

PEMBUATAN SISTEM *LOGGING* TERSENTRALISIR UNTUK *EVENT DATA* PADA *MICROSERVICES* DI CV AVATAR SOLUTION

P. A. Pasuatmadi¹, I. B. M. Mahendra², C. R. A. Pramatha³

ABSTRAK

CV Avatar Solution atau biasa disebut dengan Avatar Solution adalah *software house* yang berbasis di pulau Bali. Avatar Solutions membuka kesempatan mahasiswa semester akhir untuk melakukan kegiatan pengabdian. Program pengabdian ini membantu penulis dan mahasiswa lainnya dalam menyelesaikan permasalahan – permasalahan nyata di bidang informatika. Salah satu permasalahan yang terjadi adalah susah nya melakukan proses *debugging* terhadap sistem yang terdesentralisir. Penulis memiliki solusi yaitu dengan pembuatan sistem *logging* yang tersentralisir. Sentralisasi ini bertujuan untuk memudahkan proses *debugging*. Penulis berhasil memecahkan permasalahan mitra, yang di mana program berhasil dibuat dan mampu menghasilkan performa 993 μ s (mikrodetik) dalam pengiriman 12427 data logging aktivitas dalam kurun waktu 5 detik dengan 10 *virtual user*. Sistem di atas membantu mitra dalam mempercepat dan mempermudah aktivitas *debugging* karena sistem bersifat tersentralisasi, mudah dijangkau, dan juga cepat yang menyebabkan pekerjaan mitra jauh dilancarkan dan dipermudah.

Kata kunci : Data Besar, Back End, Microservices, Basis Data, Data Aktivitas, Logging

ABSTRACT

CV Avatar Solution or commonly known as Avatar Solution is a software house company based in Bali Island. Avatar Solution opens an opportunity for college student to give a service activity. The activity helps the writer and other college students to solve real-life problems in the informatics field. One of those problems are the difficulty of doing debugging in a decentralized system. The writer has a solution which is a creation of a centralized logging system. This centralization of data has a purpose of helping programmers to debug or find bugs on the programs while the program is live in the internet. The writer successfully made and able to perform 993 μ s (microsecond) for 12427 logging data in just 5 seconds for 10 virtual users. It can be concluded that the system helped the debugging activities of the partner as the system is centralized, easily accessed, and fast that made the partner daily job activities much faster and much easier.

Keywords: Big Data, Back End, Microservices, Database, Event Data, Logging.

1. PENDAHULUAN

Dewasa ini, perusahaan – perusahaan di dunia cenderung menghasilkan maupun menerima sebuah *Big Data*. Dengan hal ini, sistem – sistem di zaman ini membutuhkan kapabilitas untuk memproses data besar tersebut seperti data *fintech*, kesehatan, *internet of things*, dan transportasi (Zhu *et al.*,

¹Program Studi Informatika, Fakultas MIPA, Universitas Udayana, Bukit Jimbaran, Badung, audipasuatmadi@gmail.com

²Program Studi Informatika, Fakultas MIPA, Universitas Udayana, Bukit Jimbaran, Badung, ibm.mahendra@unud.ac.id

³Program Studi Informatika, Fakultas MIPA, Universitas Udayana, Bukit Jimbaran, Badung, cokorda@unud.ac.id

Submitted: 7 November 2022

Revised: 25 November 2022

Accepted: 27 November 2022

2019). Data – data tersebut pula dapat berguna mengambil sebuah keputusan yang tentunya akan dapat membantu pihak perusahaan untuk membuat keputusan untuk perusahaan ataupun instansinya (Hammond *et al.*, 2020).

Data – data pada data besar sangatlah beragam. Salah satu dari data tersebut merupakan *Event Data* atau data aktivitas. Data aktivitas atau *Event Data* ini sangatlah berguna dikalangan programmer karena dapat digunakan untuk proses *debugging* (Chen *et al.*, 2019). Kemudian data aktivitas tersebut akan penting untuk mudah dilakukan metode pencariannya (Pramartha *et al.*, 2022). Data aktivitas tersebut harus disimpan pada basis data yang teroptimasi untuk dilakukan pencarian sehingga proses pencarian dapat berlangsung secara optimal (Pramartha *et al.*, 2017). Untuk itu, dibutuhkan pula sistem yang melakukan penyimpanan terhadap basis data yang benar dengan cara yang tepat. Seluruh hal di atas sangatlah diperlukan terutama untuk perusahaan teknologi yang cukup sering melakukan proses *development*.

Salah satu perusahaan tersebut yang berada di Bali adalah CV Avatar Solution. Perusahaan ini merupakan sebuah Software House yang dimana melayani klien dengan membuat perangkat lunak sesuai dengan kebutuhan yang diberikan. Tidak berhenti di situ, perusahaan ini pula memiliki berbagai jenis proyek lain yang melibatkan berbagai *stakeholder*. Dengan fakta ini, pembuatan sebuah sistem untuk menjadi pusat sentralisir pemrosesan data besar dengan tipe data aktivitas dengan nama Avatar Event Log Ingestor akan diperlukan untuk mempermudah proses *debugging* untuk programmers.

2. METODE PELAKSANAAN

2.1. Pelaksanaan Kegiatan

Program Praktek Kerja Lapangan pada Avatar Solutions disusun menggunakan pendekatan pembelajaran berbasis proyek atau *Project-Based Learning*. Fokus pembelajaran berpusat pada keterlibatan langsung di dalam pekerjaan nyata dan juga permasalahan – permasalahan nyata. Kegiatan ini juga mampu mengasah *Soft-Skill* karena memerlukan tingkat kolaborasi dan juga koordinasi yang tinggi dengan bidang – bidang lainnya untuk menyelesaikan suatu proyek.

2.2. Kebutuhan Fungsional Sistem

Kebutuhan fungsional merupakan sebuah kumpulan dari permintaan oleh pengguna akan fitur – fitur yang dibutuhkan untuk sistem. Dalam pengerjaan proyek Avatar Event Log Ingestor untuk Avatar Solution, terdapat fitur – fitur yang harus dipenuhi agar proyek dapat mampu berjalan sesuai dengan ekspektasi dan juga kebutuhannya. Fitur – fitur tersebut adalah sebagai berikut:

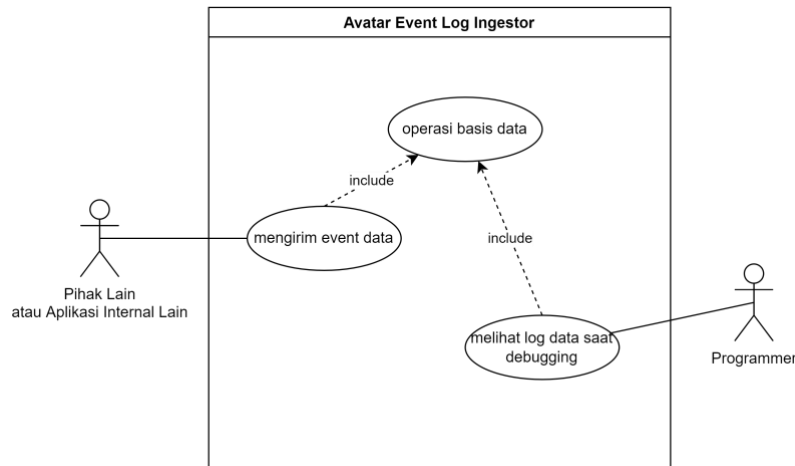
Tabel 2.1. Kebutuhan Fungsional

No	Kebutuhan Fungsional
1	Sistem dapat menerima dan menyimpan <i>Event Data</i> dari aplikasi lain
2	Sistem dapat menerima dan menyimpan <i>Error Event Data</i> dari aplikasi lain
3	<i>Event Data</i> dan juga <i>Error Event Data</i> akan disimpan pada MongoDB
4	Pembuatan <i>Client Library</i> untuk integrasi yang lebih mudah untuk <i>Backend</i> dengan <i>Node.js</i> dan juga <i>Go</i>

Kebutuhan fungsional di atas akan menjadi kebutuhan – kebutuhan yang menjadi tolak ukur keberhasilan dari sistem Avatar Event Log Ingestor. Sistem belum mendukung tampilan antarmuka karena sistem lebih memiliki fokus pada penyimpanan *Event Data* yang tersentralisir sehingga dapat memudahkan programmer lain melakukan proses *debugging*.

2.3. Perancangan Sistem

Perancangan sistem dilakukan dengan tujuan yaitu memudahkan *Software Engineer* maupun pihak – pihak lainnya dalam memahami sistem dengan cepat dan tepat. Perancangan sistem dilakukan dengan menggunakan Use Case Diagram yang akan memvisualisasikan bagaimana sistem dapat digunakan oleh berbagai pihak.



Gambar 2.1. Use Case Diagram Sistem

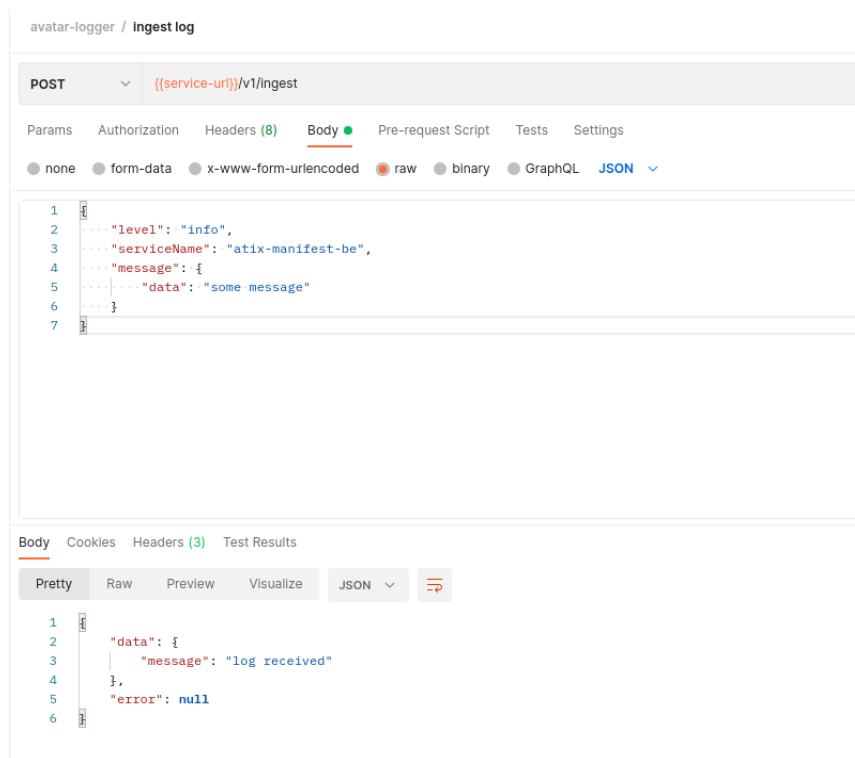
Terdapat dua aktor pada sistem ini, yaitu aplikasi lain dan juga programmer. Dapat dikatakan bahwa sistem ini hanya bekerja pada situasi yaitu sebuah perangkat lunak dengan konsep *Microservices*. Pihak lain atau aplikasi lain dapat mengirimkan *Event Data* ke dalam sistem yang dimana akan menyimpannya pada basis data, dan kemudian programmer dapat melakukan operasi basis data untuk mendapatkan data Event yang dibutuhkan untuk proses *Debugging*.

3. HASIL DAN PEMBAHASAN

Sebuah metodologi akan digunakan dalam proses pengerjaan dari sistem yang telah dirancang. Metodologi ini adalah metodologi Kanban. Metodologi ini membantu membuat pengerjaan semakin terorganisir dan juga semakin teratur. Pengerjaan kemudian dibagi lagi menjadi fase – fase untuk membuatnya lebih teratur lagi.

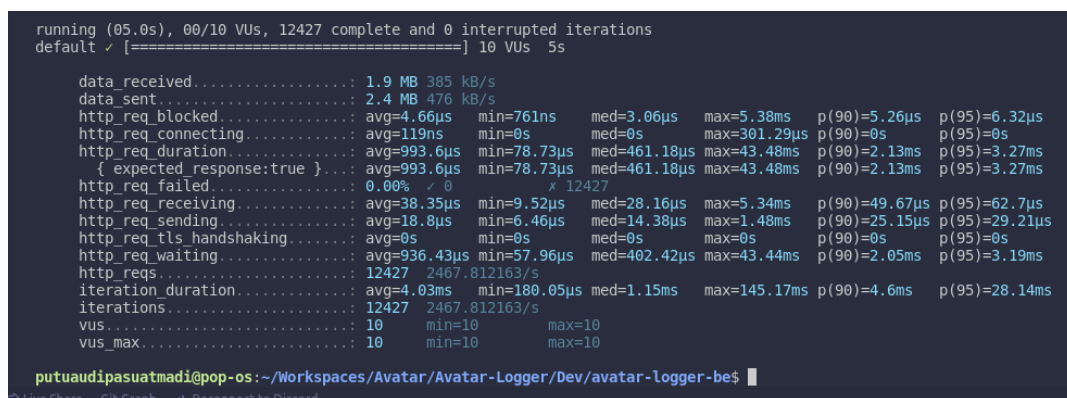
Fase pertama dalam pengerjaan yaitu fase *Scaffolding*. Fase ini merupakan fase pembuatan modul – modul yang diperlukan untuk pengerjaan lebih lanjut. Contohnya adalah inisialisasi *Client Database* dari basis data, dan juga beberapa tahapan lainnya. Fase kedua adalah fase pengerjaan dari fitur – fitur utama atau alur – alur bisnis. Fase ini berfokus pada apa hal pokok yang dilakukan dari sistem atau apa permasalahan utama yang dipecahkan oleh sistem. Fase terakhir merupakan fase *Refactoring*, dimana pada fase ini kode ditinjau ulang dan juga dibuat lebih optimal jika terdapat sebuah kode yang dapat dilakukan optimalisasi. Untuk hasil dari sistem dapat dilihat pada visualisasi di bawah ini.

Pembuatan Sistem Logging Tersentralisir Untuk Event Data Pada Microservices di CV Avatar Solution



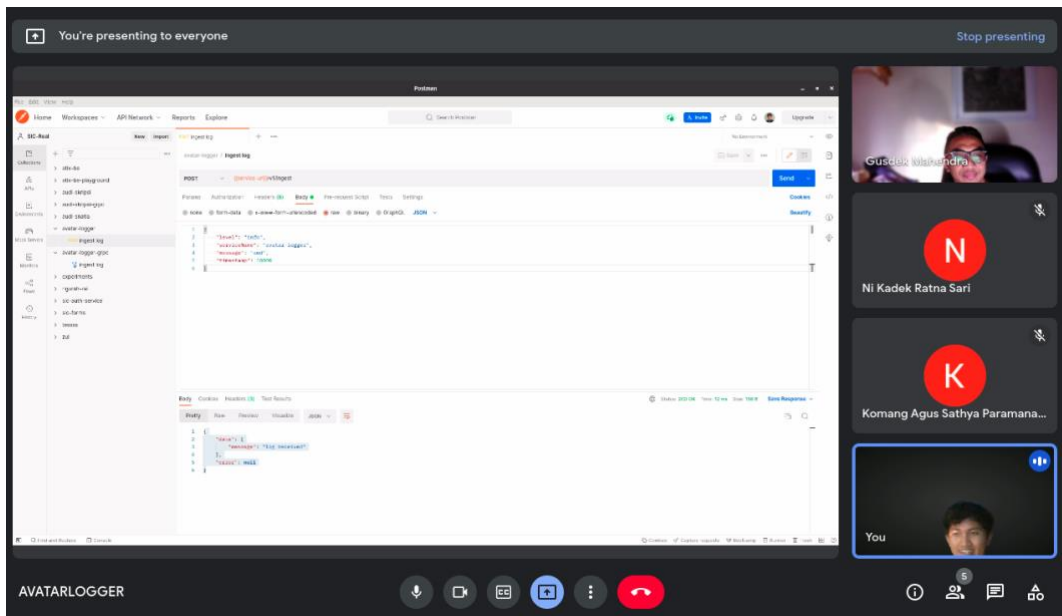
Gambar 3.1. Request dan juga Response dari Klien

Gambar di atas merupakan sebuah visualisasi dari bagaimana klien atau aplikasi lain akan mengirim dan juga menerima respon dari sistem. Berhubungan sistem ini merupakan sistem Backend, maka tidak terdapat tampilan antarmuka. Klien atau aplikasi lain dapat berubah aplikasi apapun dengan syarat yaitu dapat mengirim *Event Data* ke dalam sistem. Kemudian setelah data diterima, maka akan disimpan pada basis data di mana programmer dapat melakukan operasi – operasi yang dibutuhkan untuk proses *Debugging*.



Gambar 3.3. Hasil Pengujian Load Testing

Sistem yang dibuat kemudian diuji bagaimana dapat bertahan pada aktivitas yang tinggi dengan aktivitas *Load Testing*. Pengujian ini menggunakan *K6* di mana menghasilkan sebuah hasil seperti gambar di atas. Dapat dilihat bahwa sistem memiliki performa yang sangat tinggi dalam halnya menangani *request* dari pengirim data, yaitu 993µs (mikrodetik) dalam pengiriman 12427 data logging aktivitas dalam kurun waktu yang sangat singkat yaitu 5 detik dengan 10 *virtual user* berbeda.



Gambar 3.4. Sosialisasi dan Pelatihan Kepada Pihak Mitra

Kemudian untuk memastikan mitra dapat menggunakan sistem baru ini, dilaksanakan sosialisasi dan juga pelatihan kepada mitra untuk cara penggunaan dari sistem ini. Sosialisasi dan pelatihan dilaksanakan secara daring dan dengan melakukan demonstrasi dari penggunaan sistem baru untuk melihat aktivitas *event data* atau log yang ada. Hasil dan dampak dari pembuatan sistem ini dapat dilihat dari tabel di bawah.

Tabel 3.1. Tabel Hasil Kegiatan

Tahap Proses Kegiatan	Sebelum Proses Kegiatan	Setelah Proses Kegiatan
Melakukan <i>debugging</i> terhadap program.	Programmer harus memeriksa dan mengakses aplikasi – aplikasi yang berpotensi error.	Programmer cukup memeriksa satu sistem tersentralisasi aja.
Jangka lama <i>Event Data</i> .	<i>Event Data</i> hanya ada selama <i>Virtual Machine</i> dari aplikasi hidup.	<i>Event Data</i> ada selama – lamanya kecuali di hapus.
Kecepatan melihat <i>Event Data</i> .	Membutuhkan waktu lama, karena harus mengaksesnya melalui <i>terminal</i> dan harus mengakses <i>virtual machine</i> dan aplikasi yang dituju.	Membutuhkan waktu cepat, cukup mengakses sistem saja.
Mencari <i>Event Data</i> spesifik untuk <i>debugging</i> .	Harus melihat satu-per-satu data, tidak dapat melakukan pencarian.	Dapat melakukan pencarian dengan cepat.

Berdasarkan hasil evaluasi dari kegiatan masyarakat ini, dapat dinyatakan bahwa kegiatan ini telah berhasil dengan baik karena telah dapat mempermudah mitra dalam melakukan *debugging* jika terdapat sebuah kejadian seperti error yang terjadi pada aplikasi – aplikasi mitra. Mitra dipermudah dengan menyediakan sistem yang tersentralisasi, cepat, dan juga mudah untuk dijangkau dan diakses oleh mitra.

4. KESIMPULAN

Disimpulkan bahwa pembuatan sistem Avatar Event Log Ingestor sangat penting dan juga bermanfaat untuk penulis maupun pihak CV Avatar Solutions. Hal ini meningkatkan produktivitas

programmer di mana programmer tidak harus memeriksa setiap Backend untuk melakukan proses Debugging karena sudah terdapat sebuah sistem yang tersentralisir untuk menangani hal tersebut. Dapat dilihat pula bahwa sistem memiliki performa dan juga skalabilitas yang tinggi dari hasil Load Test yang dilakukan yaitu 993 μ s (mikrodetik) dalam pengiriman 12427 data logging aktivitas dalam kurun waktu 5 detik dengan 10 *virtual user*. Mitra dipermudah dengan ada kegiatan ini, dimana mitra dapat mengakses *event data* dengan cepat dan juga dengan tepat pada sistem yang tersentralisasi dan juga mudah diakses.

UCAPAN TERIMA KASIH

Penulsi mengucapkan terima kasih kepada CV Avatar Solution karena sudah memberikan kesempatan untuk melakukan proses pengabdian yang sangat berguna untuk penulis dan juga masyarakat. Penulis pula mengucapkan terima kasih untuk Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana karena telah memfasilitasi dan juga telah mengarahkan proses pelaksanaan pengabdian.

DAFTAR PUSTAKA

- Alaboudi, A. (2021). An Exploratory Study of Debugging Episodes. arXiv:2105.02162v1. <https://arxiv.org/pdf/2105.02162.pdf>
- Chen, A. R. (n.d.). An Empirical Study on Leveraging Logs for Debugging Production Failures. 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). <https://ieeexplore.ieee.org/document/8802657>
- Ford, N., & Richards, M. (2020). Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media
- Hamound, A. K. (2020). Improve HR Decision-Making Based On Data Mart and OLAP. Journal of Physics Conference Series. https://www.researchgate.net/publication/341673099_Improve_HR_Decision_on-Making_Based_On_Data_Mart_and_OLAP
- Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media.
- Kreps, J. (2014). I Heart Logs: Event Data, Stream Processing, and Data Integration. O'Reilly Media.
- Munshi, A. A., & Rady I. Mohamed, Y. A. (2018). Data Lake Lambda Architecture for Smart Grids Big Data Analytics. IEE Access, 2018. 10.1109/ACCESS.2018.2858256
- Penka, J. B. N., Saïd, M., & Debauche, O. (2022). An Optimized Kappa Architecture for IoT Data Management in Smart Farming. Journal of Ubiquitous Systems and Pervasive Networks, 16(1), 00-00. 10.5383/JUSPN.17.01.002
- Pramartha, C., Arka, I. W., Kuan, K. K., & Darmawan, I. (2022). Developing Semantic Ontology for Practical Digital Balinese Dictionary. Proceeding International Conference on Information Technology, Multimedia, Architecture, Design, and E-Business,
- Pramartha, C., Davis, J. G., & Kuan, K. K. Y. (2017, 4-6 December). Digital Preservation of Cultural Heritage: An Ontology-Based Approach. The 28th Australasian Conference on Information Systems, Hobart, Australia.
- Rana, M. S., Sohel, M. K., & Arman, M. S. (2018). Distributed Database Problems, Approaches and Solutions — A Study. International Journal of Rough Sets and Data Analysis, 8(5), 472-476. 10.18178/ijmlc.2018.8.5.731
- Trubiani, C., Bran, A., Hoorn, A. v., & Avritzer, A. (2017). Exploiting Load Testing and Profiling for Performance Antipattern Detection. Information and Software Technology, 95. 10.1016/j.infsof.2017.11.016
- Zhu, J. Y., Tang, B., & Li, V. O.K. (2019). A five-layer architecture for Big Data processing and analytics. Int. J. Big Data Intelligence, 6(1), 38-48. https://acm.sustech.edu.cn/btang/pub/IJBDI19_arch.pdf