1. Compare

Menurut penelitian yang dilakukan oleh Sani et al., (2019) bahwa peneliti menggunakan arsitektur *microservice* untuk meningkatkan aplikasi yang telah dibuat sebelumnya yang menggunakan arsitektur *monolithic* karena *microservice* memiliki kelebihan seperti skalabilitas dan kecepatan yang baik. *Microservice* dapat memecah komponen menjadi beberapa komponen kecil yang nantinya komponen-komponen kecil ini bergertak secara independen, yang berarti jika ada perubahan atau gangguan komponen ini tidak mengganggu kinerja komponen lain secara langsung. Hal ini dilakukan juga di penelitian lainnya dimana peneliti melakukan pengembangan dengan arsitektur *microservice* untuk membandingkan hasilnya dengan arsitektur *monolithic* dimana tujuan dari pembuatan *microservice* untuk mendapatkan performa dan skalabilitas yang lebih baik (Abdullah et al., 2019).

|  |
| --- |
|  |
|  |

1. Contrast

Pada penelitian yang dilakukan oleh Sani et al., (2019), peneliti dengan jelas menyebutkan bahwa *microservice* dapat memberikan performa lebih cepat dari *monolithic*. Sedangkan penelitian yang dilakukan oleh Abdullah et al (2019), peneliti memberikan pernyataan bahwa *microservice* bias memberikan performa yang lebih lambat dari *monolithic* karena aspek-aspek tertentu.

|  |
| --- |
|  |
|  |

1. Criticize

Pada penelitian yang dilakukan oleh Roca et al., (2020), peneliti memberikan beberapa pendapat yang valid mengapa menggunakan arsitektur *microservice* daripada *monolithic*. Tapi peneliti tidak memberikan hasil *benchmark* untuk memberikan angka pasti seberapa besar keuntungan menggunakan arsitektur *microservice* dibandingkan dengan arsitektur *monolithic*. Peneliti hanya menyimpulkan bahwa *microservice* sangatlah fleksibel untuk pengembangan aplikasi.

|  |
| --- |
|  |

1. Synthesize

Menurut Penelitian yang dilakukan oleh Huang et al., (2019) peneliti memberikan beberapa pengujian yang menurut peneliti dapat mengoptimalkan kinerja dari arsitektur *microservice* dimana peneliti menggunakan pengujian dengan membedakan *CPU Utilization* dan *Memory Utilization*. Sedangkan di penelitian yang dilakukan oleh Wan et al., (2018) peneliti memberikan beberapa pengujian seperti *library* yang digunakan setiap *service*nya, *waking-up cost,* dan *communication cost.* Dari kedua penelitian ini dapat disimpulkan bahwa pengaruh penggunaan *resource* dari *hardware* harus diperhatikan juga, dan juga untuk *behavior* dari setiap *service* harus juga diperhatikan untuk meminimalisir *cost.*

|  |
| --- |
|  |
|  |

1. Summarize

Dari penelitian yang dilakukan oleh Sani et al., (2019), dapat disimpulkan bahwa *Microservice* memiliki skalabilitas tinggi, performa yang baik, dan juga dapat menekan biaya.

|  |
| --- |
|  |

# DAFTAR PUSTAKA

Abdullah, M., Iqbal, W., & Erradi, A. (2019). Unsupervised learning approach for web application auto-decomposition into microservices. *Journal of Systems and Software*, *151*, 243–257. https://doi.org/10.1016/j.jss.2019.02.031

Huang, L., Zhang, C., & Zeng, Z. (2019). Design of a public services platform for university management based on microservice architecture. *Microsystem Technologies*, *4*(Loukides 2012). https://doi.org/10.1007/s00542-019-04474-4

Roca, S., Sancho, J., García, J., & Alesanco, Á. (2020). Microservice chatbot architecture for chronic patient support. *Journal of Biomedical Informatics*, *102*(September 2019), 103305. https://doi.org/10.1016/j.jbi.2019.103305

Sani, N. A., Fillah, W. A., Tjahyanto, A., & Suryotrisongko, H. (2019). Development of microservice based application e-inkubator: Incubation and investment service provider for SMEs. *Procedia Computer Science*, *161*, 1064–1071. https://doi.org/10.1016/j.procs.2019.11.217

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