistem absensi berbasis QR Code pada SMP Negeri 11 Kota Sukabumi. Sistem ini membantu dalam pencatatan data kehadiran siswa secara efektif dan teliti. Selain itu, sistem ini juga mampu menghasilkan laporan harian, bulanan, dan tahunan untuk absensi siswa. Penelitian ini memberikan solusi praktis untuk meningkatkan efisiensi

Tabel 2.2 Acuan Tinjauan Pustaka (Lanjutan dari halaman sebelumnya)

Penulis	Obyek	Bahasa Pemrograman	Keterangan
			dan akurasi dalam proses absensi di sekolah.
Lemuel(2019)	Pengembangan Aplikasi Mitra Bukalapak Berbasis Web-API Menggunakan Framework Ruby On Rails.	Ruby	Aplikasi ini mengembangkan fitur Product Bundling dan BRILink Grosir pada aplikasi Mitra Bukalapak. Tujuannya yaitu untuk memudahkan bagi pengguna aplikasi Mitra Bukalapak dalam hal ini pihak UKM dalam melakukan transaksi grosir dan memberikan fitur yang dibutuhkan oleh PT Bukalapak agar mudah dalam pengelolaan aplikasinya.
Aklani dan Yang (2022)	Pengembangan dan Implementasi Tracer Study di SMK Multi Studi High School Batam Menggunakan Framework Scrum	Ruby, Bootstrap	Sistem ini dapat digunakan untuk mengumpulkan data profil dan status alumni, serta memberikan akses kepada admin untuk melihat, mengubah, dan menghapus data yang sudah dikumpulkan. Sistem ini juga dilengkapi fitur visualisasi data untuk memudahkan evaluasi hasil institusi pendidikan dan meningkatkan kualitas pendidikan.
Pradana(2022)	Pengembangan Back End Aplikasi “Himaskom Undip” Menggunakan Kerangka Kerja Hapi.Js	Javascript, Node.js.	Penyebaran Aplikasi Backend “HIMASKOM UNDIP” dilakukan pada penyedia layanan komputasi awan yaitu Google Cloud Platform (GCP). Layanan.

Tabel 2.3 Acuan Tinjauan Pustaka (Lanjutan dari halaman sebelumnya)

Penulis	Obyek	Bahasa Pemrograman	Keterangan
			pertama dari GCP yang digunakan adalah Google App Engine.
			Layanan ini digunakan untuk menyebarkan aplikasi backend yang ditulis menggunakan Node.js. Layanan ini dipilih karena konfigurasi yang perlu dilakukan cukup sederhana dan terdapat free-tier yang dapat dimanfaatkan untuk menghemat pengeluaran untuk server.