

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Microservices merupakan sebuah rancangan arsitektur yang sedang populer dalam beberapa waktu belakangan ini, hal ini dikarenakan arsitektur ini memiliki beberapa kelebihan yang signifikan dibanding dengan rancangan arsitektur monolitik. *Microservices* telah menjadi desain arsitektur populer untuk sebuah sistem perangkat lunak yang diimplementasikan dalam serangkaian *service* kecil, dan saat ini *microservices* telah banyak diadopsi oleh beberapa perusahaan teknologi seperti Netflix, Amazon, dan Google (Cortellessa et al., 2022).

Arsitektur *microservices* merupakan bagian dari *Service Oriented Architecture (SOA)* yang memiliki prinsip mengelola sistem sebagai kumpulan *loosely-coupled service*, yang artinya komponen tidak bergantung secara langsung antara satu dengan yang lainnya sehingga suatu perubahan dalam komponen tidak akan memberikan efek yang signifikan terhadap komponen lainnya. Kebutuhan aplikasi akan sistem yang ringan dan skalabilitas tinggi dapat disediakan oleh arsitektur *microservices*. Arsitektur ini membagi *logic* menjadi sejumlah *service* independen yang berjalan sebagai proses terpisah. Pendekatan ini dapat mengorganisir realisasi dari *Service Oriented Architecture (SOA)* yang memungkinkan proses bisnis dan data menjadi lebih tersedia (Akbulut & Perros, 2019).

Kebutuhan bisnis di era saat ini membutuhkan sebuah sistem yang memiliki performa tinggi serta aman untuk digunakan. Arsitektur monolitik merupakan salah satu model dalam pengembangan aplikasi dimana semua fungsi bisnis ditumpuk menjadi satu aplikasi tunggal dan dibangun sebagai satu kesatuan (Indrasiri & Siriwardena, 2018). Akan tetapi seiring dengan berjalanya waktu sebuah sistem akan semakin berkembang dan muncul fitur - fitur baru yang nantinya akan diimplementasikan, dengan arsitektur monolitik hal ini akan sulit dilakukan karena seluruh aplikasi merupakan beberapa komponen yang tergabung menjadi satu kesatuan. Setiap dari komponen menawarkan fungsi bisnis yang luas dan saling terkait satu sama lain, menambahkan atau memodifikasi fungsionalitas pada komponen sangat tidak mudah karena sifatnya yang monolitik. Untuk menyediakan seluruh persyaratan bisnis, komponen tersebut harus berkomunikasi dengan komponen lain. Oleh karena itu memodifikasi atau mengganti

komponen akan menjadi sesuatu yang rumit untuk dilakukan (Indrasiri & Siriwardena, 2018). Hal ini dapat ditangani dengan menggunakan arsitektur *microservices*, inti dari arsitektur *microservices* adalah mengembangkan suatu aplikasi sebagai rangkaian layanan kecil dan independen yang berjalan sendiri (Indrasiri & Siriwardena, 2018). Keuntungan dari menggunakan arsitektur *microservices* diantaranya yaitu setiap *service* dapat dikembangkan secara terpisah tanpa harus mengganggu *service* lain, mudah untuk memodifikasi atau mengganti setiap *service* yang ada, serta proses *scalability* dan *deployment* sistem berbasis *microservices* lebih mudah untuk diintegrasikan dibanding arsitektur monolitik (Karwatka et al., 2018).

Sehubungan dengan kelebihan yang ditawarkan dari arsitektur *microservices*, maka dalam Tugas Akhir ini akan dilakukan implementasi pembuatan sistem menggunakan arsitektur *microservices* dengan teknologi antara lain *framework* Laravel sebagai *back end*, *framework* React.js dan React Native sebagai *front end* sedangkan untuk *database* yang digunakan adalah MySQL dengan contoh kasus *e-commerce*.

1.2 Perumusan Masalah

Berdasarkan masalah yang telah dijelaskan dalam latar belakang, maka rumusan masalah dalam Tugas Akhir ini adalah bagaimana mengimplementasikan arsitektur *microservices*?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari Tugas Akhir ini sesuai dengan rumusan masalah yang telah disebutkan sebelumnya yaitu mengimplementasikan arsitektur *microservices*.

1.4 Batasan Masalah

Dalam Tugas Akhir ini diterapkan beberapa batasan - batasan pembahasan agar tidak keluar dari topik yang sudah dijelaskan sebelumnya. Batasan dalam Tugas Akhir ini adalah sebagai berikut :

1. Paradigma pemrograman yang dipakai yaitu *Object Oriented Programming*.
2. Topik yang diangkat dalam penelitian ini merupakan contoh kasus *e-commerce*.
3. Sistem web *e-commerce* yang dibangun bukan merupakan sistem utuh secara keseluruhan, melainkan hanya menerapkan beberapa fungsi dari *e-commerce*.

4. Sistem *mobile* yang dibangun hanya berisi *module get product* saja dikarenakan kebutuhannya hanya sebagai pembuktian bahwa *microservices* yang telah dibangun mampu berjalan bersamaan secara *multi-platform*.
5. Data yang dikonsumsi pada *client* berupa data *JSON*.
6. Komunikasi yang digunakan menggunakan *Restful API*.
7. Pengujian yang dilakukan adalah *API testing* untuk menguji *service*, *black box testing* dan *unit testing* untuk menguji interaksi dari aplikasi *client front end*.
8. *Deployment* dari arsitektur *microservices* yang dikembangkan menggunakan jaringan yang sama yaitu jaringan *Local Area Network (LAN)*.

1.5 State of The Art

Beberapa penelitian yang menjadi acuan dalam penulisan Tugas Akhir ini adalah sebagai berikut :

Tabel 1. 1 State of The Art

No.	Judul Jurnal	Pembahasan
1	<p><i>Overcoming Security Challenges in Microservices Architectures</i></p> <p>Peneliti : Tetiana Yarygina, Anya Helene Bagge</p> <p>Jurnal : <i>Proceedings - 12th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2018 and 9th International Workshop on Joint Cloud Computing, JCC 2018</i></p> <p>Tahun : 2018</p>	<p><u>Hasil Penelitian:</u></p> <p>Penelitian ini mengidentifikasi dan menguraikan beberapa isu seputar keamanan <i>microservices</i> dengan pendekatan holistik, untuk mengatasi kurangnya pedoman pemahaman mengenai keamanan <i>microservices</i>, penelitian ini menjelaskan desain dan implementasi kerangka kerja keamanan sederhana untuk <i>microservices</i> yang dapat dimanfaatkan oleh para praktisi. Disisi lain jika isu keamanan ini</p>

	<p>Penerbit : IEEE</p> <p>DOI : 10.1109/SOSE.2018.00011</p> <p>ISBN : 9781538652060</p>	<p>dapat teratasi maka akan meningkatkan keamanan dari arsitektur <i>microservices</i> serta terciptanya sistem yang <i>robust</i>.</p> <p><u>Alasan Menjadi Tinjauan Penelitian:</u></p> <p>Mengambil penerapan pendekatan keamanan <i>authentication</i> dan <i>authorization</i> yang berbasis <i>Token-based</i> dengan menggunakan <i>JSON security standards</i> berupa <i>JSON Web Token (JWT)</i> yang dikombinasikan dengan teknologi Laravel sehingga keamanan sistem menjadi semakin optimal.</p>
2	<p><i>Performance Analysis of Microservices Design Patterns</i></p> <p>Peneliti : Akhan Akbulut, Harry G. Perros</p> <p>Jurnal : <i>IEEE Internet Computing</i></p> <p>Tahun: 2019</p> <p>Volume : 23</p> <p>Penerbit :</p>	<p><u>Hasil Penelitian:</u></p> <p>Penelitian ini membahas tentang hasil performa untuk beberapa sektor seperti <i>query response time</i>, <i>efficient hardware usage</i>, <i>hosting costs</i>, dan <i>packet loss rate</i> dalam tiga pola desain <i>microservices</i> yang telah dipraktekkan dalam dunia industri, beberapa pola desain yang digunakan yaitu <i>API Gateway design pattern</i>, <i>Chain of Responsibility design pattern</i>,</p>

	<p>IEEE</p> <p>DOI :</p> <p>10.1109/MIC.2019.2951094</p>	<p>dan <i>Asynchronous Messaging design pattern</i>.</p> <p><u>Alasan Menjadi Tinjauan Penelitian:</u></p> <p>Menggunakan <i>design pattern</i> yang telah dilakukan pada jurnal tersebut yaitu <i>API Gateway design pattern</i> untuk dijadikan acuan dalam membangun <i>API Gateway</i> pada sistem yang sedang dikembangkan.</p>
3	<p>Identifikasi Kandidat <i>Microservices</i> Dengan Analisis <i>Domain Driven Design</i></p> <p>Peneliti :</p> <p>I Wayan Adi Juliawan Pawana, Dewa Made Wiharta, Nyoman Putra Sastra</p> <p>Jurnal :</p> <p>Majalah Ilmiah Teknologi Elektro</p> <p>Tahun :</p> <p>2021</p> <p>Penerbit :</p> <p>Magister Teknik Elektro Universitas Udayana</p> <p>Volume :</p> <p>20, No.2</p> <p>DOI:</p> <p>https://doi.org/10.24843/MITE.2021.v20i02.P11</p>	<p><u>Hasil Penelitian:</u></p> <p>Penelitian ini membahas tentang proses migrasi Sistem Informasi Dosen (SIMDOSEN) pada universitas Udayana dari arsitektur <i>monolithic</i> ke <i>microservices</i>, serta membahas bagaimana memisahkan <i>domain</i> bisnis dengan metode <i>Domain Driven Design</i> yang nantinya hasil dari identifikasi ini akan dijadikan acuan dalam menentukan kandidat dari <i>service-service</i> yang ada.</p> <p><u>Alasan Menjadi Tinjauan Penelitian:</u></p> <p>Dengan melihat hasil dari identifikasi <i>domain</i> bisnis dengan menggunakan metode <i>Domain</i></p>

	<p>ISSN : 1693 – 2951</p>	<p><i>Driven Design</i> dalam menentukan kandidat <i>service</i> sehingga menghasilkan pengembangan <i>microservices</i> yang sistematis dan dapat dipahami, maka dari itu pada penelitian ini akan diterapkan metode yang sama yaitu <i>Domain Driven Design</i> sebagai acuan dalam menentukan kandidat <i>service</i> dalam membangun sistem.</p>
4	<p><i>Research on Unified Authentication and Authorization in Microservices Architecture</i></p> <p>Peneliti : Yang Shu Lin, Hu Jie Ping</p> <p>Jurnal : <i>International Conference on Communication Technology Proceedings, ICCT</i></p> <p>Tahun : 2020</p> <p>Volume : 2020-October</p> <p>Penerbit : IEEE</p> <p>DOI : 10.1109/ICCT50939.2020.9295931</p>	<p><u>Hasil Penelitian:</u></p> <p>Penelitian ini membahas tentang bagaimana merancang solusi <i>access control</i> yang berupa <i>Authorization Policies</i>, <i>role</i> dan <i>permission</i>, dan <i>middleware</i> untuk dapat memenuhi kebutuhan <i>authentication</i> serta meningkatkan efisiensi dalam verifikasi <i>authorization</i> pada arsitektur <i>microservices</i> dengan menggunakan teknologi <i>OAuth2</i> dan <i>JSON Web Token</i>.</p> <p><u>Alasan Menjadi Tinjauan Penelitian:</u></p> <p>Menerapkan konsep <i>Authorization policies</i> dengan <i>JWT (JSON Web Token)</i> untuk</p>

	<p>ISBN : 9781728181417</p>	<p>menentukan <i>role</i> dan <i>permissions</i> dari <i>user</i> serta menerapkan sistem <i>middleware</i> untuk mengamankan akses <i>endpoint</i> dari <i>API</i> yang ada.</p>
5	<p><i>Introduction to microservices API patterns (MAP)</i></p> <p>Peneliti : Olaf Zimmermann, Mirko Stocker, Daniel Lübke, Cesare Pautasso, Uwe Zdun</p> <p>Jurnal : <i>OpenAccess Series in Informatics</i></p> <p>Tahun : 2020</p> <p>Volume : 78</p> <p>Penerbit : Oasies</p> <p>DOI : 10.4230/OASICS.Microservices.2017-2019.4</p> <p>ISSN : 21906807</p> <p>ISBN : 9783959771375</p>	<p><u>Hasil Penelitian:</u></p> <p>Dalam penelitian ini dibahas tentang bagaimana membuat pola desain untuk <i>API</i> yang dapat diterapkan dalam arsitektur <i>microservices</i>, dimana terdapat dua pola desain yang dapat diterapkan berdasarkan penelitian ini yaitu <i>Embedded Entity</i> dan <i>Linked Information Holder</i>. Dua pola desain ini dapat diterapkan apabila hendak membuat <i>API</i> yang berisi informasi data berupa <i>nested data</i> atau data yang memiliki keterkaitan dengan entitas data lain.</p> <p><u>Alasan Menjadi Tinjauan Penelitian:</u></p> <p>Menerapkan pola desain <i>Embedded entity</i> dalam membangun <i>API</i> untuk pertukaran data antar <i>service</i> yang diterapkan dalam bahasa pemrograman <i>PHP</i></p>

		menggunakan <i>framework</i> Laravel dengan menggunakan <i>database</i> MySQL.
--	--	--

1.6 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini dibagi menjadi beberapa bab yang disusun sebagai berikut :

BAB 1 PENDAHULUAN

Pembahasan dalam bab ini mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, ruang lingkup, metodologi penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Pembahasan dalam bab ini mengenai berbagai teori yang berhubungan dengan penelitian Tugas Akhir.

BAB 3 METODE

Pembahasan dalam bab ini mengenai metodologi yang diterapkan dalam membangun sistem.

BAB 4 PEMBAHASAN

Pembahasan dalam bab ini mengenai hasil dari penerapan metodologi yang telah dijabarkan dalam bab 3.

BAB 5 KESIMPULAN DAN SARAN

Pembahasan dalam bab ini mengenai kesimpulan dari seluruh proses yang telah dilakukan dan saran untuk pengembangan penelitian dimasa mendatang.