

BAB I

PENDAHULUAN

1.1 Latar Belakang

Microservices mampu memberikan pilihan solusi dalam membangun sebuah aplikasi yang efisien pada proses dan efektif dalam pengembangan. Terlebih lagi harus mampu tetap berfungsi baik pada saat mendapatkan lonjakan pengunjung (Rizal, 2018). *Microservices* berarti membagi aplikasi menjadi layanan yang lebih kecil dan saling terhubung tidak seperti aplikasi monolitik. Setiap *Microservices* merupakan aplikasi kecil yang memiliki arsitektur heksagonal sendiri yang terdiri dari logika beserta berbagai adaptornya (bahasa pemrograman, dll) (Dian, 2016).

Terdapat enam karakteristik utama dari *Microservices*, yang diantaranya adalah banyak komponen, ditujukan untuk kebutuhan bisnis, proses *Routing* yang sederhana, dapat berjalan sendiri, mengurangi resiko kegagalan, dan selalu berubah (Ibnu, Accurate, 2021). *Microservices* memiliki kelebihan dan kekurangan. Kelebihan *Microservices* adalah. Scalable, aman, lebih bebas, mudah dimengerti, sederhana, error terisolasi. Namun *Microservices* memiliki beberapa kekurangan diantaranya adalah. Test yang rumit, system yang sedikit rumit, dan perlu banyak automation (Amera, 2021).

Saat ini, integrasi telah berkembang dari yang awalnya menghubungkan aplikasi monolitik melalui ESB (*Enterprise Service Bus*), menjadi pervasive, yaitu menghubungkan aplikasi modern melalui API, fungsi, dan event-driven patterns. Dengan adanya evolusi ini, *PhinCon* memberikan solusi *Microservices* yang mendukung kebutuhan ESB (*Enterprise Service Bus*), sekaligus memudahkan *developer* untuk mengadopsi arsitektur aplikasi modern. *Microservices* juga membantu perusahaan memanfaatkan *platform function-as-a-service*, memungkinkan *developer* memanfaatkan *web-scale* computing dengan biaya yang sangat efisien. *Microservices* mendukung penggunaan IoT yang inovatif dan mengurangi

biaya operasional dengan menjalankan aplikasi lebih dekat ke tempat data dihasilkan. Logika *Microservices* dapat beroperasi di perangkat edge-computing terkecil, memproses data secara real-time, menghemat penggunaan bandwidth dan sumber daya komputasi terpusat (Admin M. , 2020)

RESTful API merupakan penerapan dari API (Application Programming Interface). Sedangkan REST (Representational State Transfer) adalah sebuah arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dimana metode ini sering diterapkan dalam pengembangan aplikasi. Dengan tujuannya untuk menjadikan sistem memiliki performa yang baik, cepat dan mudah untuk di kembangkan (scale) terutama dalam pertukaran dan komunikasi data (Admin, mii.co.id/, 2021)

RESTful API bersifat Stateless, yang artinya tidak adanya state dalam sebuah aplikasi. Sifat stateless membuat setiap *request* HTTP dilakukan secara terisolasi, server tidak menyimpan state apapun mengenai sesi dari client, setiap request dari client harus berisi semua informasi yang dibutuhkan server termasuk informasi otentikasi, dan *stateless* merupakan salah satu syarat untuk membuat *RESTful API*. Jadi, dapat disimpulkan bahwa dalam membuat sebuah *RESTful API* terdapat beberapa syarat, yaitu sebagai berikut (Tristuti, 2019).

- Menggunakan HTTP method yang benar (GET, PUT, POST, DELETE)
- ENDPOINTS yang digunakan berupa kata benda
- Bersifat stateless
- Menggunakan REST secara benar

Lumen adalah Micro Framework yang diciptakan pengembang Laravel untuk mengakomodasi kebutuhan *developer* yang ingin membuat aplikasi dalam skala lebih kecil dari Laravel. Karena banyak library yang dihilangkan dalam bundle source code, Lumen bisa dijadikan framework untuk membuat REST API (Rizky, 2017)

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, masalah pokok yang akan dibahas adalah :

1. Bagaimana pengembangan *Microservices* dengan framework Laravel Lumen?
2. Bagaimana kegunaan *RESTful API* dengan *Microservices*?
3. Bagaimana penerapan *Microservices* menggunakan php?

1.3 Tujuan dan Manfaat

Tujuan dari Tugas Akhir ini adalah untuk mengetahui kegunaan *RESTful API* dalam *Microservices* dengan framework Laravel Lumen dengan menggunakan bahasa pemrograman PHP.

Manfaat dari Tugas Akhir adalah dapat mempelajari dan mengembangkan *Microservices* dengan framework Laravel Lumen.

1.4 Batasan Masalah

Agar pengerjaan tugas akhir ini menjadi lebih terarah dan mendapatkan hasil yang lebih spesifik, maka batasan masalah pada tugas akhir ini yakni pengujian fungsional dari arsitektur *Microservices* yang dikembangkan bahkan menggunakan kasus dan data dummy.

1.5 State Of The Art

State of the art adalah analisa penelitian yang sudah dilakukan sebelumnya. Untuk mengetahui sejauh mana penelitian sebelumnya dilakukan dan menjadi acuan dari penelitian sekarang ditunjukkan pada Tabel 1

No	Jurnal	Ringkasan	Perbedaan
1	IoT Microservice Deployment in Edge-cloud Hybrid Environment Using	Makalah ini menjelaskan masih ada ruang untuk perbaikan dalam masalah penyebaran layanan mikro untuk sistem IoT pintar. Dalam pekerjaan kami di masa depan, kami akan	Perbedaan jurnal tersebut dengan penelitian ini adalah Aplikasi dibangun dengan menggunakan framework Laravel Lumen. Dan perbedaan lainnya adalah

	<p>Reinforcement Learning, Lulu Chen, Y. X. (2020), 12610-12622.</p>	<p>menyelidiki penyeimbangan beban dalam strategi penyebaran lingkungan multicloud. Kebijakan penyebaran redundansi untuk menyebarkan satu layanan mikro di server tepi yang berbeda juga bisa menjadi perpanjangan dari penelitian kami dalam waktu dekat. Tradeoff antara efisiensi dan biaya membutuhkan analisis yang lebih dalam dalam kasus ini. Pendekatan pelatihan terpusat dapat menyebabkan masalah privasi. Masalah privasi berarti bahwa beberapa informasi terkait dengan privasi, termasuk kebiasaan mengunjungi dan informasi geografis perangkat IoT (Ponsel). Han Qiu mengusulkan metode berdasarkan DWT untuk melindungi keamanan data, namun, kami akan melakukan eksperimen kami dengan cara terdistribusi dalam pekerjaan kami di masa depan.</p>	<p>fitur yang digunakan pada aplikasi tersebut server menggunakan edge-cloud, sedangkan fitur aplikasi ini masih menggunakan localhost.</p>
2	<p>Dyme: Dynamic MicroService Scheduling in Edge Computing Enables IoT , Amit Samanta, J. T. (2020), 6164-6167.</p>	<p>Makalah ini menjelaskan, kami mempresentasikan skema penjadwalan layanan mikro dinamis untuk platform IoT yang mendukung MEC. Pertama, kami mengusulkan model penundaan dan penetapan harga untuk memasok sumber daya yang adil kepada pengguna akhir untuk pelaksanaan tugas layanan mikro yang optimal. Kemudian, kami merancang fungsi utilitas untuk meminimalkan total keterlambatan jaringan dan</p>	<p>Perbedaan jurnal tersebut dengan penelitian ini adalah fitur yang digunakan adalah edge computing yang berfungsi untuk menghemat bandwidth sedangkan pada penelitian ini belum menggunakan fitur tersebut. Dan aplikasi tersebut hanya mengarah pada pemilik usaha. Sedangkan dalam penelitian ini, masih bersifat pengembangan.</p>

		<p>harga jaringan untuk pelaksanaan tugas layanan mikro. Kemudian, kami mengusulkan algoritma penjadwalan layanan mikro dinamis untuk memberikan QoS yang adil dan tingkat kepuasan kepada pengguna akhir. Skema DYME memberikan peningkatan kinerja yang luar biasa dalam hal tingkat kegagalan, pemanfaatan sumber daya dan harga rata-rata. Kami membayangkan mengusulkan skema offloading layanan mikro sadar privasi dalam waktu dekat untuk IoT yang mendukung MEC. Dalam waktu dekat, kami ingin membangun sistem nyata atau prototipe untuk penjadwalan layanan mikro dinamis dengan pengoptimalan berbutir halus yang berbeda. Kami ingin mengatasi masalah penjadwalan layanan mikro adaptif dalam sistem perawatan kesehatan berbasis tepi.</p>	
3	<p>Implementasi Teknologi Mikroservice pada Pengembangan Mobile Learning. Maksy Sendiang, S. K. (2018), Vol 2 No 2.</p>	<p>microservice telah menyebabkan perangkat lunak yang dikembangkan menggunakan teknologi ini dibatasi. Penelitian ini bertujuan untuk menerapkan teknologi microservice dalam pengembangan pembelajaran mobile (MLearning). Implementasi layanan mikro akan memecah aplikasi menjadi entitas yang berdiri sendiri dan terisolasi dengan yang lain. Dengan demikian kegagalan dalam suatu entitas tidak akan mempengaruhi</p>	<p>Perbedaan jurnal tersebut dengan penelitian ini adalah penelitian tersebut hanya menangani pengembangan pembelajaran. Sementara pada penelitian yang dilaksanakan, Aplikasi akan dipergunakan untuk <i>ecommerce</i> dengan menggunakan RESTful API</p>

		<p>aplikasi secara keseluruhan sistem. Penelitian ini menggunakan campuran metode kuantitatif dan kualitatif dalam mengumpulkan dan menganalisis data. Data yang diperoleh dimodelkan dengan pendekatan berorientasi objek dengan menggunakan. Alat Bahasa Pemodelan Terpadu (UML). Metode RUP sebagai salah satu perangkat lunak metode pengembangan yang memberikan fokus pada arsitektur perangkat lunak digunakan untuk mengembangkan Belajar. Dengan penelitian ini diharapkan sebuah MLearning dihasilkan dengan menggunakan teknologi layanan mikro. MLearning ini diharapkan dapat meningkatkan kualitas pendidikan di SMK Negeri Manado. Kesimpulan dari ini penelitian adalah bahwa pengembangan MLearning dengan teknologi microservice akan membentuk aplikasi tangguh yang memberikan dampak positif pada proses belajar mengajar</p> <p>Proses di SMK Negeri Manado</p>	
4	<p>Analisa Implementasi Arsitektur <i>Microservices</i> Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka</p>	<p>Pada penelitian yang ditulis dalam Tesis ini, bertujuan untuk dapat meneliti dari faktor keberhasilan terhadap implementasi arsitektur <i>Microservices</i> kontainer pada komunitas pengembangan perangkat lunak sumber terbuka Opendaylight project. Faktor-faktor tersebut adalah</p>	<p>Perbedaan jurnal tersebut dengan penelitian ini adalah ruang lingkup jaringan yang lebih luas karena terhubung langsung dengan internet. Dimana faktor tersebut adalah <i>source code commit, merge source code, installed dependency, connection</i></p>

	(Opendaylight Devops Community. Putra, R. A. (2019).Vol.9,150-162.	source code commit, merge source code, installed dependency, connection latency to master node, construction Time.	<i>latency to master node, construction time.</i>
5	Perancangan Arsitektur <i>Microservices</i> Untuk Resiliensi Sistem Informasi Perpustakaan Pusat (Studi Kasus UPN “Veteran” Jakarta). Alif Garinda, T. W. (2020), Vol 9, No 2.	Jurnal ini menjelaskan Arsitektur tersebut adalah arsitektur microservice, dirancang untuk membagi service lebih kecil dan terstruktur guna meningkatkan kualitas sistem informasi. Pada sistem informasi perpustakaan pusat di Universitas Pembangunan Nasional “Veteran” Jakarta, arsitektur ini belum digunakan dan masih menggunakan arsitektur monolitik.	Perbedaan jurnal tersebut dengan penelitian ini adalah penelitian tersebut dibuat untuk membangun aplikasi perpustakaan pusat. Sedangkan aplikasi tersebut dibangun menggunakan <i>Node.js, Docker, MongoDB</i> sebagai database NoSQL. Sementara pada penelitian yang akan dilaksanakan dibangun dengan Laravel Lumen dan RESTful API.

1.6 Sistematika Penulisan

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bab yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini menerangkan secara singkat dan jelas mengenai latar belakang penulisan penelitian tugas akhir, rumusan masalah, tujuan dan manfaat, batasan permasalahan, metodologi yang digunakan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan pengertian tentang teori dasar RESTful API, *Microservices*, MySQL, Laravel Lumen.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis permasalahan dan desain aplikasi, dengan menggunakan RESTful API, *Microservices*, Laravel Lumen.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi penggunaan *Microservices* pada aplikasi, kemudian akan dilakukan pengujian untuk mengetahui apakah hasilnya sudah sesuai dengan kebutuhan.

BAB V PENUTUP

BAB ini membahas kesimpulan dan saran