

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan seluruh proses yang telah dikerjakan oleh penulis dengan melakukan analisis dan pembangunan untuk mengubah sistem KRS *online* yang berarsitektur monolitik menjadi *microservices* serta pengujian yang dilakukan dalam penelitian maka dapat disimpulkan bahwa telah berhasil dibangun *backend* sistem KRS *online* yang dapat menangani proses pengisian KRS di Universitas Atma Jaya Yogyakarta. Dengan mengubah arsitektur monolitik menjadi *microservices* maka *scaling* dalam sistem KRS *online* lebih mudah dilakukan sehingga memudahkan pemantauan sumber daya yang digunakan pada saat pengisian KRS secara *online*. Penggunaan fitur *load balancing* yang dimiliki oleh *docker engine* dapat membuat sistem KRS *online* menjadi *reliable* dalam menangani beban berat pada saat pengisian KRS karena dapat menangani pengalokasian sumber daya pada saat *host* atau *container* mengalami *down*.

6.2 Saran

Dari seluruh proses yang telah dilakukan dalam pelaksanaan Tugas Akhir ini, terdapat beberapa saran yang dapat dilakukan untuk pengembangan lebih lanjut dari Pembangunan *Backend* Sistem Informasi KRS *Online* Berbasis *Microservices*, yaitu:

1. Penyediaan *server* lokal di Universitas Atma Jaya Yogyakarta dengan *node* yang lebih banyak, karena dalam penelitian ini *server* berupa *virtual machine* pada Azure dengan 4 buah *node*.
2. Melakukan update pada basis data Universitas Atma Jaya Yogyakarta yang saat ini adalah SQL Server 2008 R2 agar kompatibel dengan sistem operasi Linux.
3. Melakukan pengujian berupa simulasi KRS untuk mengetahui spesifikasi *server* yang dibutuhkan untuk menangani jumlah dan prilaku pengguna.

DAFTAR PUSTAKA

- .NET Core, 2018. *.NET Core*. [Online]
Available at: <https://docs.microsoft.com/en-us/dotnet/core/>
[Accessed 3 May 2018].
- Microservices, 2018. *Microservices*. [Online]
Available at: <https://www.docker.com/why-docker>
[Accessed 3 May 2018].
- Docker, 2018. *Docker*. [Online]
Available at: <https://microservices.io>
[Accessed 3 May 2018].
- Abrams, S., Kunze, J., & Loy, D. (2010). An Emergent Micro-Services Approach to Digital Curation Infrastructure. *The International Journal of Digital Curation*, 5(1), 172-186.
- Torre, C., Wagner, B., & Rousos, M. (2017). *.NET Microservices. Architecture For Containerized .NET Applications*. 1st Ed. Washington: Microsoft Corporation.
- Esposte, A., Costa, F. M., Kon, F., & Lago, N. (2017). InterSCity: A Scalable Microservice-based Open Source Platform for Smart. *6th International Conference on Smart Cities and Green ICT Systems* (pp. 35-46). Porto: International Conference on Smart Cities and Green ICT Systems.
- Firmansyah, A. (2015). Container dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web. *Jurnal Simantec*, IV(3), 167 -176.
- Munawar , G. (2018). Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK. *SinkrOn*, 3(1), 232-239.
- Newman, S. (2015). *Building Microservices DESIGNING FINE-GRAINED SYSTEMS*. Sebastopol: O'Reilly Media.
- Tome, P., & Silva, N. (2017). MICROSERVICES ARCHITECTURE FOR TOUR ACTIVITIES APPLICATION. *Asia Pacific Journal Of*

- Contemporary Education And Communication Technology*, III(1), 367-372.
- Smith, F. (2017). *Microservices From Design to Deployment*. San Francisco: NGIX.
- Suryotrisongko, H. (2017). Arsitektur Microservice Untuk Resiliensi Sistem Informasi. *Jurnal Sisfo*, VI(2), 235–250.
- Wolff, E. (2016). *Microservices: Flexible Software Architecture 1st Edition*. Boston: Addison-Wesley Professional.

