

BAB I

PENDAHULUAN

A. Latar Belakang

Platform kursus *online* menyelenggarakan pembelajaran daring tanpa batasan geografis. Wu dkk. (2025) membuktikan bahwa skalabilitas platform pembelajaran berkorelasi langsung terhadap capaian akademik peserta didik. Fakta ini menegaskan bahwa keandalan arsitektur perangkat lunak merupakan fondasi non-fungsional fundamental dalam penyelenggaraan platform pembelajaran daring.

TIME Language Centre merupakan lembaga pendidikan di Kota Padang yang berfokus pada persiapan tes IELTS, bimbingan *Work and Holiday Visa* (WHV), dan studi ke luar negeri, dengan rekam jejak melayani lebih dari 1.500 alumni selama 10 tahun beroperasi. Selama ini, seluruh program pembelajaran masih diselenggarakan secara tatap muka (*offline*) sehingga lembaga belum memiliki data historis konkurensi pengguna digital. Dalam rangka memperluas jangkauan layanan ke luar Kota Padang, lembaga berencana meluncurkan platform kursus daring secara publik. Rencana ini menuntut kesiapan arsitektur *back-end* yang andal untuk menangani peningkatan volume interaksi pengguna secara bersamaan.

Secara fungsional, platform kursus *online* melibatkan empat proses utama: pendaftaran pengguna, pengelolaan materi kursus, pemrosesan pembayaran, dan pengiriman notifikasi. Keempat proses ini tidak harus dieksekusi secara berurutan dalam satu alur permintaan tunggal. Sebagai contoh, ketika *student* menyelesaikan pembayaran kursus, pembukaan hak akses materi dan pengiriman *email* konfirmasi berjalan di latar belakang tanpa menahan respons server ke klien. Karakteristik ini bersifat asinkron dan menuntut arsitektur yang mampu memisahkan proses latar belakang dari alur respons utama. Arsitektur monolitik menghadapi kendala dalam mengakomodasi pemrosesan asinkron semacam ini akibat tingginya keterikatan antarkomponen (Blinowski dkk., 2022). Di sisi lain, pemisahan layanan mikro yang hanya mengandalkan komunikasi sinkron REST API juga belum tentu memberikan manfaat yang diharapkan; pengujian empiris Permana dkk. (2023) membuktikan

bahwa pemisahan layanan mikro dengan komunikasi REST API sinkron justru menghasilkan laju pemrosesan (*throughput*) yang lebih rendah dibandingkan arsitektur monolitik pada pengujian beban hingga 200 pengguna. Temuan tersebut menggarisbawahi bahwa pemisahan layanan harus disertai mekanisme komunikasi antarlayanan yang tepat. Dari perspektif ini, platform kursus *online* TIME Language Centre membutuhkan arsitektur yang modular sekaligus mendukung komunikasi asinkron antarlayanan sesuai karakteristik proses bisnisnya.

Arsitektur *event-driven microservices* hadir sebagai pendekatan yang mengatasi kedua keterbatasan tersebut. Dengan memecah sistem menjadi layanan independen yang berkomunikasi asinkron melalui *message broker* (Velepucha & Flores, 2023), pendekatan ini memberikan fleksibilitas penuh: setiap unit layanan dapat dirancang, disebar, serta ditingkatkan kapasitasnya secara mandiri menyesuaikan beban masing-masing. Mekanisme komunikasi asinkron berbasis *Publish/Subscribe* melalui *message broker* terbukti menurunkan latensi antarlayanan serta meningkatkan *throughput* jika dibandingkan dengan komunikasi sinkron berbasis HTTP (Weerasinghe & Perera, 2023). Meskipun skala pengguna awal TIME Language Centre saat ini masih dapat ditangani oleh server monolitik tunggal, keputusan mengadopsi arsitektur ini sejak tahap awal sebagai sistem baru (*greenfield project*) didasari pertimbangan utang teknis arsitektur (*architectural technical debt*). Proses refaktorisasi dari sistem monolitik ke layanan mikro di kemudian hari membawa kompleksitas integrasi yang tinggi serta biaya migrasi basis data yang besar (Blinowski dkk., 2022), sehingga membangun dengan fondasi arsitektur yang tepat sejak tahap awal menghindari biaya migrasi tersebut.

Berdasarkan kajian literatur yang telah dilakukan, penelitian terdahulu mengenai arsitektur *microservices* pada domain pendidikan (Dahri dkk., 2022; Dinova & Utomo, 2023; Sismadi dkk., 2024) umumnya masih berfokus pada pemisahan layanan dan komunikasi sinkron berbasis REST API tanpa *message broker*. Kajian yang telah menerapkan pola *event-driven* (Syarif & Pizaini, 2022) belum mengempatkan alur pendaftaran, pengelolaan kursus, pembayaran, dan notifikasi sebagai proses transaksi asinkron yang diuji kinerjanya dalam satu rancangan *back-*

end terpadu. Dari pemetaan literatur tersebut, belum ditemukan penelitian yang merancang, mengimplementasikan, dan mengukur kinerja arsitektur *event-driven microservices* secara terpadu pada domain platform kursus *online*. Penelitian ini mengisi celah penelitian tersebut.

Menjawab tantangan tersebut, penelitian ini berfokus merealisasikan rancang bangun arsitektur *backend* berbasis *event-driven microservices* untuk platform pembelajaran daring TIME Language Centre. Arsitektur yang dikembangkan terdiri atas empat layanan independen dengan pola *Database per Service*, yang menerapkan komunikasi asinkron berbasis pesan (*asynchronous messaging*) melalui protokol AMQP pada RabbitMQ. Kinerja arsitektur selanjutnya diverifikasi melalui pengujian beban (*load testing*) menggunakan Grafana k6 berdasarkan metrik *response time*, *throughput*, dan *error rate* guna membuktikan kelayakan arsitektur dalam menangani variasi beban pengguna konkuren.

B. Rumusan Masalah

Berlandaskan latar belakang yang telah dipaparkan, persoalan pokok yang diteliti dirumuskan ke dalam pertanyaan penelitian berikut:

1. Bagaimana rancangan arsitektur *backend* berbasis *event-driven microservices* yang mampu mendukung komunikasi asinkron berbasis pesan (*asynchronous messaging*) antarlayanan untuk platform kursus *online* TIME Language Centre?
2. Bagaimana kinerja arsitektur *backend event-driven microservices* yang dikembangkan dalam menangani beban pengguna berdasarkan metrik *response time*, *throughput*, dan *error rate*?

C. Batasan Masalah

Guna menjaga fokus kajian agar tetap spesifik dan terarah, batasan ruang lingkup penelitian ini ditetapkan sebagai berikut:

1. Penelitian ini difokuskan pada perancangan dan pengujian arsitektur *backend*, sedangkan antarmuka pengguna hanya dikembangkan sebagai pendukung pengujian sistem.
2. Arsitektur *backend* terdiri atas empat layanan mandiri, yaitu *User Service*,

Course Service, dan *Notification Service* yang dikembangkan menggunakan TypeScript di atas *runtime* Node.js, serta *Payment Service* yang dikembangkan menggunakan bahasa Go.

3. Penelitian ini tidak menerapkan mekanisme penskalaan otomatis dinamis (*auto-scaling* berbasis metrik pada klaster produksi seperti Kubernetes). Evaluasi kinerja sistem berfokus pada konfigurasi dasar satu instans (*single replica*), sedangkan pengujian multi-replika dilakukan secara terarah melalui Docker Compose sebagai pembuktian terhadap kemampuan skalabilitas arsitektur.
4. Pengujian dan evaluasi arsitektur dalam penelitian ini bersifat mandiri (*standalone*) dengan mengukur kinerja sistem terhadap ambang batas kinerja (*performance threshold*) yang ditetapkan, sehingga tidak mencakup pengujian komparatif (*head-to-head benchmarking*) secara empiris terhadap arsitektur alternatif (seperti arsitektur monolitik).

D. Tujuan Penelitian

Mengacu pada rumusan masalah yang diajukan, tujuan yang hendak dicapai dalam penelitian ini dijabarkan sebagai berikut:

1. Merancang arsitektur *backend* berbasis *event-driven microservices* dan mengimplementasikan prototipe sistemnya ke dalam empat layanan mandiri (*User Service*, *Course Service*, *Payment Service*, dan *Notification Service*) dengan menerapkan pola *Database per Service* dan komunikasi asinkron berbasis pesan (*asynchronous messaging*) melalui RabbitMQ untuk platform kursus *online* TIME Language Centre.
2. Menganalisis dan mengevaluasi kinerja sistem *backend* yang dikembangkan melalui pengujian beban (*load testing*) menggunakan Grafana k6 bertahap hingga 200 pengguna virtual berdasarkan metrik *response time*, *throughput*, dan *error rate*, guna memvalidasi ketercapaian kriteria kebutuhan non-fungsional yang ditetapkan.

E. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat teoretis dan praktis:

1. Manfaat Teoretis

Pada aspek keilmuan rekayasa perangkat lunak, penelitian ini memberi kontribusi berupa:

- a. Menyediakan data dan bukti empiris mengenai penerapan arsitektur *event-driven microservices* dengan pola *Database per Service* dan komunikasi asinkron berbasis pesan (*asynchronous messaging*) melalui RabbitMQ pada platform kursus *online*, yang kajiannya masih terbatas dalam literatur akademik berbahasa Indonesia.
- b. Menyediakan bukti empiris mengenai karakteristik kinerja dan batas kapasitas arsitektur layanan mikro asinkron yang diintegrasikan dengan orkestrasi kontainer Docker dan instrumen pemantauan sistem (*system monitoring*) dalam menangani variasi beban konkurensi pengguna.
- c. Menjadi bahan rujukan konseptual serta panduan metodologis bagi riset lanjutan yang mendalami perancangan sistem terdistribusi berbasis *event-driven* pada ranah sistem informasi edukasi.

2. Manfaat Praktis

Secara terapan, temuan dan luaran dari penelitian ini memberikan manfaat langsung bagi pihak terkait:

- a. Bagi TIME Language Centre, penelitian ini menghasilkan rancangan arsitektur dan prototipe sistem *backend* platform kursus *online* yang teruji kinerjanya, sehingga menjadi fondasi teknis bagi lembaga untuk mengembangkan platform digital yang siap diskalakan tanpa menimbulkan utang teknis arsitektur (*architectural technical debt*).
- b. Bagi pengembang perangkat lunak, penelitian ini menyediakan dokumentasi rancangan teknis yang dapat dijadikan rujukan mengenai dekomposisi layanan mikro, mekanisme pertukaran pesan asinkron antarlayanan, serta implementasi orkestrasi kontainer Docker dan instrumen pemantauan sistem

(*system monitoring*) pada arsitektur layanan mikro.

- c. Bagi institusi pendidikan, penelitian ini menjadi acuan strategis dalam memilih arsitektur perangkat lunak yang berkinerja andal, mudah diskalakan (*scalable*), dan toleran terhadap kegagalan untuk penyelenggaraan platform pembelajaran daring.

