CoMBInES
CONFERENCE ON MANAGEMENT, BUSINESS, INNOVATION, EDUCATION AND SOCIAL SCIENCES

Volume 1 No 1 (2021)

Diterima : February 01, 2021
Disetujui : February 05, 2021
Diterbitkan: February 24, 2021

Conference on Management, Business, Innovation, Education and Social Science
https://journal.uib.ac.id/index.php/combines

Analisa dan Implementasi Microservice pada Container Menggunakan Docker

Stefanus Eko Prasetyo[1], Ardyansyah Wijaya[2]
Email korespndensi : stefanus.eko@uib.edu, 1731024.ardyansyah@uib.edu

[1]Fakultas Ilmu Komputer, Universitas Internasional Batam, Batam, Indonesia
[2]Fakultas Ilmu Komputer, Universitas Internasional Batam, Batam, Indonesia

Abstrak
Umumnya sebuah organisasi maupun individu yang membangun perangkat menggunakan konsep pembuatan dengan arsitektur monolitis dan untuk menghemat resources biasanya diaplikasikan pada virtualisasi berbasis hypervisor. Tetapi arsitektur monolitis memiliki permasalahan pada fleksibilitas dan skalabilitas. Pada penelitian yang dilakukan oleh Jaramillo, Nguyen, & Smart menghasilkan data yang menunjukkan bahwa penggunaan Docker dengan bantuan alat-alat lainnya dapat meningkatkan efisiensi serta otomatisasi pada arsitektur aplikasi microservices. Oleh karena itu, penulis merancang sebuah penilitian yang mengaplikasikan konsep microservices sebagai pengganti arsitektur monolitis pada container menggunakan Docker. Kemudian penulis akan melakukan pengujian performa berbasis Load Testing menggunakan aplikasi Apache JMeter. Hasil implementasi menunjukkan kemudahan dalam penggunaan dan aplikasi microservice pada Docker dan stabilitas yang baik saat pengujian berbagai load menggunakan Apache Jmeter.

Kata Kunci:
Microservice , Container , Docker

Pendahuluan
Pada umumnya, sebuah organisasi maupun individu yang membangun perangkat lunak atau software saat ini menggunakan konsep pembuatan dengan arsitektur monolitis. Arsitektur monolitis adalah jenis pembuatan software dimana sistem software tersebut memiliki fungsi yang banyak bahkan semua fungsi dapat dimasukkan dalam satu sistem software tersebut. Software yang menggunakan arsitektur monolitis ini masih dapat dikategorikan baik bila diimplementasikan dalam grup atau organisasi yang berskala kecil. Tetapi jika sebuah perusahaan atau organisasi besar yang membutuhkan fleksibilitas dan skalabilitas yang baik pada perangkat lunaknya untuk meningkatkan daya saing perusahaan, maka arsitektur monolitis akan menjadi hambatan besar(Villamizar et al., 2015).

Untuk melakukan perubahan seperti update akan menjadi semakin sulit pada software monolitis seiring waktu karena basis coding pada software akan menjadi semakin besar dan

rumit dengan semakin banyaknya orang yang bekerja dan mengubahnya. Oleh karena itu, setiap perubahaan, patch, ataupun update akan menjadi semakin lama.

Microservices diperkenalkan sebagai solusi untuk mengatasi masalah yang terdapat pada arsitektur monolitis. Microservices merupakan konsep dimana sebuah software dipecahkan dan dibagi ke dalam komponen yang lebih kecil dan memiliki fungsinya masing-masing (Cao, 2020). Pada arsitektur microservices, setiap microservice memiliki peran dan fungsi yang berbeda. Perubahan yang dilakukan pada satu microservice tidak selalu mengakibatkan perubahan pada bagian lainnya sehingga dapat mempermudah skalabilitas sistem aplikasi sebuah organisasi. Pengembangan atau software development pada arsitektur microservices juga menjadi lebih mudah karena setiap microservice bersifat independen, dapat menggunakan bahasa pemograman yang berbeda, dapat dijalakankan pada sistem operasi yang berbeda, dan dapat menggunakan jenis database yang berbeda. Untuk melakukan implementasi microservice yang efisien, umumnya dapat dikemas dan dijalankan pada container yang dapat meningkatkan skalabilitas sebuah aplikasi (Wan, Guan, Wang, Bai, & Choi, 2018).

Docker adalah salah satu open platform virtualisasi container yang dilengkapi dengan seperangkat alat yang dapat membantu developer dalam megimplementasi dan mengelola container dengan lebih mudah dan menggunakan resource yang lebih rendah dibanding dengan virtualisasi VM (Abraham, Dutta, Kumar, Abhishek, & Dutta, 2018). Container Docker merupakan sebuah platform untuk membangun aplikasi dan menempatkan semua komponen yang dibutuhkan ke dalam platform tersebut. Penggunaan resource pada Docker rendah diakibatkan karena setiap instance container yang dibangun tidak memiliki guest OS seperti virtualisi hypervisor (Khalida, Muhajirin, & Setiawati, 2019).

Dengan alasan di atas yang telah dijelaskan oleh penulis, maka penulis akan merancang sebuah analisa yang menilai performa microservice yang diimplementasikan pada container dengan menggunakan Docker. Agar dapat memperjelas fungsi dan kemampuan microservice yang diaplikasikan dalam container, maka penulis akan membuatkan sebuah laporan penelitian dengan judul "Analisa dan Implementasi Microservices pada Container dengan Menggunakan Docker".

Tinjauan Pustaka

Penelitian yang berasal dari perusahaan IBM dan oleh (Jaramillo, Nguyen, & Smart, 2016) yang berjudul "Leveraging microservices architecture by using Docker technology" menjelaskan kesulitan yang ditemukan dalam penggunakan arsitektur monolitis seperti masalah fleksibilitas dan skalabilitas. Pada penelitian ini dibahas kelebihan dan kekurangan yang diperoleh dari penggunakan microservices dan juga bagaimana Docker dapat membantu dalam implementasi arsitektur ini. Hasil dari penelitian menjelaskan bahwa implementasi microservice dengan Docker dan dibantu dengan alat lainnya dapat meningkatkan efisiensi serta otomatisasi pada arsitektur microservice ini.

Penelitian kedua dilakukan oleh (Bucchiarone, Dragoni, Dustdar, Larsen, & Mazzara, 2018) dengan judul "From Monolithic to Microservices: An Experience Report from the Banking Domain". Penelitian berbasis studi kasus ini dilakukan pada Danske Bank yang merupakan bank terbesar di Denmark. Penelitian ini membahas tentang perubahan skalabilitas dalam mengimplementasikan program arsitektur monolitis menjadi microservices. Studi kasus ini didasarkan pada sistem FX Core yang merupakan sistem konversi mata uang. Sistem microservices yang dirancang kemudian dijalankan pada container Docker dan memiliki sistem

otomatisasi. Hasil dari penelitian menunjukkan bahwa implementasi arsitektur microservice memberikan banyak keuntungan seperti skalabilitas yang baik, replikasi service yang mudah, dan keuntungan lainnya yang didapatkan pada arsitektur microservices. Tetapi dengan ini juga memunculkan beberapa permasalahan baru pada mekanisme, penanganan, dan monitoring yang sebelumnya tidak ditemukan pada arsitektur monolitis.

Penelitian ketiga dilakukan oleh (Bashari Rad, John Bhatti, & Ahmadi, 2017) dengan judul "An Introduction to Docker and Analysis of its Performance". Penelitian ini membahas tentang teknologi yang didapatkan dengan menggunakan Docker sebagai container virtualization dan juga analisa performanya. Dalam penelitian ini dibahas kecepatan boot time dan kalkulasi kecepatan operasional dengan menggunakan bahasa pemograman python pada Docker dibanding dengan container virtualization lainnya. Hasil dari penelitian menunjukkan bahwa docker menambahkan layer ekstra sebagai host operating system sehingga memiliki kecepatan performa yang lebih baik dibanding dengan virtual machines dan Docker juga memiliki keunggulan performa dibanding dengan container virtualization lainnya.

Penelitian keempat dilakukan oleh (Herrera-Izquierdo & Grob, 2017) dengan judul "A performance evaluation between Docker container and Virtual Machines in cloud computing architectures". Pada penelitian ini dilakukan perbandingan performa antara Virtual Machine dengan Docker container pada segi performa CPU, RAM, network, dan disk. Hasil peneltian menunjukkan bahwa teknologi Docker container memiliki efisiensi penggunaan hardware yang lebih baik dan memiliki performa yang lebih unggul dibanding dengan teknologi virtual machine.

Penelitian kelima yang dilakukan oleh (Permatasari, 2020) berjudul "Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian". Penelitian ini membahas dan menguji performa aplikasi pertanian MeTANI dengan konsep Black Box dan Gorilla Testing dengan aplikasi Apache JMeter sebagai load test toolnya. Hasil testing didapatkan berupa tabel yang dihasilkan dari aplikasi Apache JMeter dan juga grafik yang dihasilkan dari diagnosa Microsoft Visual Studio. Penelitian ini menunjukkan bahwa aplikasi MeTANI dapat bekerja dengan baik pada load kecil maupun besar sehingga load testing menggunakan Apache JMeter menghasilkan data yang baik.

Berdasarkan beberapa hasil penelitian diatas yang telah dijelaskan oleh penulis sebelumnya, maka penulis akan merancang sebuah penelitian dengan implementasi sebuah microservice pada Docker dan melakukan analisa performa dengan menggunakan metode Load Testing. Penulis menerapkan konsep microservice berdasarkan keunggulan yang didapatkan dari hasil penelitian yang dilakukan oleh Jaramillo, Nguyen, & Smart, (2016) dan Bucchiarone, Dragoni, Dustdar, Larsen, & Mazzara, (2018) dengan penggunaan Docker sebagai container virtualization service yang memiliki performa yang lebih unggul dibanding virtualization service lainnya seperti hasil penelitian yang dilakukan oleh Bashari Rad, John Bhatti, & Ahmadi, (2017) dan Herrera-Izquierdo & Grob, (2017). Penulis kemudian akan melakukan uji performa menggunakan metode Load Testing dengan menggunakan aplikasi Apache JMeter seperti yang dilakukan oleh Permatasari, (2020).

Metodologi Penelitian

Uji coba dilakukan dua kali dengan konfigurasi pertama sebanyak 200 users yang melakukan request baru setiap 0,05 detik dan diulang sebanyak 10 kali sehingga menghasilkan 2000 hasil data. Pengujian kedua menggunakan konfigurasi sebanyak 2000 user yang melakukan request baru setiap 0,05 detik dan diulang sebanyak 10 kali sehingga menghasilkan

20000 hasil data. Metode pengujian yang diterapkan penulis dilakukan dengan cara membangun sebuah container menggunakan Docker kemudian container tersebut akan diaplikasikan sebuah jenis microservice yang kemudian akan dilakukan Load Testing pada container tersebut menggunakan Apache JMeter.

Adapun kebutuhan hardware yang digunakan dalam pengujian ini adalah sebuah komputer sebagai sarana pemasangan Docker Desktop dan Load Testing.

Kebutuhan software untuk mengimplementasikan microservice pada Docker adalah sebagai berikut:

1. Docker Desktop versi 2.5.0.1 sebagai aplikasi virtualisasi berbasis container
2. Apache Jmeter versi 5.3 sebagai aplikasi pengujian Load Testing pada container
3. Google Chrome browser sebagai sarana untuk mengakses aplikasi resource monitor container.

Hasil dan Pembahasan

penulis akan melakukan pengujian performa microservice tersebut pada container dengan metode *Load Testing* menggunakan Apache JMeter.

Tabel 1. Hasil Pengujian Response Time

Load Test	Response Time					
	200 User			2000 User		
	Min	Max	Avg	Min	Max	Avg
Docker	0.5	11	1	0.5	34	1

Tabel 2. Hasil Pengujian Standard Deviation

Load Test	Deviation	
	200 User	2000 User
Docker	1.40	1.36

Tabel 3 Hasil Throughput

Server	Throughput	
	200 User	2000 User
Docker	200.8	200.1

Tabel 4 Hasil Error

Server	Error					
		200 User			2000 User	
	Total Req.	Comp. Req.	Error (%)	Total Req.	Comp. Req.	Error (%)
Docker	2000	2000	0	20000	20000	0.00

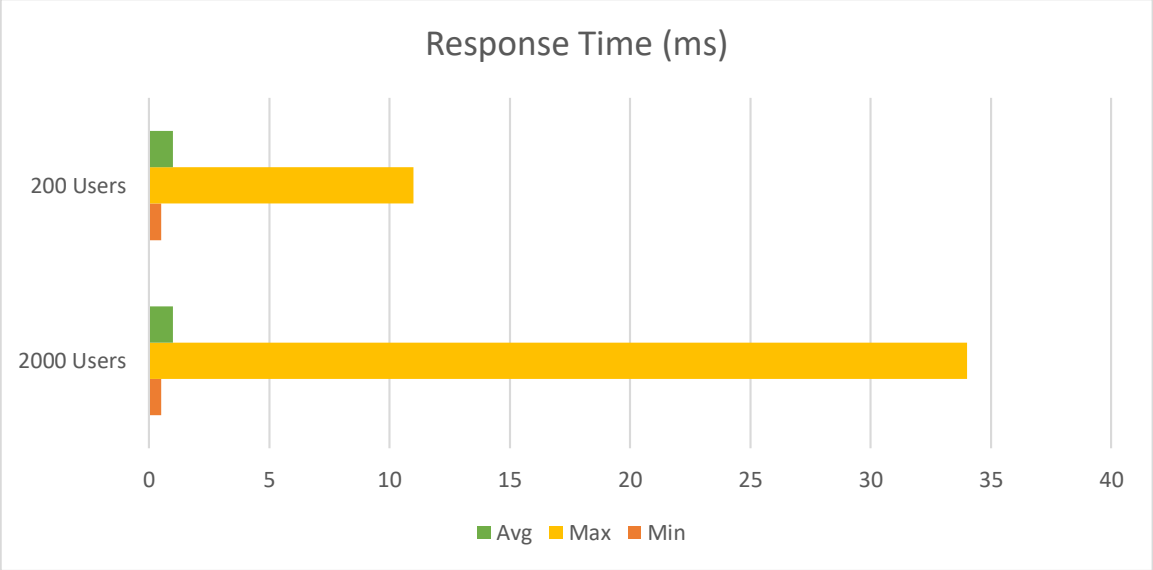
Tabel 5 Hasil Pengujian RAM dan CPU

RAM Utilization(MBs)		CPU Utilization(Cores)	
Idle	Running(High)	Idle	Running(High)
546.8	547.09	0.01	0.1148
446.52	596.16	0.05	0.7259

Pembahasan

Berdasarkan hasil data yang diperoleh dari Load Test, dapat dijelaskan sebagai berikut:

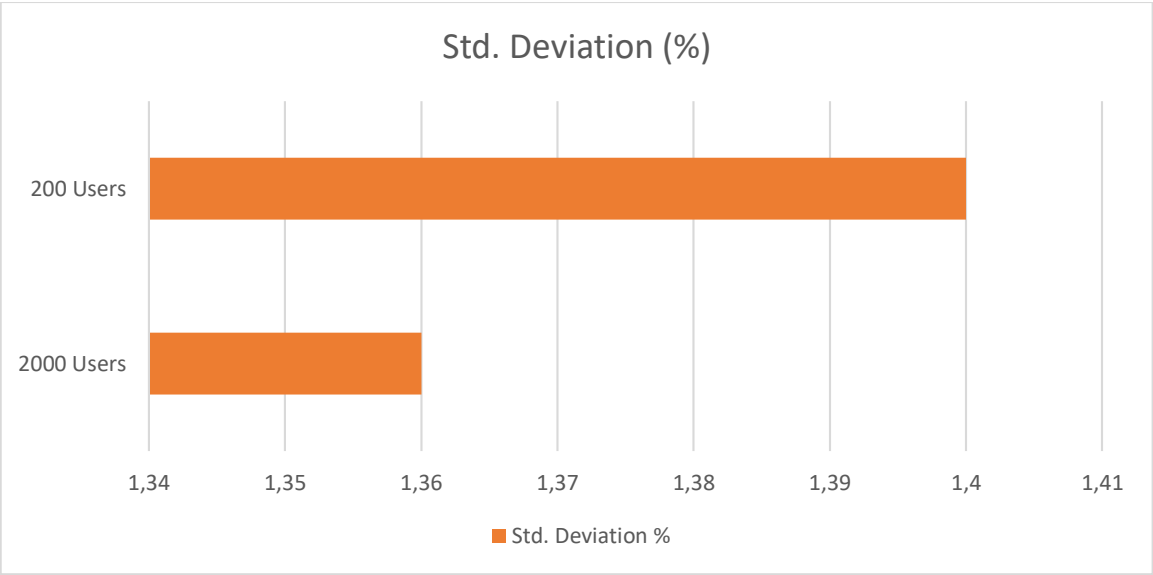
1. Response Time



Gambar 1 Chart Response Time (ms)

Pengujian yang dilakukan pertama dengan load sebanyak 200 user dan yang kedua sebanyak 2000 user menampilkan hasil response time yang tidak jauh berbeda. Kedua pengujian menunjukkan average response time setinggi 1ms walaupun dengan max response time 11 dibanding 24. Hal ini menunjukkan bahwa Container memiliki kestabilan performa yang baik dengan berbagai jumlah load yang ditangani.

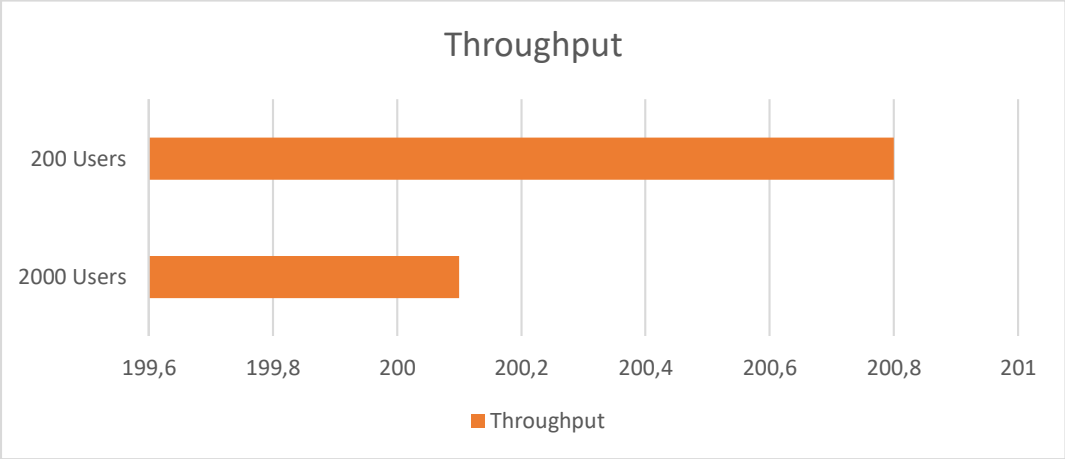
2. Deviation



Berdasarkan data hasil pengujian dapat dilihat bahwa deviasi standar pada kedua pengujian menunjukkan angka yang mirip dengan poin 1.40 pada 200 user dan 1.36 pada 2000

user load. Hasil data ini menunjukkan bahwa walaupun dengan load user yang besar, container Docker tetap dapat memberikan performa yang stabil.

3. Throughput

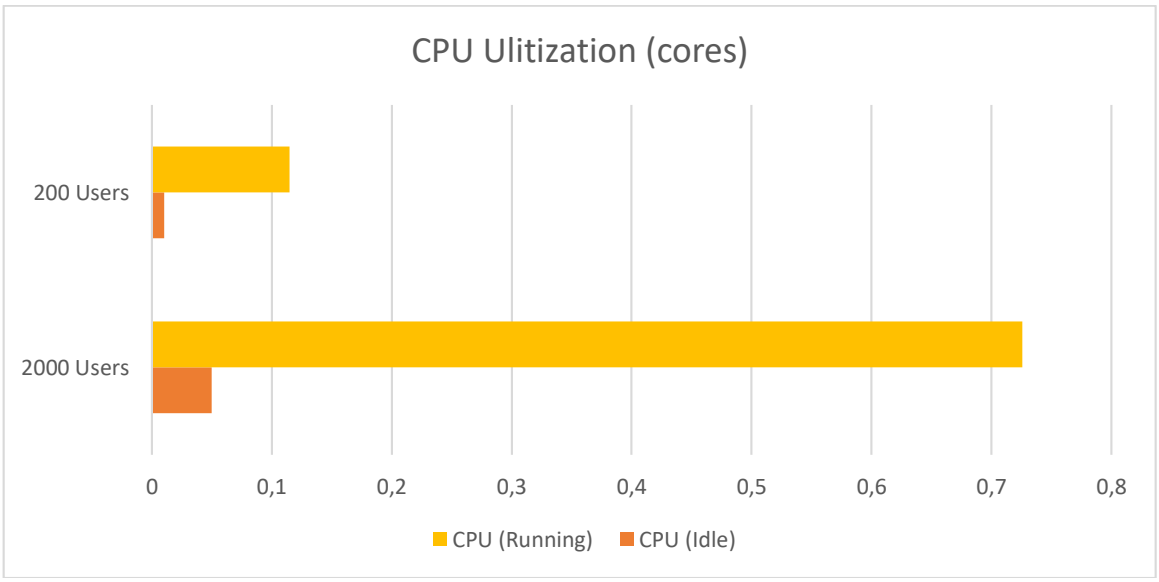


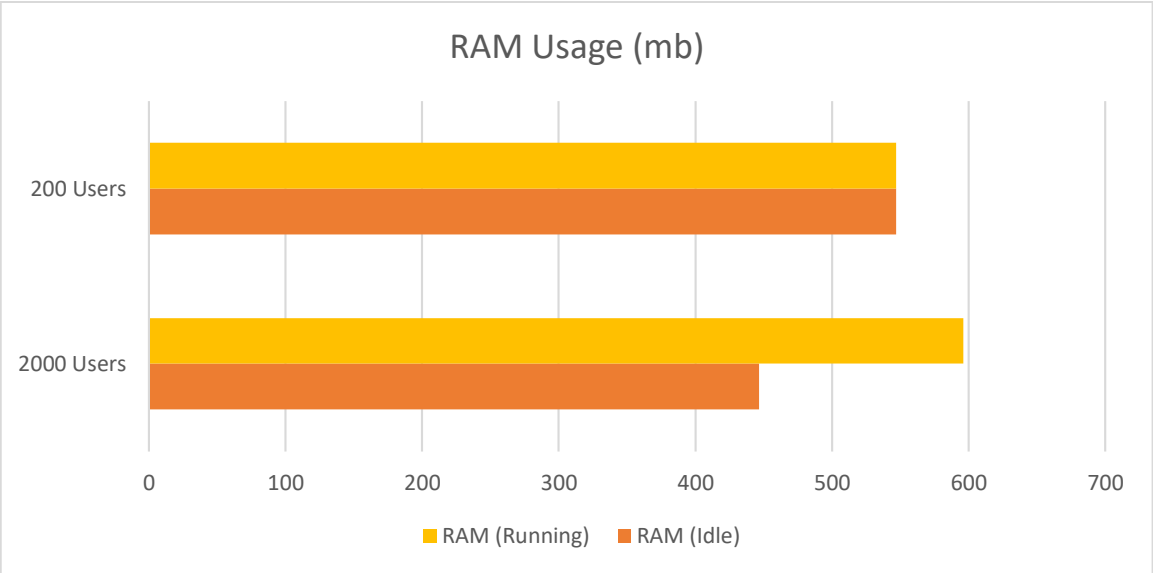
Hasil data Throughput yang diperoleh dari 2 pengujian di atas dengan 200.8/sec pada 200 user dan 200.1/sec pada 2000 user menunjukkan performa yang dihasilkan pada kondisi load yang berbeda tetap relatif stabil.

4. Error

Berdasarkan data yang dihasilkan, dapat dilihat bahwa dalam kondisi normal, Docker dengan load yang berbeda memiliki persentase error 0.00% atau tidak memiliki error. Hal ini menunjukkan bahwa, Docker sangat baik sebagai sarana implementasi aplikasi berarsitektur microservice yang memiliki tingkat data transfer yang banyak dan luas.

5. RAM dan CPU Utilization





Berdasarkan data yang dihasilkan pada pengujian, dapat dilihat bahwa pada pengujian load sebesar 200 user, penggunaan RAM memiliki peningkatan yang sangat sedikit dan penggunaan CPU yang meningkat sebesar 11%. Sedangkan pada pengujian kedua dengan load sebesar 2000 user, penggunaan RAM mengalami peningkatan hingga 33% dan penggunaan CPU yang meningkat dari 0.05 cores pada saat idle menjadi 0.7259. Penggunaan resource yang meningkat banyak dari pengujian pertama hingga kedua merupakan hal yang wajar karena jumlah user load yang bertambah 10 kali lipat. Walaupun peningkatan resource yang signifikan tetapi Docker tetap dapat memberikan kestabilan pada penggunaan dan performanya.

Melaporkan tentang penemuan dari penelitian anda berdasarkan pada informasi yang dikumpulkan dari hasil metodologi penelitian. Pembahasan adalah bagian yang paling penting untuk menjelaskan hasil dari penelitian termasuk hasil olah data dan analisisnya. (Font Tahoma 12, spasi 1, 300-1000 kata).

Kesimpulan

Berdasarkan penelitian yang dilakkan penulis dengan judul "Analisa dan Implementasi Microservice pada Container Menggunakan Docker" maka penulis akan menjelaskan kesimpulan penelitian ini sebagai berikut:

1. Penggunaan arsitektur microservice dapat mengatasi masalah skalabilitas dan masalah lainnya yang terdapat pada arsitektur monolitis tetapi juga terdapat beberapa kelemahan pada arsitektur ini.
2. Penggunaan Docker sebagai virtualisasi berbasis container memiliki efisiensi yang lebih baik dibanding dengan virtualisasi hypervisor
3. Pengujian load testing yang dilakukan pada container Docker yang berisikan microservice menampilkan penggunaan resource yang relatif rendah dan stabilitas yang baik.

Daftar Pustaka

Abraham, A., Dutta, P., Kumar, J., Abhishek, M., & Dutta, S. (2018). *Emerging Technologies in Data Mining and Information* (Vol. 1). Retrieved from https://link.springer.com/book/10.1007/978-981-13-1501-5

Bashari Rad, B., John Bhatti, H., & Ahmadi, M. (2017). An Introduction to Docker and Analysis of its Performance. *IJCSNS International Journal of Computer Science and Network Security*, *17*(3), 228–235.

Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S. T., & Mazzara, M. (2018). From Monolithic to Microservices: An Experience Report from the Banking Domain. *IEEE Software*, *35*(3), 50–55.

Cao, J. (2020). Design on deployment of microservices on container-based cloud platform. *Journal of Physics: Conference Series*, *1624*(6).

Herrera-Izquierdo, L., & Grob, M. (2017). A performance evaluation between Docker container and Virtual Machines in cloud computing architectures. *Maskana*, 127–133.

Jaramillo, D., Nguyen, D. V., & Smart, R. (2016). Leveraging microservices architecture by using Docker technology. *Conference Proceedings - IEEE SOUTHEASTCON*, *2016-July*, 0–4.

Khalida, R., Muhajirin, A., & Setiawati, S. (2019). Teknis Kerja Docker Container untuk Optimalisasi Penyebaran Aplikasi. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, *7*(2), 167–176. https://doi.org/10.33558/piksel.v7i2.1819

Permatasari, D. I. (2020). Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, *8*(1), 135. https://doi.org/10.26418/justin.v8i1.34452

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., & Gil, S. (2015). Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud Evaluando el Patrón de Arquitectura Monolítica y de Micro Servicios Para Desplegar Aplicaciones en la Nube. *10th Computing Colombian Conference*, 583–590.

Wan, X., Guan, X., Wang, T., Bai, G., & Choi, B. Y. (2018). Application deployment using Microservice and Docker containers: Framework and optimization. *Journal of Network and Computer Applications*, *119*(December 2017), 97–109. https://doi.org/10.1016/j.jnca.2018.07.003