

## OPTIMALISASI E-LEARNING MELALUI IMPLEMENTASI MICRO-SERVICE UNTUK PENINGKATAN SKALABILITAS DAN EFISIENSI PEMBELAJARAN *ONLINE*

Muhammad Iqbal<sup>1</sup>, Wiwin Handoko<sup>2</sup>, Abdul Karim Syahputra<sup>3</sup>

<sup>1,2,3</sup>Sistem Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer Royal

*email*: codegoc@gmail.com

**Abstract:** The increased use of technology in education has driven the development of e-learning platforms that can provide effective and affordable learning experiences. However, the scalability and viability of e-learning platforms is often a challenge in the face of an increasing number of users and workloads. SMK Swasta As-Syifa Kisaran uses a monolithic-based E-learning platform which causes the more platform activities are used the heavier. This dedication is a continuation of previous PKM activities regarding the design of E-learning platforms at the school. The previous monolithic architecture implemented needs to be updated with a microservice architecture aimed at optimizing scalability and maintaining the continuity of the E-Learning platform. The problem of high traffic usage and focusing on one server is increasingly burdening the server. Based on these problems, the service is broken down into several webservices using a microservice architecture consisting of independent microservices, such as user management, course, content, and scoring systems. Vertical increases in CPU capacity are required so that microservices can cope with increasing server workloads. The results of the microservice implementation applied to the SMK As-Syifa e-learning platform are able to provide optimal scalability for the continuity of the platform in the face of increasing user needs.

**Kata kunci:** e-learning; microservice; scalability; performance; monolitik

**Abstract:** Peningkatan penggunaan teknologi dalam pendidikan telah mendorong perkembangan platform e-learning yang dapat memberikan pengalaman pembelajaran yang efektif dan terjangkau. Namun, skalabilitas dan kelangsungan platform e-learning sering menjadi tantangan dalam menghadapi peningkatan jumlah pengguna dan beban kerja. SMK Swasta As-Syifa Kisaran menggunakan platform E-learning berbasis monolitik yang menyebabkan semakin banyak aktivitas platform digunakan semakin berat. Beberapa tahapan dilakukan dimulai dengan identifikasi masalah, perencanaan dan desain program, pelaksanaan, evaluasi dan analisis, serta penyusunan dan publikasi laporan. Arsitektur monolitik sebelumnya yang diterapkan perlu diperbaharui dengan arsitektur microservice bertujuan untuk mengoptimalkan skalabilitas dan menjaga kelangsungan platform E-Learning. Permasalahan pada penggunaan traffic yang tinggi dan berfokus pada satu server semakin lama semakin membebani server. Berdasarkan masalah tersebut maka layanan dipecah menjadi beberapa layanan webservice menggunakan arsitektur microservice yang terdiri dari layanan-microservice independen, seperti manajemen pengguna, kursus, konten, dan sistem penilaian. Peningkatan kapasitas CPU secara vertikal diperlukan sehingga microservice dapat mengatasi peningkatan beban kerja server. Hasil implementasi microservice yang diterapkan pada platform e-learning SMK As-Syifa mampu memberikan skalabilitas yang optimal untuk kelangsungan platform dalam menghadapi peningkatan kebutuhan pengguna

**Keywords:** e-learning; *microservice*; skalabilitas; kinerja; monolitik

## PENDAHULUAN

Pengabdian ini merupakan lanjutan dari kegiatan pengabdian sebelumnya di SMK As-syifa Kisaran tentang penerapan Platform Pembelajaran *Online* (E-Learning). Platform yang digunakan sebelumnya menerapkan arsitektur monolitik sebagai tahapan awal rancangan sistem. Akan tetapi, masalah beban kerja server meningkat disebabkan meningkatnya trafik pengguna aplikasi. Melalui aplikasi e-learning, siswa dapat mengakses materi pembelajaran, menyelesaikan tugas, berpartisipasi dalam diskusi *Online*, dan berinteraksi dengan guru dan siswa lainnya (Uyainah, 2023).

Pada penelitian (Riyanto et al., 2022) yang berjudul “Implementasi Service Choreography Pattern Arsitektur *Microservice* Classroom Akademik Menggunakan Docker”, menyatakan bahwa Penggunaan arsitektur monolitik masih umum diterapkan. Tapi karena e-learning menjadi lebih banyak digunakan, masalah baru muncul yang harus diselesaikan untuk menyediakan lingkungan belajar sebaik mungkin. Masalah terjadi ketika peningkatan jumlah pengguna atau beban kerja yang tinggi. Aplikasi monolitik sering kali kesulitan menangani beban tersebut secara efisien.

Arsitektur sistem informasi yang kurang tahan terhadap elastisitas adalah arsitektur monolitik dimana sistem backend dan frontend digabungkan dalam satu tempat dari segi fungsionalitas dan layanan, selain itu arsitektur monolitik juga menjalankan semua logika dalam satu server aplikasi (Atmojo et al., 2022). Hal tersebut dapat mengakibatkan penurunan kualitas layanan, waktu respon yang lambat, dan bahkan kegagalan sistem. Aplikasi Ujian *Online* yang dimiliki oleh SMK As-syifa masih menggunakan arsitektur monolitik (Iqbal

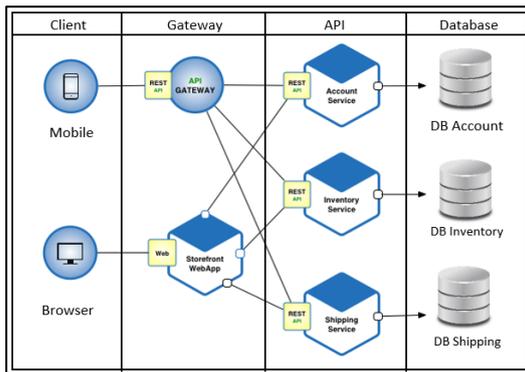
et al., 2022). Arsitektur Monolitik merupakan aplikasi yang dikembangkan dan di-deploy sebagai satu entitas (Rezaldy et al., 2017).

Pengembangan LMS (*Learning Management System*), masih umum digunakan arsitektur monolitik yang menjalankan semua logika aplikasi dalam satu server. Arsitektur monolitik memiliki kelemahan dalam meng-adaptasi perubahan kebutuhan sistem, kompleksitas kode, dan pemeliharaan (Dinova & Utomo, 2023). Berbeda dengan Arsitektur *Microservice*, Arsitektur *microservices* secara sederhana merupakan sebuah arsitektur dengan memecah aplikasi menjadi bagian-bagian dari fungsi-fungsi kecil yang spesifik (Dahri et al., 2022). Karena layanan-layanan tersebut berdiri sendiri membuat kinerja aplikasi akan lebih cepat dan mudah untuk dikembangkan menjadi lebih kompleks (Mulyawan, 2022)

Skalabilitas dan keberlanjutan platform e-learning menjadi dua isu utama. Ada persyaratan untuk infrastruktur yang dapat menangani beban kerja yang lebih berat dengan efisiensi dan keandalan yang tinggi karena jumlah pengguna dan konten pembelajaran terus meningkat. Arsitektur *microservice* merupakan arsitektur alternatif yang lebih scalable dan lebih fleksibel. Dalam arsitektur *microservice*, sistem informasi dirancang untuk didistribusikan dan menyediakan layanan dengan cara yang lebih fokus dan spesifik. Masalah besar akan dipecah menjadi beberapa solusi kecil yang disusun dalam satu layanan, dimana masing-masing layanan memiliki tanggung jawabnya masing-masing (Dahri et al., 2022).

Arsitektur *microservices* ditujukan agar setiap *services* dapat fokus melakukan satu tugas atau fungsionalitas

tertentu dengan baik. Tujuan dari pelaksanaan pengabdian ini memberikan edukasi dalam skalabilitas penggunaan *e-learning* yang semakin kompleks sehingga dibutuhkan arsitektur lain untuk menangani beban kerja yang lebih berat menggunakan *microservice*.



**Gambar 1.** Mekanisme *Microservice*

Penggunaan teknologi *micro-service* dalam konteks ini menjadi penting. Aplikasi dibagi menjadi komponen otonom kecil yang menggunakan metodologi pengembangan perangkat lunak yang dikenal sebagai *microservice*. Platform *e-learning* dapat ditingkatkan menjadi lebih terukur dan berkelanjutan dengan menggunakan desain *micro-service*. Berdasarkan penelitian terdahulu Implementasi Teknologi Mikroservice pada Pengembangan Mobile Learning memberikan kesimpulan bahwa Aplikasi mobile learning yang dibangun memiliki enam mikroservice yang dapat dikembangkan sesuai kebutuhan dengan menambahkan service tertentu ke sistem yang telah ada (Sendiang et al., 2018).

Pengabdian ini menganalisa urgensi dalam mengadopsi teknologi *microservice* untuk mengoptimalkan platform *e-learning*. Pengabdian ini juga membahas tentang bagaimana strategi ini dapat meningkatkan skalabilitas, efektivitas, dan efisiensi pembelajaran digital.

## METODE

Pengabdian Masyarakat ini dilaksanakan di SMK Swasta Assyifa Kisaran. Terdiri dari Tim Dosen, Kegiatan ini diselenggarakan dalam bentuk presentasi sebagai sosialisasi dan pengerjaan sistem beberapa hari setelah kegiatan dilakukan. Prosedur yang digunakan untuk mengimplementasikan web service dalam proyek pengembangan perangkat lunak disebut WSIM (atau *web service implementation method*). Dengan menggunakan teknik ini, *web service* dapat dirancang, dibangun, diuji, dan dipelihara dengan baik. Sejumlah prosedur disertakan dalam Metode Implementasi *Web service* untuk menjamin implementasi layanan web yang berhasil. Tindakan umum dalam proses ini meliputi (Redha et al., 2022):

**Analisis Kebutuhan**, Memahami kebutuhan pengguna dan persyaratan organisasi yang harus dipatuhi oleh layanan web agar dapat dikembangkan.

**Perancangan Web Service**, Merancang struktur dan fungsi layanan web, termasuk pemilihan teknologi, definisi antarmuka, dan pengaturan keamanan.

**Implementasi Web Service**, Menerapkan kode dan logika bisnis yang diperlukan untuk membangun dan menjalankan *web service*.

**Pengujian**, Melakukan serangkaian pengujian untuk memastikan kualitas, keandalan, dan kinerja *web service*. Ini meliputi pengujian fungsional, pengujian integrasi, dan pengujian beban.

**Konfigurasi Infrastruktur**, Menyiapkan lingkungan infrastruktur yang diperlukan, seperti server, *database*, dan alat-alat

pendukung lainnya untuk menjalankan *web service* dengan baik.

**Peluncuran dan Pemeliharaan Layanan Web**, Meluncurkan layanan web ke lingkungan produksi dan melaksanakan pemeliharaan rutin untuk memastikan kelancaran operasi dan ketersediaan layanan.

## PEMBAHASAN

Hasil Analisa yang dilakukan memperoleh beberapa instance untuk diterapkan pada *microservice*.

**Manajemen pengguna**, *Instance* yang bertanggung jawab untuk mengelola informasi pengguna, seperti pendaftaran, otentikasi, profil, dan pengelolaan hak akses.

**Manajemen Kursus**, *Instance* yang menangani fungsionalitas terkait kursus, seperti pembuatan, pengeditan, penghapusan, dan pemetaan peserta pada kursus tertentu.

**Manajemen Konten**, *Instance* yang bertugas mengelola konten pembelajaran, termasuk unggahan, pengeditan, pembaruan, dan penyebaran materi pembelajaran kepada pengguna.

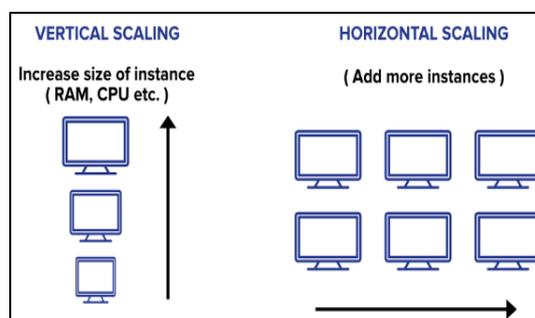
**Sistem Penilaian**, *Instance* yang mengelola proses penilaian dan evaluasi, termasuk penugasan tugas, ujian, dan penghitungan nilai.

**Interaksi**, *Instance* yang memungkinkan interaksi dan kolaborasi antara pengguna, seperti forum diskusi, obrolan langsung, atau kemampuan berbagi informasi dan pengetahuan.

Skalabilitas menjadi krusial dalam konteks pengembangan platform e-learning untuk memastikan bahwa sistem dapat berkembang dan beradaptasi dengan perubahan kebutuhan. Dalam pengembangan aplikasi, skalabilitas

umumnya terdiri dari dua jenis: **Skalabilitas Horizontal**, Meningkatkan kapasitas dengan menambahkan lebih banyak *instance* atau komponen yang sebanding. Hal ini mengacu pada peningkatan jumlah *instance microservice* untuk mengakomodasi beban kerja yang terus bertambah. Sedangkan pada **Skalabilitas vertikal**, meningkatkan fitur dan sumber daya dari satu instans atau komponen, kapasitas dapat ditingkatkan. Mengacu pada peningkatan sumber daya yang tersedia untuk setiap *instance* dalam kerangka arsitektur *microservice*.

Platform E-Learning SMK Swasta As-syifa memiliki CPU dengan 1,5 core. Perhitungan penambahan kapasitas secara vertikal tentunya membutuhkan pemikiran yang lebih cermat. Inti CPU adalah unit pemrosesan independen yang mampu mengeksekusi instruksi. Dalam kasus ini, memiliki 1,5 core berarti memiliki 1 core penuh dan setengah inti CPU.



Gambar 2. Scaling Concept

Dalam konteks peningkatan kapasitas secara vertikal, dapat melakukan perhitungan berdasarkan proporsi yang tersedia. Menjadikan setengah inti sebagai 50% dari satu core penuh dengan perhitungan:

Kapasitas CPU saat ini = 1,5 core  
Peningkatan kapasitas CPU yang diinginkan: = 0,5 core  
Kapasitas CPU setelah peningkatan:  
= (1,5 core) + CPU (0,5 core)

= 2 core

Core pada CPU adalah unit pemrosesan fisik yang tidak dapat dibagi menjadi pecahan, seperti 1,5 core. Oleh karena itu, untuk dapat melakukan optimalisasi dengan upgrade core CPU lebih besar agar platform dapat berkelanjutan melalui layanan provider (*hosting*).

## SIMPULAN

Pemecahan layanan pada aplikasi e-learning menjadi beberapa *micro-service* independen, sehingga dapat mengatasi beberapa masalah yang dihadapi oleh SMKs As-syifa saat menggunakan arsitektur monolitik. Masalah penurunan kualitas layanan, waktu respon yang lambat, dan bahkan kegagalan sistem dapat diminimalisir dengan layanan yang terpisah sehingga mengurangi beban kerja server yang terpusat. Selain itu, Peningkatan kapasitas CPU pada setiap instance *microservice* pada Platform E-Learning As-syifa dilakukan secara skalabilitas vertikal. Pada perhitungan yang dijelaskan, peningkatan kapasitas CPU dilakukan dengan menambahkan 50% inti CPU per instance.

Terlihat bahwa dengan menambahkan kapasitas CPU, jumlah inti CPU pada setiap instance *microservice* meningkat dari 1.5 Core menjadi 2 Core. Hal ini mengarah pada peningkatan kapasitas total CPU pada platform *e-learning* secara keseluruhan. Dengan memiliki kapasitas CPU yang lebih tinggi, setiap instance *microservice* dapat menangani lebih banyak permintaan dari pengguna dengan responsivitas yang baik.

## DAFTAR PUSTAKA

Atmojo, S., Utami, R., Dewi, S., & Widhiyanta, N. (2022).

- Implementasi Sistem-informasi Desa Berbasis Arsitektur Microservices. *Smatika Jurnal*, 12(01), 55–66. <https://doi.org/10.32664/smatika.v12i01.658>
- Dahri, F., Hanafi, A. M. El, Handoko, D., & Wulan, N. (2022). Implementation of Microservices Architecture in Learning Management System E-Course Using Web Service Method. *Sinkron*, 7(1), 76–82. <https://doi.org/10.33395/sinkron.v7i1.11229>
- Dinova, C. A., & Utomo, I. C. (2023). Pengembangan Arsitektur Microservice pada Learning Management System E-learning Menggunakan Metode Web Service. 1(1), 65–82.
- Iqbal, M., Syahputra, A. K., & Handoko, W. (2022). PENERAPAN MONOLITIC ARSITEKTUR PADA APLIKASI UJIAN PENDAHULUAN Dalam dunia pendidikan ujian dimaksudkan untuk mengukur taraf pencapaian oleh siswa, sehingga siswa dapat mengetahui tingkat kemampuannya dalam memahami bidang studi. 2(2), 213–218.
- Mulyawan, A. A. (2022). Sistem Pengelolaan Target Perusahaan dengan Microservices Architecture untuk Membantu Peningkatan Kinerja Perusahaan. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(1), 12–22. <https://doi.org/10.35957/jatisi.v9i1.1423>
- Redha, T., Fath, A., Husni, I., & Amin, A. (2022). Implementasi Arsitektur Microservices

- menggunakan RESTful API untuk Website Online Course Esploor.* 07, 2657–1501.
- Rezaldy, M., Asror, I., & Sardi, I. L. (2017). Desain dan Analisis Arsitektur Microservices Pada Sistem Informasi Akademik Perguruan Tinggi Dengan Pendekatan Architecture Tradeoff Analysis Method (ATAM) (Studi Kasus: iGracias Universitas Telkom). *E-Proceeding of Engineering*, 4(2).
- Riyanto, D. J., Pizaini, P., H., N. S., & Affandes, M. (2022). Implementasi Service Choreography Pattern Arsitektur Microservice Classroom Akademik Menggunakan Docker. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 7(3), 768–779. <https://doi.org/10.29100/jipi.v7i3.3126>
- Sendiang, M., Kasenda, S., & Purnama, J. (2018). Implementasi Teknologi Mikroservice pada Pengembangan Mobile Learning. *Journal of Applied Informatics and Computing*, 2(2), 63–66. <https://doi.org/10.30871/jaic.v2i2.1046>
- Uyainah, K. (2023). Pengembangan Aplikasi E-Learning Berbasis Web dengan Fitur Interaktif untuk Meningkatkan Efektivitas Pembelajaran di Perguruan Tinggi. 3(3), 1–19.