

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini menggunakan beberapa sumber pustaka terdahulu yang berhubungan dengan kasus atau metode yang akan dipakai dalam pembuatan skripsi ini diantaranya :

Ahmad Qomaruddin (2018), telah melakukan penelitian tentang *Implementasi Arsitektur Microservice Menggunakan RESTFUL API Untuk Portal Akademik PP AL-MUNAWWIR*. Penelitian ini bertujuan untuk menghasilkan aplikasi portal akademik untuk Pondok Pesantren Al-Munawwir, dimana layanan dari aplikasi dibangun secara terpisah berdasarkan fungsionalitasnya masing-masing dengan menggunakan *RESTful API* sebagai komunikasi antara *frontend* dengan *backend*. Dibuat dengan bahasa pemrograman Golang, dan menggunakan database CockroachDB.

Riko (2019), telah melakukan penelitian tentang *Implementasi REST API Untuk Aplikasi Penjadwalan Pendaran dan Seminar Proposal Skripsi Mahasiswa (Studi Kasus Program Studi Aplikasi Informasi STMIK AKAKOM)*. Penelitian ini bertujuan untuk menghasilkan aplikasi penjadwalan ujian pendaran dan seminar proposal mahasiswa program studi Aplikasi Informasi STMIK AKAKOM yang berintegrasi dengan Whatsapp dengan menerapkan *REST API*. Dibuat dengan bahasa pemrograman PHP dan menggunakan database MySQL.

Andre Putra (2020), telah melakukan penelitian tentang *RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices* untuk di tampilkan di *frontend*. Penelitian ini bertujuan untuk menghasilkan aplikasi list berita yang lebih dinamis dan berkembang dengan memanfaatkan teknologi arsitektur *Microservice*. Dibuat dengan bahasa pemrograman Python, menggunakan database PostgreSQL.

Aji Karuniadi Irawan (2021), telah melakukan penelitian tentang *Implementasi Arsitektur Microservice Untuk Input Nilai Praktikum Mahasiswa STMIK AKAKOM YOGYAKARTA Menggunakan RESTFUL API*. Penelitian ini bertujuan untuk menghasilkan aplikasi sebagai media input nilai praktikum mahasiswa STMIK AKAKOM YOGYAKARTA dan memberikan informasi nilai hasil praktikum untuk Mahasiswa, Dosen dan Penjaga LAB. Dibuat dengan bahasa pemrograman PHP, menggunakan database MySQL.

Ferdi Dirgantara (2021), telah melakukan penelitian tentang *Implementasi Arsitektur Microservice Untuk Pembuatan Surat Unit Kegiatan Mahasiswa Informatika dan Komputer menggunakan Node.js*. Penelitian ini bertujuan untuk menghasilkan aplikasi pembuatan dan pengelolaan surat-menyurat bagi sekretaris organisasi UKM IK dan memberikan informasi hasil surat-surat yang sudah dibuat kepada sekretaris organisasi. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman Javascript dan Node.js, menggunakan database NoSQL yaitu MongoDB sebagai media penyimpanan data.

Wisnu Andrian (2021), telah melakukan penelitian tentang *Implementasi Arsitektur Microservice Untuk Aplikasi Pencarian Barang Hilang Menggunakan*

Restfull Api (Studi Kasus Papan Informasi Kehilangan STMIK AKAKOM).

Penelitian ini bertujuan untuk menghasilkan aplikasi berupa informasi berita kehilangan pada seluruh warga STMIK AKAKOM. Dibuat dengan bahasa pemrograman PHP menggunakan database MySQL.

Pembahasan yang dibuat kali ini tentang “Implementasi Arsitektur Microservice Untuk Aplikasi Pengelolaan Proyek Perangkat Lunak Berbasis Website Menggunakan RESTFUL API (Studi Kasus: PT Aviera Global Indonesia)”. Penelitian ini bertujuan untuk menghasilkan aplikasi manajemen proyek perangkat lunak bagi PT Aviera Global Indonesia dengan menerapkan arsitektur sistem dari Microservice. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP dan menggunakan framework Codeigniter 3, kemudian menggunakan database MySQL sebagai media penyimpanan data.

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 :

Table 2.1 Perbandingan Tinjauan Pustaka

Penulis / Tahun	Topik	Bahasa	Teknologi	Database
Ahmad Qomaruddin (2018)	Implementasi Arsitektur Microservice Menggunakan RESTFUL API Untuk Portal Akademik PP AL-MUNAWWIR	Golang	Microservice, RESTful API	CockroachDB
Riko (2019)	Implementasi REST API Untuk Aplikasi	PHP	Microservice, RESTful API	MySQL

Penulis / Tahun	Topik	Bahasa	Teknologi	Database
	Penjadwalan Pendaran dan Seminar Proposal Skripsi Mahasiswa (Studi Kasus Program Studi Aplikasi Informasi STMIK AKAKOM)			
Andre Putra (2020)	RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices	Python	Microservice, RESTful API	PostgreSQL
Aji Karuniadi Irawan (2021)	Implementasi Arsitektur Microservice Untuk Input Nilai Praktikum Mahasiswa STMIK AKAKOM YOGYAKARTA Menggunakan RESTFUL API	PHP	Microservice, RESTful API, Vuetify, Laravel Lumen	MySQL
Ferdi Dirgantara (2021)	Implementasi Arsitektur Microservice Untuk Pembuatan Surat Unit Kegiatan Mahasiswa Informatika dan Komputer menggunakan Node.js	Javascript	Microservice, Node.js	NoSQL, MongoDB
Wisnu Andrian (2021)	Implementasi Arsitektur Microservice Untuk Aplikasi Pencarian Barang Hilang Menggunakan Restfull Api (Studi Kasus Papan Informasi)	PHP	Microservice, RESTful API, Codeigniter	MySQL

Penulis / Tahun	Topik	Bahasa	Teknologi	Database
	Kehilangan STMIK AKAKOM)			
Agrin Fauzi Nurrohman (2024)	Implementasi Arsitektur <i>Microservice</i> Untuk Aplikasi Pengelolaan Proyek Perangkat Lunak Berbasis Website Menggunakan <i>RESTFUL API</i> (Studi Kasus: PT Aviera Global Indonesia)	PHP	Microservice, Codeigniter	MySQL

2.2 Dasar Teori

2.2.1 Software House

Secara teori *Software House* adalah sebuah organisasi, perusahaan, maupun jasa, yang menyediakan layanan mengembangkan *software* atau perangkat lunak. Kata *software* disini mengacu pada berbagai kebutuhan komputasi seperti aplikasi *desktop* maupun *mobile*. Para pelaku usaha pengembangan perangkat lunak ini akan menerjemahkan kebutuhan Klien kedalam bentuk aplikasi.

Karena pesatnya perkembangan teknologi, bidang usaha *Software House* memiliki potensi yang besar. Mengingat kebutuhan aplikasi tak terbatas pada bisnis saja. Berbagai bidang, seperti pendidikan, medis, pemerintahan, dan masih banyak lagi juga membutuhkan bantuan komputasi untuk efisiensi kerja ke depannya.

Tak heran, para pelaku usaha teknologi ini berlomba-lomba untuk dapat membangun tim yang solid dengan kualifikasi dasar pemahaman terhadap bidang IT. Biasanya dalam sebuah perusahaan pengembangan perangkat lunak akan mencakup berbagai bidang teknis, khususnya yang berkaitan dengan

pengembangan aplikasi. Mulai dari desainer grafis, desainer UI/UX, analisa proses bisnis, pengembang, hingga berbagai bidang perkantoran pada umumnya. Walaupun ada kemungkinan satu projek di *handle* oleh satu orang, namun untuk hasil yang lebih baik pastinya butuh tenaga ahli di setiap bidangnya.

Walaupun begitu, masih banyak yang menyamakan perusahaan *Software House* ini dengan penyedia layanan IT. Padahal keduanya memiliki definisi, dasar, dan juga tujuan yang sangat berbeda. Karena secara konsep kita bisa menyimpulkan, bahwa pengembang aplikasi baik individu atau kelompok pastinya memiliki tujuan untuk membuat aplikasi sesuai kebutuhan klien. Sedangkan untuk penyedia layanan IT, klien akan berlangganan dengan layanan yang sudah ada sebelumnya. (Software House: Pengertian, Layanan, Struktur, Cara Kerja, 2023)

2.2.2 PHP

Pretext Hyper-Processor (Pratama, 2010). PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server-side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirim ke *browser* hanya hasilnya saja. PHP menyatu dengan *HTML* untuk membuat halaman *website* yang menarik.

Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat *website* yang bersifat *server-side-scripting*. PHP bersifat dinamis sehingga dapat dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan MacOs. Selain Apache, PHP juga mendukung beberapa *web server* lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan basis data untuk menghasilkan halaman *website* yang dinamis. Aplikasi manajemen basis data yang

sering digunakan bersama PHP adalah MYSQL. Namun, PHP juga mendukung aplikasi manajemen basis data seperti Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL. (Andi, Wahana Komputer, 2014:h,73).

Menurut (Budi Raharjo 2012:41) dalam jurnal Yesi Susanti, dkk, (ISSN: 1858 – 2680) PHP adalah salah satu bahasan pemrograman skrip yang dirancang untuk membangun aplikasi *website*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-parsing di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML yang selanjutnya akan ditampilkan kembali *web server*. Karena pemrosesan program PHP dilakukan di dalam lingkungan *web browser*, PHP dikatakan sebagai bahasa sisi *server (server-side)*. Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “*View Source*” pada *web browser* yang mereka gunakan.

2.2.3 Framework

Menurut Betha Sidik (2012) *Framework* adalah :“kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan pengembang dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu”.

2.2.4 Codeigniter

Menurut Betha Sidik (2012) CodeIgniter adalah :“ Sebuah *framework* PHP yang bersifat *open source* dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan *programmer* dalam membangun sebuah aplikasi berbasis *website* tanpa harus membuatnya dari awal”. Dalam situs resmi

codeigniter, (Official Website CodeIgniter, 2002) menyebutkan bahwa Codeigniter merupakan *framework* PHP yang kuat dan minim dari *bug*. Codeigniter ini dibangun untuk para pengembang dengan bahasa pemrograman PHP yang membutuhkan alat untuk membuat *website* dengan fitur lengkap. *Framework* Codeigniter dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. Kelebihan dari *framework* Codeigniter jika dibandingkan dengan *framework* lain adalah sebagai berikut :

1. Gratis (Open-Source)

Kerangka kerja Codeigniter memiliki lisensi dibawah Apache / BSD *open-source* sehingga bersifat bebas atau gratis.

2. Berukuran kecil

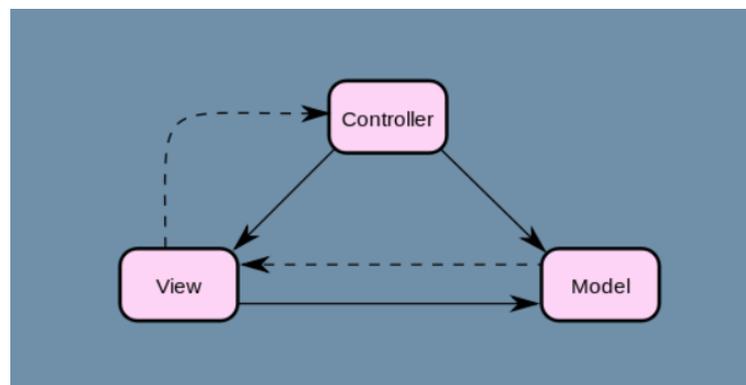
Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan *framework* lain yang berukuran besar dan membutuhkan *resource* yang besar dan juga dalam eksekusi maupun penyimpanannya.

3. Menggunakan konsep M-V-C

Codeigniter merupakan konsep M-V-C (ModelView-Controller) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Dengan konsep ini kode PHP, query MySQL, Javascript dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

- a. Model merupakan program (berupa OOP class) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (*input-edit-delete*).

- b. View merupakan kode program berupa *template* atau PHP untuk menampilkan data pada *browser*.
- c. Controller merupakan kode program (berupa OOP class) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol *model* dan *view*. Adapun alur dari program aplikasi berbasis codeigniter yang menggunakan konsep M-V-C ditunjukkan pada gambar berikut :



Gambar 2. 1 Model View Controller

2.2.5 MySQL

Menurut Arief (2011d:152) “MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *website* yang menggunakan basis data sebagai sumber dan pengolahan datanya”.

Menurut Adi Nugroho (2011) MySQL (*My Structured Query Language*) adalah: “ Suatu aplikasi basis data *relation* atau *Relational Database management System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan. MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multi user* (banyak pengguna). MySQL didistribusikan secara gratis dibawah lisensi GPL (General Public License).

Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan *closed source* atau komersial”.

MySQL merupakan basis data yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *website* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis *website*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP.

2.2.6 Microservice

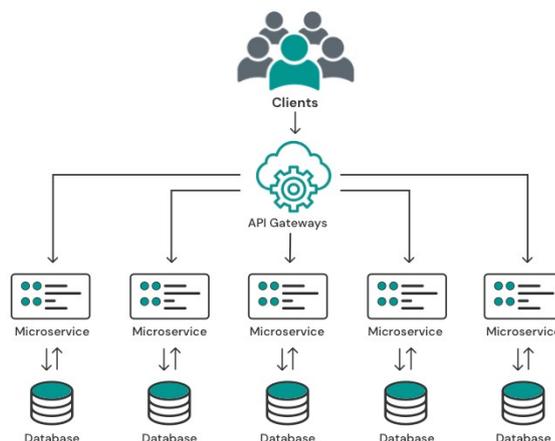
Microservices adalah *framework* yang digunakan untuk membagi aplikasi ke dalam beberapa jenis layanan atau fitur yang lebih kecil, namun masih saling berhubungan satu sama lainnya. Desain arsitektur ini akan memberikan kemudahan bagi *programmer* karena dapat mengerjakan secara mandiri di setiap fiturnya meskipun berasal dari tim yang berbeda-beda.

Ada berbagai contoh *microservices* yang bisa kita temui, salah satunya adalah layanan pada aplikasi Grab. Aplikasi ini memiliki beberapa fitur *microservices* seperti GrabRide, GrabFood, hingga GrabExpress. Fitur-fitur inilah yang saat ini menjadi salah satu andalan banyak aplikasi besar untuk mewujudkan tujuan mereka menjadi “*Lifestyle App*” ataupun “*Super App*”.

Hadirnya *microservices* dalam sebuah aplikasi ternyata bisa meningkatkan *customer experience*, karena bisa memenuhi berbagai kebutuhan pengguna hanya dalam satu aplikasi saja. Namun, hal ini harus dibarengi juga dengan *User Experience* yang baik dan sesuai.

Sebagai penyedia layanan, kita harus memastikan apakah pengguna merasa terbantu dengan hadirnya *microservices* dalam aplikasi tersebut atau tidak. Sehingga, jika *microservices* pada aplikasi kita berhasil, akan berdampak pada peningkatan penjualan untuk perusahaan.

Tujuan utama dari *microservices* adalah membuat proses pengembangan aplikasi menjadi lebih efektif sehingga tak hanya mempermudah bagi si pengguna, namun juga mempermudah dari sisi *programmer* dalam melakukan *maintenance* apabila terdeteksi adanya *bug* atau *error*. (Mengenal Apa Itu Microservices dalam Pembuatan Aplikasi, 2022)



Gambar 2.2 Arsitektur Microservice

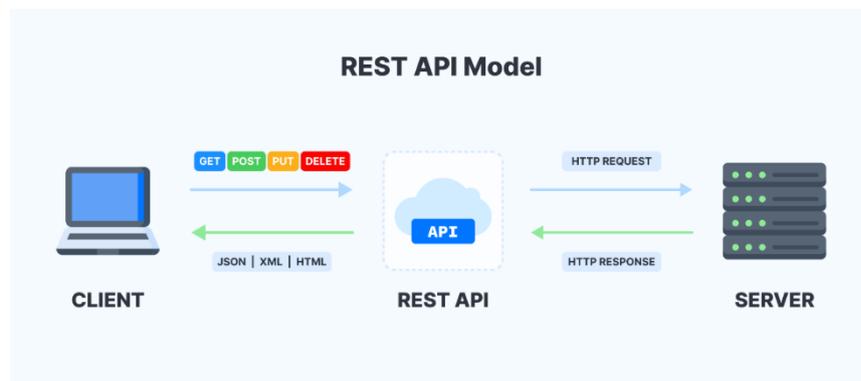
2.2.7 RESTFUL API

RESTful API didasarkan pada teknologi *state transfer* (*representational state transfer* / REST), gaya arsitektur dan pendekatan komunikasi yang sering digunakan dalam pengembangan layanan *website*.

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 *website*.

REST adalah pilihan yang 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 2.3 Arsitektur REST API