

## INTISARI

E-Learning System (ELLS) di Universitas Amikom Purwokerto memerlukan adaptasi terhadap karakteristik mahasiswa yang beragam, salah satunya melalui deteksi gaya belajar model Felder-Silverman (FSLSM). Namun, integrasi langsung antara sistem utama dengan layanan Machine Learning sering kali terkendala masalah latensi yang tinggi, yang dapat menurunkan kenyamanan pengguna. Penelitian ini bertujuan untuk merancang dan mengimplementasikan arsitektur *backend* terintegrasi yang efisien dengan memisahkan logika bisnis dan komputasi cerdas menggunakan pendekatan *microservices*.

Sistem dikembangkan dengan menggabungkan *framework* Express.js sebagai *backend* utama dan FastAPI sebagai layanan prediksi, yang berkomunikasi melalui *REST API*. Untuk mengatasi masalah performa, diterapkan teknik *caching* dengan strategi *Stale-While-Revalidate*. Metode pengembangan sistem meliputi perancangan spesifikasi *API*, modifikasi basis data, dan perancangan alur interaksi asinkron. Pengujian dilakukan secara fungsional menggunakan *Black Box Testing* dan pengujian performa menggunakan Artillery.io dengan skenario pembebanan tinggi.

Hasil penelitian menunjukkan bahwa seluruh *endpoint* integrasi berfungsi sesuai rancangan. Penerapan teknik *caching* terbukti meningkatkan performa sistem secara signifikan. Pada pengujian beban 2.265 permintaan, latensi rata-rata berhasil diturunkan dari 2.200 ms (tanpa *cache*) menjadi 7 ms (dengan *cache*), serta berhasil menghilangkan tingkat kegagalan permintaan (*error rate*) dari 37,17% menjadi 0%. Arsitektur ini terbukti efektif dalam menyajikan prediksi gaya belajar secara konsisten dan stabil.

Kata kunci: Gaya Belajar, Microservices, FastAPI, Express.js, Caching.

## **ABSTRACT**

*The E-Learning System (ELLS) at Universitas Amikom Purwokerto requires adaptation to diverse student characteristics, specifically through the Felder-Silverman Learning Style Model (FSLSM) detection. However, direct integration between the main system and Machine Learning services is often hindered by high latency issues, which can degrade user experience. This study aims to design and implement an efficient integrated backend architecture by separating business logic and intelligent computing using a microservices approach.*

*The system was developed by combining the Express.js framework as the main backend and FastAPI as the prediction service, communicating via REST API. To address performance issues, a caching technique using the Stale-While-Revalidate strategy was applied. The system development methods included API specification design, database modification, and asynchronous interaction flow design. Testing was conducted functionally using Black Box Testing and performance testing using Artillery.io under high-load scenarios.*

*The results showed that all integration endpoints functioned as designed. The implementation of caching significantly improved system performance. In a load test of 2,265 requests, the average latency was drastically reduced from 2,200 ms (without cache) to 7 ms (with cache), and the request failure rate was eliminated from 37.17% to 0%. This architecture proves effective in serving learning style predictions in consistency and stably.*

*Keyword: Learning Style, Microservices, FastAPI, Express.js, Caching.*