

Implementasi GenAI dalam Sistem Pembuatan Butir Soal Otomatis dengan Pendekatan Microservices

Hafiih Herdian

*Program Studi Sistem Informasi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia*
hafiihherdian@student.telkom.univers
ity.ac.id

Oktariani Nurul Pratiwi

*Program Studi Sistem Informasi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia*

Faqih Hamami

*Program Studi Sistem Informasi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia*

Pembuatan soal yang relevan, bervariasi, dan sesuai dengan kurikulum merupakan tantangan besar bagi tenaga pengajar, terutama dengan keterbatasan waktu dan beban kerja administratif. Penelitian ini bertujuan mengembangkan sistem pembuatan butir soal otomatis berbasis Generative Artificial Intelligence (GenAI) dengan model Gemini 1.5 Flash dan pendekatan arsitektur microservices. Sistem menghasilkan soal pilihan ganda dan isian singkat dari input materi berupa teks, PDF, atau dokumen. Pengembangan dilakukan menggunakan metodologi Extreme Programming (XP) dengan implementasi teknologi Next.js, TypeScript, PostgreSQL, Docker, dan RabbitMQ. Hasil menunjukkan sistem mampu menghasilkan 10 soal dengan waktu rata-rata 8,5 detik dan menyediakan layanan modular, skalabel, serta responsif terhadap perubahan kebutuhan pengguna. Validasi dilakukan uji fungsional dan observasi pengguna (user acceptance test), serta menunjukkan efektivitas dan efisiensi sistem dalam konteks pendidikan digital.

Kata kunci— generative AI, Gemini 1.5 Flash, pembuatan soal otomatis, microservices, XP

I. PENDAHULUAN

Dalam dunia pendidikan, evaluasi pembelajaran memegang peranan penting sebagai alat ukur efektivitas proses belajar-mengajar [4]. Salah satu bentuk evaluasi yang krusial adalah penyusunan soal yang tidak hanya relevan dengan kurikulum, tetapi juga mampu mencerminkan variasi tingkat kognitif siswa. Proses ini seringkali menjadi tantangan bagi tenaga pendidik karena membutuhkan waktu, pemahaman mendalam terhadap materi, dan konsistensi dalam penyusunan soal. Beban administratif yang semakin kompleks juga turut membatasi kemampuan tenaga pengajar dalam menghasilkan soal secara optimal dan berkelanjutan [7].

Kemajuan dalam bidang kecerdasan buatan, khususnya Generative Artificial Intelligence (GenAI), membuka peluang baru untuk mengatasi tantangan tersebut. GenAI memungkinkan sistem untuk menghasilkan konten edukatif secara otomatis dan adaptif berdasarkan masukan yang diberikan [1]. Penerapan model Gemini 1.5 Flash sebagai bagian dari teknologi GenAI menunjukkan kemampuan tinggi dalam memahami konteks materi ajar dan mengonversinya menjadi soal dalam berbagai format. Namun, integrasi teknologi ini dalam lingkungan sistem

pendidikan membutuhkan pendekatan arsitektur yang skalabel, fleksibel, dan mudah dikembangkan.

Dengan menggunakan pendekatan microservices, sistem dapat dikembangkan dalam bentuk layanan-layanan modular yang independen dan saling berkomunikasi melalui antarmuka pemrograman aplikasi (API) [6]. Hal ini memungkinkan skalabilitas tinggi serta efisiensi dalam pengelolaan dan pengembangan sistem. Untuk memastikan sistem yang responsif terhadap kebutuhan pengguna dan mudah beradaptasi dengan perubahan, metodologi Extreme Programming (XP) diterapkan dalam proses pengembangan. XP menekankan iterasi singkat, komunikasi yang intensif, serta penerimaan masukan pengguna sebagai bagian dari siklus pengembangan.

Melalui penelitian ini, dikembangkan sebuah sistem pembuatan butir soal otomatis GenAI yang mampu mengolah materi pembelajaran dalam format teks, PDF, maupun dokumen dan menghasilkan soal pilihan ganda serta isian singkat secara adaptif. Sistem ini diharapkan tidak hanya mampu meningkatkan efisiensi kerja tenaga pengajar, tetapi juga menghadirkan solusi teknologi yang dapat diintegrasikan secara luas ke dalam platform pembelajaran digital.

II. KAJIAN TEORI

A. Generative Artificial Intelligence (GenAI)

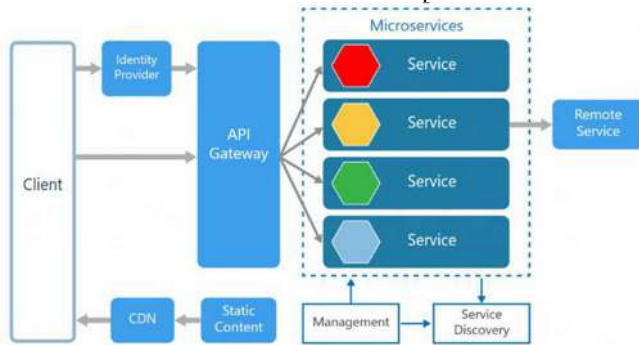
GenAI merupakan cabang dari kecerdasan buatan yang dirancang untuk menghasilkan konten baru seperti teks, gambar, maupun audio, berdasarkan pembelajaran dari data sebelumnya. Teknologi ini bekerja dengan model-model besar seperti transformer-based models (misalnya GPT, BERT), yang mampu memahami konteks dan membangun pola linguistik kompleks [1]. Dalam konteks pendidikan, GenAI digunakan untuk membuat soal ujian, ringkasan materi, serta konten pembelajaran lain secara otomatis. Kemampuan adaptif GenAI menjadikannya solusi untuk efisiensi waktu dan tenaga dalam penyusunan soal dengan mempertahankan kualitas dan variasi tingkat kesulitan.

B. Gemini 1.5 Flash

Gemini 1.5 Flash adalah salah satu model GenAI dari Google yang dirancang untuk memahami konteks panjang (long-context understanding) serta menghasilkan output dengan kecepatan tinggi. Model ini dapat diakses melalui API dan mendukung aplikasi dalam skenario pendidikan seperti pembuatan soal otomatis, karena mampu memahami isi dari dokumen seperti teks, PDF, dan DOC. Kemampuan ini menjadikannya pilihan tepat dalam pengembangan sistem ini, terutama dalam skenario yang membutuhkan personalisasi konten dan efisiensi waktu respons.

C. Microservices

Microservices adalah pendekatan arsitektur sistem yang membagi aplikasi menjadi beberapa layanan kecil, independen, dan terfokus pada satu fungsi spesifik. Setiap layanan dapat dikembangkan, diuji, dan di deploy secara terpisah, serta berkomunikasi melalui protokol ringan seperti HTTP REST atau message broker seperti RabbitMQ. Dalam sistem pembuatan butir soal otomatis, pendekatan ini memungkinkan pemisahan tugas antara layanan autentikasi, generasi soal, manajemen soal, dan pengiriman notifikasi. Hal ini meningkatkan skalabilitas sistem, mempercepat waktu pengembangan, dan memudahkan integrasi dengan sistem eksternal seperti LMS.



GAMBAR 1

D. Extreme Programming (XP)

Extreme Programming adalah metodologi pengembangan perangkat lunak yang termasuk dalam agile methods. Fokus utamanya adalah pada iterasi cepat, komunikasi antar tim yang kuat, serta penerimaan dan integrasi perubahan kebutuhan pengguna secara berkelanjutan. XP melibatkan lima tahapan utama: planning, design, coding, testing, dan listening (perbaikan berdasarkan feedback pengguna) [3]. Dalam penelitian ini, XP digunakan untuk memastikan sistem yang dibangun responsif terhadap kebutuhan aktual tenaga pengajar, serta dapat berkembang secara adaptif dalam jangka panjang.

III. METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan metode Extreme Programming (XP), yang merupakan salah satu metode dalam Agile Development. XP dipilih karena kemampuannya dalam mendukung pengembangan sistem secara iteratif, adaptif, dan responsif terhadap umpan balik pengguna secara cepat.

Life Cycle of Extreme Programming (XP)



GAMBAR 2

A. Tahapan dan Pengembangan

Tahapan dalam XP yang diterapkan dalam penelitian ini mencakup:

1. Planning, yaitu mengidentifikasi kebutuhan sistem berdasarkan masukan (tenaga pengajar) untuk fitur utama seperti autentikasi, upload materi pembelajaran, pembuatan soal otomatis, pengelolaan soal, dan notifikasi.
2. Design, yakni mendesain arsitektur sistem berbasis microservices, dengan layanan terpisah seperti Auth Service, Generate Soal Service, Manage Soal Service, Notification Service, dan API Gateway sebagai pengatur lalu lintas antar layanan.
3. Coding, yaitu implementasi fungsionalitas sistem menggunakan teknologi Next.js (frontend dan gateway), Express.js (backend), serta integrasi API Gemini 1.5 Flash sebagai model GenAI untuk membuat butir soal secara otomatis.
4. Testing, mencakup uji fungsional terhadap endpoint API, pengujian integrasi antar layanan, serta validasi output soal yang dihasilkan oleh GenAI.
5. Listening, yaitu evaluasi berbasis feedback pengguna untuk penyempurnaan sistem, termasuk penyesuaian format soal, pengalaman antarmuka, dan kecepatan respons sistem.

B. Teknologi dan Alat Pendukung

Pengembangan dilakukan menggunakan Next.js (React dan TypeScript) untuk frontend, Docker untuk containerisasi layanan, RabbitMQ sebagai message broker, PostgreSQL sebagai basis data, serta integrasi API eksternal Gemini 1.5 Flash untuk pemrosesan generasi soal. Prisma digunakan sebagai ORM untuk mengelola data secara efisien dan terstruktur.

C. Metode Evaluasi

Pengujian sistem dilakukan secara menyeluruh untuk memastikan fungsionalitas berjalan dengan baik, komunikasi antar layanan berjalan stabil, dan sistem memberikan pengalaman pengguna yang optimal. Fokus utama pengujian adalah:

- Fungsionalitas aplikasi
Menguji semua fitur utama seperti autentikasi, *generate* soal, *manage* soal, dan notifikasi.
- Integrasi dengan GenAI (Gemini 1.5 Flash)
Menguji apakah layanan dapat berinteraksi dengan baik dengan model GenAI dan menangani *output* yang tidak terstruktur.

Metode pengujian yang diterapkan:

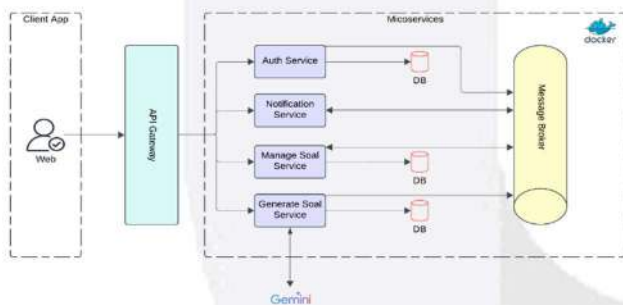
1. Pengujian Unit
Pengujian fungsi individual pada setiap *service*, seperti validasi *input*, transformasi data, dan *error handling*.
2. Pengujian Integrasi
Pengujian komunikasi antar *service* melalui API Gateway dan RabbitMQ.
3. Pengujian End-to-End
Simulasi penggunaan lengkap dari sisi pengguna, mulai dari *login*, unggah materi, *input* materi, hingga pengelolaan soal dan notifikasi.
4. Pengujian API
Pengujian *endpoint* dilakukan secara manual menggunakan Postman, *endpoint* yang diuji untuk status, respons, validasi autentikasi, dan hak akses.
5. Pengujian Kinerja (Load Testing)

Pengujian kinerja dilakukan menggunakan k6 untuk menyimulasikan beban pengguna yang tinggi pada fitur-fitur kritis, terutama pada fungsionalitas yang bergantung pada layanan eksternal seperti GenAI.

IV. HASIL DAN PEMBAHASAN

A. Arsitektur Sistem Microservices

Sistem yang di kembangkan terdiri dari beberapa *microservices* independen yang berinteraksi melalui API Gateway. Web Client berfungsi sebagai antarmuka pengguna. Komunikasi antar layanan, terutama untuk notifikasi dan tugas asinkron, di fasilitasi oleh *message queue* [5].



GAMBAR 3

B. Hasil Pengujian

Pengujian kinerja difokuskan pada fitur generate soal yang merupakan proses paling intensif. Skenario pengujian dirancang untuk mengukur stabilitas dan skalabilitas sistem di bawah tekanan.

TABEL 1

Skenario Pengujian	Hasil Pengujian
1 pengguna Virtual (VU) dengan 10 Iterasi	<ul style="list-style-type: none"> - Success Rate: 100% (10/10 request berhasil) - Latensi: <ul style="list-style-type: none"> - Rata-rata: 8428ms (8.43s) - Minimum: 7901ms (7.9s)

	<ul style="list-style-type: none"> - Maksimum: 9022ms (9.02s) - P(95): 8963ms (8.96s) - Error Rate: 0% - Throughput: 0.118 iterasi/detik - Durrasi eksekusi: 1m24.4s - Resource Usage: <ul style="list-style-type: none"> - Data recieved: 54kB - Data sent: 32kB <p>VU Aktif: 1 virtual user konstan</p>
5 pengguna Virtual (VU) dengan 10 Iterasi	<ul style="list-style-type: none"> - Success Rate: 100% (10/10 request berhasil) - Latensi: <ul style="list-style-type: none"> - Rata-rata: 8567ms (8.57s) - Minimum: 8144ms (8.14s) - Maksimum: 9255ms (9.25s) - P(95): 9208ms (9.21s) - Error Rate: 0% - Throughput: 0.566 iterasi/detik - Durrasi eksekusi: 17.7s - Resource Usage: <ul style="list-style-type: none"> - Data recieved: 53kB - Data sent: 32kB <p>VU Aktif: 5 virtual user</p>
10 pengguna Virtual (VU) dengan 10 Iterasi	<ul style="list-style-type: none"> - Success Rate: 100% (10/10 request berhasil) - Latensi: <ul style="list-style-type: none"> - Rata-rata: 8623ms (8.62s) - Minimum: 8110ms (8.11s) - Maksimum: 9654ms (9.65s) - P(95): 9517ms (9.52s) - Error Rate: 0% - Throughput: 1.117 iterasi/detik - Durrasi eksekusi: 9.8s - Resource Usage: <ul style="list-style-type: none"> - Data recieved: 53kB - Data sent: 32kB <p>VU Aktif: 10 virtual user konstan</p>
15 pengguna Virtual (VU) dengan 15 Iterasi	<ul style="list-style-type: none"> - Success Rate: 40% (6/15 request berhasil) - Error Rate: 56.25% (9/16 request gagal) - Latensi: <ul style="list-style-type: none"> - Rata-rata: 3768ms (3.77s) - Minimum: 7300ms (7.3s) - Maksimum: 8726ms (8.73s) - P(95): 8963ms (8.96s) - Throughput: 1.701 iterasi/detik - Durrasi eksekusi: 8.8s - Error Detail: <ul style="list-style-type: none"> - Error kode 500 - Rate Limit tercapai <p>VU Aktif: 6-15 virtual user konstan</p>

V. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan sistem pembuatan butir soal otomatis berbasis Generative Artificial Intelligence (GenAI) dengan model Gemini 1.5 Flash dan arsitektur microservices, dapat disimpulkan bahwa sistem ini mampu memberikan solusi inovatif dalam otomatisasi pembuatan soal. Pertama, penerapan model GenAI Gemini 1.5 Flash berhasil menginterpretasi materi pembelajaran dalam berbagai format, seperti PDF, DOC, maupun input langsung dari pengguna. Materi ini kemudian dikonversi menjadi soal pilihan ganda dan isian singkat dalam struktur JSON yang telah distandarkan. Proses ini difasilitasi melalui integrasi

endpoint seperti generate-soal/extract-content dan generate-soal/questions, yang memungkinkan konversi dilakukan secara otomatis dan efisien tanpa membutuhkan komputasi lokal yang besar. Selain itu, validasi format JSON yang diterapkan mampu menangani ketidakteraturan struktur output dari model, sehingga menjaga konsistensi data.

Kedua, sistem dirancang dengan pendekatan arsitektur microservices yang terdiri dari beberapa layanan modular, yaitu Auth Service untuk autentikasi dan manajemen pengguna, Generate Soal Service untuk ekstraksi materi dan pembuatan soal, Manage Soal Service untuk manajemen soal oleh pengguna, serta Notification Service yang mendukung notifikasi realtime melalui WebSocket. Komunikasi antar layanan dikelola secara terpusat dan efisien dengan bantuan API Gateway dan RabbitMQ, memungkinkan sistem untuk menangani permintaan simultan dengan waktu respons rata-rata 1–2 menit untuk menghasilkan sepuluh soal.

Ketiga, penggunaan metodologi Extreme Programming (XP) memberikan dampak signifikan terhadap pengembangan sistem. Iterasi cepat dan umpan balik langsung dari pengguna mendorong terjadinya peningkatan berkelanjutan pada sistem. Beberapa perbaikan penting yang dilakukan antara lain validasi struktur output dari GenAI, penanganan file besar dengan ukuran lebih dari 10MB, serta penambahan fitur pencarian dan filter soal. Hal ini menunjukkan bahwa pendekatan XP sangat sesuai untuk proyek yang bersifat dinamis dan memerlukan adaptasi cepat terhadap kebutuhan pengguna.

Dengan demikian, keseluruhan sistem yang dibangun berhasil menjawab rumusan masalah dengan menyediakan

solusi otomatisasi pembuatan butir soal yang adaptif, efisien, dan terintegrasi dalam satu platform pembelajaran yang modern dan fleksibel.

REFERENSI

- [1] L. Chen, P. Chen, dan Z. Lin, "Artificial Intelligence in Education: A Review," *AI in Education*, vol. 7, pp. 12–29, 2020.
- [2] X. Chen, D. Zou, H. Xie, G. Cheng, and C. Liu, "Two Decades of Artificial Intelligence in Education,"
- [3] A. Shrivastava et al., "A Systematic Review on Extreme Programming," *International Journal of Software Engineering*, vol. 9, pp. 55–61, 2021.
- [4] R. Luckin dan M. Cukurova, "Designing Educational Technologies in The Age of AI," *British Journal of Educational Technology*, vol. 50, no. 6, pp. 2824–2841, 2019.
- [5] N. Dragoni et al., "Microservices: Yesterday, Today, and Tomorrow," *Springer Series in Software Engineering*, 2017.
- [6] D. Taibi, V. Lenarduzzi, dan C. Pahl, "Architectural Patterns for Microservices: A Systematic Mapping Study," *Proceedings of the 8th International Conference on Cloud Computing and Services Science*, 2018.
- [7] I. Magdalena, H. Fauzi, dan I. Putri, "Pentingnya Evaluasi Dalam Pembelajaran dan Akibat Memanipulasinya," *Jurnal Pendidikan*, 2020.