

BAB V PENUTUP

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan evaluasi yang telah dilakukan terhadap arsitektur *backend* platform kursus daring TIME Language Centre, rancang bangun arsitektur berbasis *event-driven microservices* berhasil di-realisasikan ke dalam empat layanan mandiri, yaitu *User Service*, *Course Service*, dan *Notification Service* yang dikembangkan menggunakan Node.js dan TypeScript, serta *Payment Service* yang dikembangkan menggunakan bahasa Go. Setiap layanan mengelola basis data PostgreSQL masing-masing secara terisolasi sesuai pola *Database per Service*. Komunikasi asinkron antarlayanan berjalan melalui *message broker* RabbitMQ menggunakan protokol AMQP dengan mekanisme *dead letter queue* untuk menampung pesan yang gagal diproses. Lalu lintas permintaan dari klien dikelola oleh Nginx sebagai *API gateway*, dipadukan dengan Redis untuk *cache* katalog kursus dan manajemen sesi. Seluruh modul layanan terbukti berfungsi dengan benar melalui kelulusan 100% pada seluruh kasus uji unit, skenario pengujian integrasi, dan skenario pengujian fungsional *end-to-end* yang memverifikasi keterpenuhan seluruh kebutuhan fungsional sistem.

Kinerja arsitektur sistem yang dievaluasi menggunakan Grafana k6 di bawah beban bertahap hingga 200 *virtual users* (VU) selama 360 detik berhasil memenuhi seluruh ambang batas kebutuhan non-fungsional yang ditetapkan. Sistem memproses total 34.951 permintaan HTTP dengan *throughput* agregat 96,56 permintaan per detik dan tingkat kegagalan permintaan HTTP sebesar 0,00% (nol kegagalan), memenuhi target NFR-02 ($< 1\%$). Waktu respons global mencatatkan nilai persentil ke-95 (P95) sebesar 135,51 ms dan persentil ke-99 (P99) sebesar 223,85 ms, memenuhi target NFR-01 (P95 < 200 ms dan P99 < 400 ms). Pada Skenario B, sebanyak 14.170 transaksi *webhook* pembayaran diteruskan ke RabbitMQ tanpa kehilangan pesan (*zero message loss*) dengan latensi respons P95 sebesar 65 ms. Pengujian skalabilitas membuktikan bahwa penambahan replika kontainer *Course Service* menjadi dua instans menurunkan latensi P95 agregat sebesar 58,9% (dari

135,51 ms menjadi 55,70 ms) tanpa perubahan kode program, memenuhi target NFR-04.

Secara praktis, temuan penelitian ini menunjukkan bahwa pemisahan proses bisnis platform kursus daring ke dalam layanan mikro berbasis pesan asinkron memisahkan alur transaksi pembayaran dari proses aktivasi hak akses materi dan pengiriman notifikasi. Pemisahan ini mencegah terjadinya *blocking* di sisi pengguna saat volume transaksi meningkat. Bagi TIME Language Centre, prototipe arsitektur ini menyediakan fondasi teknis yang teruji untuk mendukung perluasan layanan pembelajaran daring ke luar Kota Padang.

B. Saran

Berdasarkan hasil evaluasi dan batasan sistem yang telah diidentifikasi, beberapa saran diajukan untuk pengembangan sistem serta penelitian lanjutan:

- a. *Horizontal Pod Autoscaling* (HPA) di Kubernetes: Penelitian lanjutan dapat mengevaluasi arsitektur ini pada kluster produksi berbasis Kubernetes dengan mekanisme *auto-scaling* dinamis berdasarkan metrik penggunaan CPU, memori, atau panjang antrean pesan RabbitMQ saat beban pengguna melebihi 200 pengguna virtual.
- b. Pengujian komparatif arsitektur (*head-to-head benchmarking*): Mengingat pengujian kinerja dalam penelitian ini bersifat mandiri, penelitian lanjutan disarankan merancang prototipe pembandingan dengan arsitektur monolitik atau *microservices* sinkron murni. Evaluasi komparatif pada kondisi perangkat keras dan skenario beban yang identik akan memberikan bukti empiris langsung mengenai signifikansi peningkatan *throughput* dan penurunan latensi yang dihasilkan oleh pendekatan *event-driven*.

DAFTAR PUSTAKA

- Anjasmara, V. M., & Sumitro, A. H. (2023). Pengembangan sistem informasi masjid darul arham menggunakan metode v-model dan uat (user acceptance testing). *Information System for Educators and Professionals*, 8(1), 47–58. <http://doi.org/10.51211/isbi.v8i1.2443>
- Bjørndal, N., Pires de Araújo, L. J., Bucchiarone, A., Dragoni, N., Mazzara, M., & Dustdar, S. (2021). Benchmarks and performance metrics for assessing the migration to microservice-based architectures. *Journal of Object Technology*, 20(3), 1–16. <https://doi.org/10.5381/jot.2021.20.2.a3>
- Blinowski, G., Ojdowska, A., & Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. *IEEE Access*, 10, 20357–20374. <https://doi.org/10.1109/ACCESS.2022.3152803>
- Broadcom Inc. (2022). RabbitMQ documentation. Dipetik February 12, 2026, dari <https://www.rabbitmq.com/docs>
- BunnyCDN. (2024). Bunny.net documentation. Dipetik May 10, 2026, dari <https://docs.bunny.net/>
- Cabane, H., & Farias, K. (2023). On the impact of event-driven architecture on performance: An exploratory study. *Future Generation Computer Systems*, 153. <https://doi.org/10.1016/j.future.2023.10.021>
- Chansanam, W., Tuamsuk, K., Poonpon, K., & Ngootip, T. (2021). Development of online learning platform for thai university students. *International Journal of Information and Education Technology*, 11(8), 348–354. <https://doi.org/10.18178/ijiet.2021.11.8.1534>
- Dahri, F., Elhanafi, A. M., Handoko, D., & Wulan, N. (2022). Implementation of microservices architecture in learning management system e-course using web service method. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 6(1), 76–82. <https://doi.org/10.33395/sinkron.v7i1.11229>
- de Almeida, M. G., & Canedo, E. D. (2022). Authentication and authorization in microservices: A systematic literature review. *Applied Sciences*, 12(6), 3023. <https://doi.org/10.3390/app12063023>
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems analysis and design: An object-oriented approach with uml* (5th). John Wiley & Sons.
- Dinova, C. A., & Utomo, I. C. (2023). Pengembangan arsitektur microservice pada learning management system e-learning menggunakan metode web service.

Jurnal Ilmu Komputer dan Informatika (JIKI), 3(2), 125–141. <https://doi.org/10.54082/jiki.102>

Docker Inc. (2023). Docker documentation: Overview. Dipetik March 15, 2026, dari <https://docs.docker.com/get-started/overview/>

draw.io. (2026). *Uml class diagrams*. Dipetik August 25, 2026, dari <https://www.drawio.com/docs/diagram-types/uml/class-diagrams>

Google. (2024). The Go programming language documentation. Dipetik March 15, 2026, dari <https://go.dev/doc/>

Google. (2026). Firebase authentication documentation. Dipetik April 18, 2026, dari <https://firebase.google.com/docs/auth>

GORM Contributors. (2026). Gorm documentation. Dipetik April 18, 2026, dari <https://gorm.io/docs/>

Grafana Labs. (2025a). Api load testing: A practitioner's guide with Grafana k6. Dipetik August 20, 2026, dari <https://grafana.com/docs/k6/latest/testing-guides/api-load-testing/>

Grafana Labs. (2025b). Distribute workloads across VUs. Dipetik August 28, 2026, dari <https://grafana.com/docs/k6/latest/examples/distribute-workloads/>

Grafana Labs. (2025c). Thresholds: Codify your SLOs as pass/fail criteria in k6. Dipetik August 20, 2026, dari <https://grafana.com/docs/k6/latest/using-k6/thresholds/>

Grafana Labs. (2026a). Grafana documentation. Dipetik February 12, 2026, dari <https://grafana.com/docs/grafana/latest/>

Grafana Labs. (2026b). Grafana Loki documentation. Dipetik March 15, 2026, dari <https://grafana.com/docs/loki/latest/>

Handoyo, E. D., Santoso, S., & Surjawan, D. J. (2024). Pengembangan aplikasi kursus digital menggunakan framework multiplatform dan clean architecture. *Jurnal Teknik Informatika dan Sistem Informasi*, 10(2), 259–271. <https://doi.org/10.28932/jutisi.v10i2.8402>

Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)* (RFC No. 7519). Internet Engineering Task Force. Dipetik June 13, 2026, dari <https://datatracker.ietf.org/doc/html/rfc7519>

Nielsen, J. (1993). *Usability engineering*. AP Professional.

- OpenJS Foundation. (2022). Node.js documentation. Dipetik February 12, 2026, dari <https://nodejs.org/en/docs>
- Permana, D. P., Ashaury, H., & Sabrina, P. N. (2023). Reengineering arsitektur monolithic ke microservices pada website management content MQTV. *Jurnal LOCUS: Penelitian & Pengabdian*, 2(9), 924–931. <https://doi.org/10.58344/locus.v2i9.1723>
- Pressman, R. S. (2010). *Software engineering: A practitioner's approach* (7th). McGraw-Hill.
- Resend Inc. (2024). Resend documentation. Dipetik May 10, 2026, dari <https://resend.com/docs>
- Sismadi, W., Martono, B. A., Susanto, Y., & Muzaeni, A. (2024). Implementasi arsitektur microservices pada web aplikasi penerimaan mahasiswa baru. *EDUTECH: Jurnal Inovasi Pendidikan Berbantuan Teknologi*, 4(2), 105–114. <https://doi.org/10.51878/edutech.v4i2.3062>
- Surantha, N., Utomo, O. K., Lionel, E. M., Gozali, I. D., & Isa, S. M. (2022). Intelligent sleep monitoring system based on microservices and event-driven architecture. *IEEE Access*, 10, 42069–42081. <https://doi.org/10.1109/ACCESS.2022.3167637>
- Syarif, U., & Pizaini. (2022). Penerapan event-driven microservices pada aplikasi layanan penerimaan peserta didik baru. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(3), 745–756. <https://doi.org/10.29100/jupi.v7i3.3067>
- Taha, M., & Ali, A. (2023). Redirection and protocol mechanisms in content delivery network-edge servers for adaptive video streaming. *Applied Sciences*, 13(9), 5386. <https://doi.org/10.3390/app13095386>
- The Linux Foundation. (2026). Prometheus documentation: Overview. Dipetik April 18, 2026, dari <https://prometheus.io/docs/introduction/overview/>
- The PostgreSQL Global Development Group. (2022). PostgreSQL documentation. Dipetik April 18, 2026, dari <https://www.postgresql.org/docs/>
- Tinambunan, D. H., Baehaqi, A., Avrianto, R. P., & Indrajit, R. E. (2023). Microgen implementation for building online learning management system with microservices and graphql generator approach. *Jurnal Teknik Informatika (JUTIF)*, 4(4), 967–976. <https://doi.org/10.52436/1.jutif.2023.4.4.1313>
- TypeORM Contributors. (2026). TypeORM documentation. Dipetik February 12, 2026, dari <https://typeorm.io/>

- Velepucha, V., & Flores, P. (2023). A survey on microservices architecture principles, patterns and migration challenges. *IEEE Access*, 11, 1–1. <https://doi.org/10.1109/ACCESS.2023.3305687>
- Venčkauskas, A., Kukta, D., Grigaliūnas, Š., & Bruzgienė, R. (2023). Enhancing microservices security with token-based access control method. *Sensors*, 23(6), 3363. <https://doi.org/10.3390/s23063363>
- Weerasinghe, S., & Perera, I. (2023). Optimized strategy for inter-service communication in microservices. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(2), 272–279. <https://doi.org/10.14569/IJACSA.2023.0140233>
- Wu, X., Zhang, H., & Yoon, M. (2025). Scalable online education platforms in higher education for enhancing students academic performance. *Scalable Computing: Practice and Experience*, 26(4), 1625–1638. <https://doi.org/10.12694/scpe.v26i4.4507>
- Xendit. (2024). Xendit api reference. Dipetik May 10, 2026, dari <https://developers.xendit.co/api-reference/>
- Zuki, S. Z. M., Mohamad, R., & Saadon, N. A. (2024). Containerized event-driven microservice architecture. *Baghdad Science Journal*, 21(2), 0584–0591. <https://doi.org/10.21123/bsj.2024.9729>

UIN IMAM BONJOL
PADANG