

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian ini menggunakan beberapa sumber pustaka yang berhubungan dengan kasus atau metode yang diteliti, Diantaranya yaitu :

Eko Yuswanto (2018), telah melakukan penelitian yang menghasilkan aplikasi untuk pengelolaan data penilaian yang dilakukan secara efisien dan dapat mengurangi terjadinya *human errors*. Dibuat dengan menggunakan *framework* Codeigniter, Bahasa pemrograman PHP, menggunakan *database* MySQL.

Riko (2019), dalam penelitiannya menghasilkan aplikasi yang bertujuan untuk penjadwalan ujian pendadaran dan seminar proposal mahasiswa yang berintegrasi dengan Whatsapp dengan menerapkan REST API. Dibuat dengan bahasa pemrograman PHP dan menggunakan *database* MySQL.

Andre Putra (2020), telah melakukan penelitian penggunaan RESTful API untuk menampilkan list berita yang ditampilkan di *frontend*. Aplikasi bertujuan untuk membangun sebuah sistem yang lebih dinamis dan berkembang dengan memanfaatkan teknologi Arsitektur *Microservice*. Adapun *tools* yang digunakan yaitu Bahasa pemrograman Python, menggunakan *database* PostgreSQL.

Aji Kurniadi Irawan (2021), dalam penelitiannya menghasilkan aplikasi yang bertujuan untuk mempermudah menginputkan nilai praktikum dengan memanfaatkan teknologi Arsitektur *Microservice*. Dibuat menggunakan *framework* Laravel.

Penelitian yang diajukan oleh peneliti tentang “Implementasi Arsitektur Microservice Untuk Media Pembelajaran Menggunakan Restful API”. Penelitian ini bertujuan untuk menghasilkan aplikasi sebagai media pembelajaran yang efektif kepada siswa dengan mengimplementasikan Arsitektur Microservice sebagai layanan server dan menggunakan RESTfull API. Dibuat dengan Bahasa pemrograman PHP menggunakan *database* MySQL.

Tabel tinjauan pustaka merupakan tabel yang dibuat untuk mendefinisikan penelitian yang sebelumnya hampir sama dilakukan dengan penelitian yang diajukan saat ini, adapun perbandingan yang menjadi tabel tinjauan pustaka penelitian yakni :

Tabel 2. 1 Tinjauan Pustaka

Peneliti	Topik	Masalah	Teknologi	Hasil
Eko Yuswanto (2018)	Implementasi Framework Codeigniter Untuk Pengelolaan di SMK ASH-SHIDDIQYAH Balingsal	Membangun sistem pengelolaan siswa yang efektif dan efisien	PHP, Javascript, Codeigniter	Aplikasi pengelolaan data penilaian dilakukna secara efisien, dapat mengurangi resiko human errors.
Riko (2019)	Implementasi REST API untuk sistem penjadwalan dan pendadaran seminar proposal skripsi mahasiswa (Studi Kasus: Program Studi Sistem Informasi STMIK AKAKOM)	Membangun aplikasi untuk penjadwalan ujian pendadaran dan seminar menggunakan REST API	PHP, MySql, Microservice, RESTfull API	Aplikasi untuk sistem penjadwalan pendadaran dan seminar proposal skripsi mahasiswa.

Andre Putra (2020)	Restfull API untuk menampilkan list berita arsitektur miscroservice	Bagaimana cara membuat REST API untuk menampilkan list berita menggunakan microservice dan ditampilkan di front-end	Javascript, python, MySQL	Aplikasi REST API untuk menampilkan list berita menggunakan microservice dan ditampilkan di front-end
Aji Kurniadi Irawan (2021)	Implementasi arsitektur untuk input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta menggunakan RESTfull API	Membangun sistem untuk menginput nilai praktikum memanfaatkan arsitektur microservice menggunakan restfull API	Microservice, vuetify, Laravel, lumen, Restfull API	Telah dibuat aplikasi untuk menginput nilai praktikum dengan memanfaatkan arsitektur microservice menggunakan restfull API
M Teguh Hardianto (2022)	Implementasi arsitektur microservice untuk media pembelajaran menggunakan RESTfull API (Studi Kasus: SMA N 1 Wanasaba)	Membuat sistem media memanfaatkan arsitektur microservice menggunakan RESTfull API	PHP, javascript, microservice, codeigniter, mysql, restfull API	Akan dibuat sistem media pembelajaran yang efektif dengan memanfaatkan arsitektur microservice menggunakan RESTfull API

2.2 Dasar Teori

2.2.1 *Microservice*

Microservices adalah sebuah pendekatan untuk mengembangkan aplikasi dengan rangkaian *service-service* yang kecil, yang mana setiap *service* berjalan pada prosesnya sendiri-sendiri. Aplikasi dipecah tidak sekedar hanya memisahkan antara *backend* dan *frontend*, melainkan dipecah menjadi *service-service* yang

berfungsi secara spesifik. Apabila *client* menjalankan layanan *website blog*, maka dengan pendekatan *Microservices* aplikasi bisa di-*breakdown* menjadi layanan khusus konten, statistik, ad, *reporting*, photo/video, *messaging*, *searching*, komentar dan lainnya, maka setiap *service* dapat berkomunikasi dengan mekanisme yang ringan.

Setiap *service* yang dibuat harus mengenkapsulasi data dengan logika bisnis yang beroperasi pada data itu sendiri, dan hanya dapat diakses melalui *published service interface*. Tidak ada database yang dapat diakses secara langsung dari luar *service* dan tidak ada data yang dibagikan (*sharing*) antara setiap *service*.

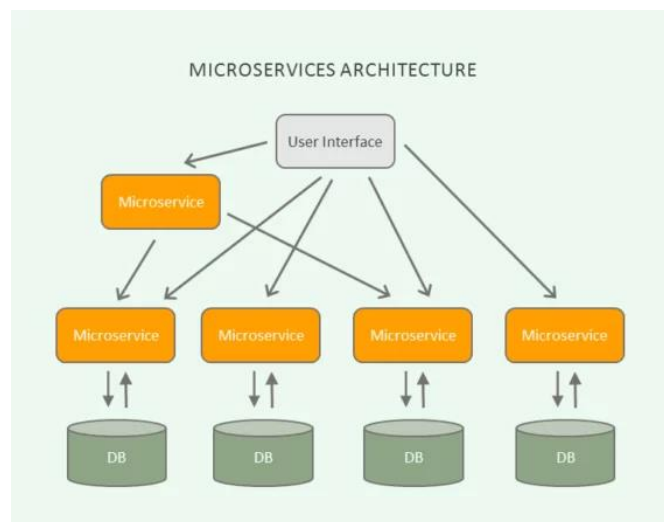
Oleh karena itu, setiap *service* yang dibangun harus memiliki landasan domainnya sendiri-sendiri dan process *sharing* data domain yang satu dengan yang lainnya hanya dapat dilakukan melalui *published service interface*.

Microservices memperkuat struktur modular yang sangat penting bagi tim yang sangat besar. Menurut Martin, Fowler ini adalah *key benefit* yang juga aneh jika dikatakan kelebihan, karena tidak ada alasan apapun mengapa *Microservices* memiliki struktur modular yang lebih kuat daripada *monolithic*.

Dalam arsitektur *monolithic* pada umumnya, sangatlah mudah bagi *developer* untuk melewati batas. Umumnya digunakan untuk mencari jalan pintas dalam mengimplementasikan fitur dengan lebih cepat. Akan tetapi berakhir pada merusak struktur modular yang berimplikasi pada penurunan produktivitas Tim.

Service yang sederhana lebih mudah di-*deploy* dan digunakan, karena berdiri sendiri, kecil kemungkinan kegagalan sistem terjadi saat salah satu *service* mengalami kesalahan.

Penggunaan *Microservices*, dapat mencampur dan menggunakan beragam bahasa pemrograman, *framework*, dan teknologi penyimpanan *database* yang digunakan. (Refractory, 2017)



Gambar 2. 1 Microservice Architecture

2.2.2 RESTfull API

RESTful API didasarkan pada teknologi *state transfer* (*Representational State Transfer*), gaya arsitektur dan pendekatan komunikasi yang sering digunakan dalam pengembangan layanan *web*.

Meskipun REST dapat digunakan di hampir semua protokol, tapi biasanya memanfaatkan HTTP ketika digunakan untuk Web API. Hal ini membantu pengembang *web* tidak perlu menginstal *library* atau perangkat lunak tambahan untuk memanfaatkan desain REST API. *Design* REST API pertama kali diperkenalkan oleh Dr. Roy Fielding dalam disertasi doktor tahun 2000-nya. REST API terkenal karena fleksibilitasnya yang luar biasa. Data tidak terikat dengan metode dan sumber daya, REST memiliki kemampuan untuk menangani beberapa

jenis panggilan, mengembalikan format data yang berbeda dan bahkan mengubah secara struktural tentunya dengan implementasi yang benar.

REST yang digunakan oleh *browser* dapat dianggap sebagai bahasa internet. Dengan meningkatnya penggunaan *cloud*, API muncul untuk mengekspos layanan *web*. REST adalah pilihan logis untuk membangun API yang memungkinkan pengguna untuk terhubung dan berinteraksi dengan layanan *cloud*. API telah banyak digunakan oleh situs-situs seperti Amazon, Google, LinkedIn dan Twitter. (Yudana, 2019)



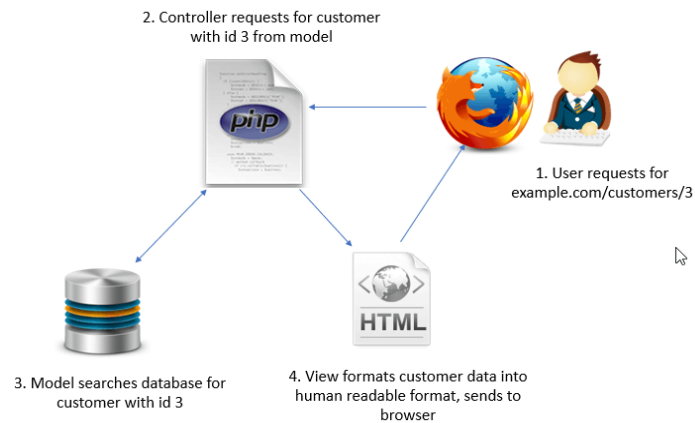
Gambar 2. 2 Arsitektur RESTfull API

2.2.3 Codeigniter

Codeigniter adalah *framework* PHP yang digunakan untuk mengembangkan aplikasi, keseluruhan *source code* pada Codeigniter hanya sebesar 2 MB, yang mana membuatnya menjadi mudah dipelajari dan cara kerjanya, fitur *built-in* pada Codeigniter didesain agar bekerja secara independen tanpa harus bergantung pada komponen lainnya.

Framework Codeigniter menggunakan desain arsitektur MVC, Codeigniter didokumentasikan dengan baik, dan memiliki *library* yang mirip dengan bahasa program PHP yang lain, dalam Codeigniter *user* meminta informasi, *controller* merespon permintaan tersebut, dan menyajikan informasi sesuai yang diminta.

Controller menerima permintaan dari pengguna lalu berinteraksi dengan model *database* jika perlu kemudian mengembalikan hasilnya kembali ke *browser* dalam bentuk kode HTML yang ditafsirkan oleh *browser* menjadi format yang dapat dibaca manusia dan ditampilkan kepada pengguna. (Techfor.id, 2020)



Gambar 2. 3 Konsep MVC

2.2.4 SMA Negeri 1 Wanasaba

SMA Negeri 1 Wanasaba merupakan Sekolah Menengah Atas di bawah naungan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi yang beralamat di jalan Raya Mataram, Labuhan Lombok, Wanasaba, Kec. Wanasaba, Kabupaten Lombok Timur, Nusa Tenggara Barat 83658. SMA Negeri 1 Wanasaba mulai beroperasi pada 2 September 2006, dan pada 22 Juni 2020 mendapatkan akreditasi A dengan No. SK 458/BAN-SM/SK/2020. (Kementerian Pendidikan, 2022)

Pada mulanya SMA Negeri 1 Wanasaba melakukan ujian harian, ujian tengah semester, ujian akhir semester, dan ujian lainnya masih dengan metode manual, di mana guru sebagai tenaga pengajar masih kesulitan dalam mengelola soal ujian maupun dalam mengelola nilai-nilai hasil ujian tersebut.