

MICROGEN IMPLEMENTATION FOR BUILDING ONLINE LEARNING MANAGEMENT SYSTEM WITH MICROSERVICES AND GRAPHQL GENERATOR APPROACH

Daniel Hasiholan Tinambunan¹, Ahmad Baehaqi², Refgiufi Patria Avrianto³, Richardus Eko Indrajit⁴

Email: ¹daniel.hasiholan@student.pradita.ac.id, ²ahmad.baehaqi@student.pradita.ac.id,
³refgiufi.patria@student.pradita.ac.id, ⁴eko.indrajit@pradita.ac.id

(Article received: August 9, 2023; Revision: August 18, 2023; published: August 25, 2023)

Abstract

The development of a Learning Management System (LMS) has become a major focus in modernizing the educational approach. However, the complexities of designing an efficient database, consistently manipulating data, adaptively managing services, and building responsive interfaces are challenges that need to be overcome. In dealing with these problems, the purpose of this research is to provide practical guidance for LMS developers in combining innovative concepts such as Microgen, Microservices, and GraphQL Generator. The research method adopted includes literature analysis, database schema implementation using Microgen, self-service development using the Microservices approach, interactive interface implementation through the GraphQL Generator, as well as testing the resulting LMS performance. The results of the implementation show that the integration of the three concepts has succeeded in overcoming the existing problems. Through the use of Microgen, the database design process becomes easier and allows for consistent data manipulation. The Microservices approach enables more adaptive and efficient management of LMS services. Meanwhile, the use of the GraphQL Generator has succeeded in producing an LMS interface that is responsive to various devices. Overall, this study concludes that a development approach that combines the concepts of Microgen, Microservices, and GraphQL Generator is capable of producing an innovative LMS that is ready to face the challenges of the latest educational technology. With the resulting guide, the developers have a foundation for designing an LMS that incidentally is efficient, adaptive and responsive to meet the needs and expectations of users in the modern education era.

Keywords: *Microgen, Learning Management System (LMS), Microservices, GraphQL Generator*

PENERAPAN MICROGEN DALAM MEMBANGUN LEARNING MANAGEMENT SYSTEM DENGAN PENDEKATAN MICROSERVICES DAN GRAPHQL GENERATOR

Abstrak

Pengembangan Learning Management System (LMS) telah menjadi fokus utama dalam memodernisasi pendekatan pendidikan. Namun, kompleksitas dalam merancang basis data yang efisien, memanipulasi data secara konsisten, mengelola layanan secara adaptif, dan membangun antarmuka yang responsif merupakan tantangan yang perlu diatasi. Dalam menghadapi permasalahan tersebut, tujuan penelitian ini adalah memberikan panduan praktis bagi pengembang LMS dalam menggabungkan konsep-konsep inovatif seperti Microgen, Microservices, dan GraphQL Generator. Metode penelitian yang diadopsi mencakup analisis literatur, penerapan skema basis data menggunakan Microgen, pengembangan layanan mandiri dengan pendekatan Microservices, implementasi antarmuka interaktif melalui GraphQL Generator, serta pengujian kinerja LMS yang dihasilkan. Hasil implementasi menunjukkan bahwa integrasi ketiga konsep tersebut berhasil mengatasi permasalahan yang ada. Melalui penggunaan Microgen, proses perancangan basis data menjadi lebih mudah dan memungkinkan manipulasi data yang konsisten. Pendekatan Microservices memungkinkan pengelolaan layanan LMS secara lebih adaptif dan efisien. Sementara itu, penggunaan GraphQL Generator berhasil menghasilkan antarmuka LMS yang responsif terhadap berbagai perangkat. Secara keseluruhan, penelitian ini menyimpulkan bahwa pendekatan pengembangan yang menggabungkan konsep Microgen, Microservices, dan GraphQL Generator mampu menghasilkan LMS inovatif yang siap menghadapi tantangan teknologi pendidikan terkini. Dengan adanya panduan yang dihasilkan, para pengembang memiliki landasan untuk merancang LMS yang notabene efisien, adaptif, dan responsif guna memenuhi kebutuhan dan ekspektasi pengguna dalam era pendidikan modern.

Kata kunci: : *Microgen, Learning Management System (LMS), Microservices, GraphQL Generator*

1. PENDAHULUAN

Di era digital yang berkembang pesat, pendidikan telah mengalami transformasi mendasar dalam cara belajar dan mengajar dilakukan[1][2]. *Learning Management System (LMS)* telah memainkan peran penting dalam mendukung model pembelajaran jarak jauh yang semakin populer, memungkinkan akses ke materi pembelajaran online, kolaborasi guru-siswa, dan penilaian terintegrasi [3]. Namun, mengembangkan LMS yang responsif, efisien, dan terukur tetap menjadi tantangan utama [4]. Dalam konteks tersebut, penelitian ini bertujuan untuk menerapkan konsep *Microgen* dalam membangun LMS dengan pendekatan *Microservices* dan *GraphQL Generator* untuk mengatasi tantangan pengembangan tersebut.

Konsep *Microgen* yang menggabungkan pendekatan *Microservices* dan *GraphQL Generator* telah terbukti memiliki potensi besar dalam pengembangan sistem informasi yang kompleks [5][6]. Dalam penelitian ini, kami mengeksplorasi penerapan *Microgen* dalam membangun LMS yang inovatif dan efektif bernama LMS Pradita. Kami menggunakan *MgSchema* untuk merancang skema basis data, yang menyediakan visualisasi untuk perencanaan struktur data yang lebih efisien [7].

Adopsi fungsionalitas *Microgen* adalah langkah selanjutnya dalam manipulasi data di LMS Pradita. Fungsi *mikrogen* memungkinkan pengembangan dengan cepat dan konsisten membuat fungsi pengikatan data [8][9]. Kami juga mengadopsi pendekatan *microservices*, di mana setiap fitur LMS diimplementasikan sebagai layanan mikro independen[10]. Pendekatan ini menguntungkan skalabilitas dan pemeliharaan sistem[11], dengan kemampuan untuk memperluas atau meningkatkan fungsionalitas tanpa mengganggu bagian lain dari sistem [12].

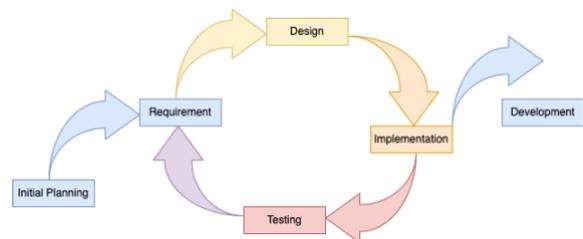
Melalui penggunaan *GraphQL Generator*, kami juga mengatasi tantangan pengembangan antarmuka dengan cara yang lebih efisien dan responsif [13]. Menggunakan *GraphQL Generator* menyediakan cara interaktif untuk berinteraksi dengan API yang ada, mempercepat siklus pengembangan, dan memastikan antarmuka yang responsif dan personal [14][15].

Secara keseluruhan, penelitian ini memberikan wawasan tentang bagaimana implementasi *Microgen*, pendekatan *microservices*, dan *GraphQL Generator* dapat mengatasi tantangan kompleks pengembangan LMS. Hasil penelitian ini tidak hanya berpotensi memberikan panduan bagi pengembang, akademisi, dan praktisi di bidang teknologi pendidikan, tetapi juga menyumbangkan solusi inovatif untuk memenuhi tuntutan pendidikan modern yang terus berkembang. Dengan tujuan yang jelas, penelitian ini bertujuan untuk memberikan sumbangan berarti

dalam merancang dan mengembangkan solusi pendidikan yang lebih adaptif dan responsif.

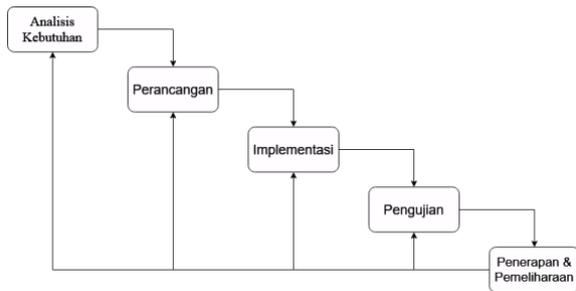
2. METODE PENELITIAN

Pengembangan sistem backend LMS Pradita akan menggunakan pendekatan SDLC iteratif, artinya kebutuhan akan ditentukan sesuai dengan kebutuhan sistem akhir dan akan dieksekusi secara iteratif. Cara ini akan lebih cocok karena pengembangan sistem backend akan dipecah menjadi banyak *microservices*, sehingga setiap fungsi sistem *frontend* akan memanggil setiap *microservice* atau sistem *backend* yang ada. Hasil yang diharapkan dari proses pengembangan dikomunikasikan dengan jelas kepada klien di awal proyek sehingga proyek dapat direncanakan secara jelas dari awal hingga akhir dan dilaksanakan sesuai dengan tujuan.[16] Hal ini juga akan berpengaruh baik terhadap kinerja sistem LMS Pradita yang sedang dikembangkan, karena apabila terjadi kegagalan pada salah satu layanan maka tidak akan mempengaruhi layanan lainnya. Dalam metode di atas, banyak proses dimulai ketika sistem antarmuka pengguna meminta proses utama untuk menjalankan suatu fungsi atau fitur. Gambaran dari alur pengembangan pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Model Metode SDLC

Penelitian ini juga menerapkan pendekatan metode *waterfall* sebagai landasan strategis dalam merancang dan mengimplementasikan LMS Pradita berbasis online dengan mengintegrasikan pendekatan *Microgen*, *Microservices*, dan *GraphiQL Generator*. Metode *waterfall* yang dipilih merupakan salah satu pendekatan pengembangan perangkat lunak yang terstruktur dan linier. Keuntungan dari metode ini adalah seseorang dapat merancang sistem secara berurutan dan ketika ada perbaikan, dia dapat langsung ke langkah perbaikan tanpa harus mengulang dari awal[17]. Metode ini memiliki tahapan berurutan yang meliputi analisis kebutuhan, perancangan, implementasi, pengujian, serta penerapan dan pemeliharaan sistem seperti yang terlihat pada gambar 2.



Gambar 2. Tahapan Model Waterfall

- Analisis Kebutuhan

Pada tahap analisis kebutuhan, dilakukan interaksi mendalam dengan para pemangku kepentingan dan pihak terkait untuk memperoleh pemahaman menyeluruh tentang persyaratan dan tujuan esensial LMS Pradita berbasis online. Proses analisis yang cermat mencakup penentuan fitur-fitur yang diinginkan, kebutuhan fungsional yang diharapkan, dan kriteria kinerja yang relevan guna membentuk dasar spesifikasi sistem yang komprehensif.

- Perancangan

Langkah berikutnya adalah perancangan, di mana tim peneliti merancang secara terinci berbagai aspek sistem, termasuk arsitektur yang cocok untuk mengakomodasi teknologi Microgen, pemecahan modul dalam kerangka microservices, serta penggunaan GraphQL Generator untuk menciptakan antarmuka pengguna interaktif yang mendukung pengelolaan dan analisis konten pembelajaran.

- Implementasi

Tahap implementasi menyusul perancangan, di mana tim melakukan proses penerapan dan pemrograman berdasarkan desain sebelumnya. Tahap ini berjalan secara bertahap sesuai dengan prinsip waterfall, di mana setiap tahap harus selesai dengan baik sebelum melanjutkan ke tahap berikutnya.

- Pengujian

Tahap pengujian menjadi krusial dalam metode waterfall, di mana sistem yang telah diimplementasikan akan diuji secara menyeluruh guna memverifikasi kinerja yang optimal, mengidentifikasi dan mengatasi kesalahan atau bug, serta memastikan sistem memenuhi persyaratan dan standar yang telah ditetapkan.

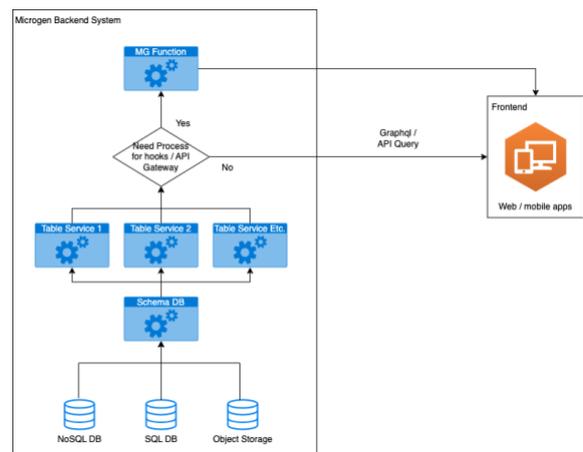
- Penerapan dan Pemeliharaan Sistem

Terakhir, LMS Pradita berbasis online yang telah diimplementasikan akan diintegrasikan dan dijalankan secara keseluruhan. Setelah melalui tahap implementasi dan pengujian yang cermat, sistem siap digunakan oleh para pengguna, termasuk siswa dan pendidik, untuk mendukung proses pembelajaran secara online dengan efisien dan adaptif sesuai

kebutuhan masyarakat pendidikan di era dinamis teknologi digital.

2.1. Microgen

Microgen adalah suatu pendekatan inovatif dalam pengembangan perangkat lunak yang menggabungkan prinsip *model-driven development* (MDD) dengan generasi kode otomatis. Pendekatan ini bertujuan untuk mempercepat proses pengembangan dan meningkatkan produktivitas dalam pembuatan perangkat lunak dengan mengandalkan alat yang dapat secara otomatis menghasilkan kode berdasarkan model dan definisi yang telah disediakan.[18] Microgen membantu mengatasi kompleksitas dalam proses pengembangan dengan menyediakan pemodelan domain yang tepat dan menghasilkan kode otomatis sesuai dengan aturan dan logika yang telah ditentukan sebelumnya. Dengan pendekatan ini, para pengembang dapat fokus pada aspek esensial dari perangkat lunak yang hendak dikembangkan tanpa perlu memikirkan implementasi detail, sehingga memungkinkan proses pengembangan yang lebih cepat, konsisten, dan efisien.



Gambar 3. High Level Design Microgen Platform

Sistem Microgen memiliki beberapa bagian yang digunakan dalam memproses service backend (gambar 3). Pada dasarnya sistem backend memiliki penyimpanan berupa database hanya saja sistem backend pada umumnya akan memiliki spesifik database yang akan digunakan seperti NoSQL atau SQL. Microgen memiliki beberapa penyimpanan yang dapat digunakan untuk menyimpan data seperti NoSQL DB, SQL DB dan Object Storage. Setiap penyimpanan memiliki fungsi yang dapat digunakan oleh Schema DB yang akan dibuat pada platform microgen. Schema tersebut akan menjadi sebuah table-table yang akan menjadi sebuah service yang dapat digunakan oleh aplikasi baik itu mobile ataupun web. Apabila seorang pengembang akan membuat sebuah sistem backend pada dasarnya pengembang akan menuliskan sebuah code untuk memanipulasi dan menyimpan data. Berbeda jika menggunakan

platform microgen maka pengembang hanya perlu mendefinisikan sebuah schema database yang kemudian setiap tabel pada database tersebut akan menjadi sebuah bagian microservice di dalam sistem microgen. Hal tersebut dapat dilakukan oleh pengembang tanpa harus menuliskan sebuah code, sehingga platform microgen juga bisa disebut sebuah alat yang disebut sebagai Low Code.[19] Selain itu untuk properti pada tabel tidak akan menggunakan tipe data primitif yang hanya spesifik pada SQL yang digunakan. Pada microgen akan terdapat beberapa tipe data yang lebih spesifik seperti penyimpanan untuk file atau sebuah email dan kontak. Dari setiap service tabel yang ada pada microgen dapat digunakan langsung untuk digunakan sistem frontend, atau pengembang bisa menggunakan MgFunction untuk memproses data apabila terdapat spesifik alur kerja yang diperlukan oleh sistem frontend. Pengembang dapat menuliskan sebuah code pada MgFunction jika membutuhkan spesifik proses untuk mengelola data pada service backend. Sistem frontend akan menggunakan sebuah query baik itu GraphQL maupun API untuk mengelola data[8] pada backend service microgen.

2.2. *Learning Management System (LMS)*

Learning Management System (LMS) merupakan sebuah platform berbasis teknologi informasi yang bertujuan untuk menyediakan lingkungan pembelajaran virtual yang efisien dan terorganisir bagi institusi pendidikan, organisasi bisnis, atau lembaga lainnya.[20] LMS Pradita berfungsi sebagai sarana utama dalam mengelola dan mengkoordinasikan seluruh proses pembelajaran secara daring, menghubungkan peserta didik dengan pendidik serta berbagai sumber daya pembelajaran dengan cara yang interaktif dan terstruktur. Platform ini menjadi motor penggerak untuk menghadirkan pendidikan jarak jauh yang canggih dan terjangkau, mengatasi batasan-batasan geografis dan waktu yang biasanya menjadi hambatan dalam pembelajaran tradisional.

LMS Pradita mengintegrasikan beragam fitur dan fungsi yang menyelaraskan elemen pembelajaran, termasuk menyediakan dan mengelola konten pembelajaran, tugas, kuis, diskusi, serta alat evaluasi untuk mengukur kinerja peserta didik. Dalam konteks pendidikan formal, LMS Pradita menyediakan jalur kurikulum yang terstruktur, memfasilitasi pengiriman materi pembelajaran dalam berbagai format, seperti teks, video, audio, dan gambar. Keunggulan utama LMS Pradita adalah fitur kolaborasi yang memungkinkan peserta didik dan pendidik berinteraksi secara aktif melalui forum diskusi, ruang kelas maya, dan alat komunikasi lainnya. Selain itu, LMS Pradita juga menyediakan analisis kinerja untuk memantau kemajuan belajar siswa, memungkinkan pendidik untuk memberikan bimbingan yang lebih tepat sasaran, serta memberikan laporan dan evaluasi kinerja untuk

menilai efektivitas kurikulum dan meningkatkan kualitas pembelajaran secara keseluruhan. Dengan demikian, LMS Pradita menjadi landasan penting dalam era pendidikan digital yang dinamis, memungkinkan lembaga pendidikan dan organisasi untuk menciptakan pengalaman pembelajaran inklusif dan adaptif, yang sesuai dengan kebutuhan dan harapan masyarakat dalam perubahan dunia pendidikan.

2.3. *Microservices*

Microservice adalah pendekatan arsitektur perangkat lunak yang memecah aplikasi menjadi serangkaian layanan mandiri, terpisah, dan independen yang berkomunikasi melalui mekanisme tertentu, seperti protokol HTTP.[21] Setiap layanan ini berfokus pada satu fungsi atau fitur khusus dalam aplikasi dan dapat dikembangkan, diperbarui, atau di deploy secara terpisah tanpa mempengaruhi bagian lain dari aplikasi secara keseluruhan. Pendekatan ini bertujuan untuk meningkatkan fleksibilitas dan skalabilitas sistem, memungkinkan pengembang untuk bekerja pada bagian-bagian aplikasi secara terisolasi, mengurangi dampak perubahan, serta memungkinkan penggunaan bahasa pemrograman atau teknologi yang berbeda dalam setiap layanan.[22]

Dalam implementasinya, microservice juga berarti bahwa aplikasi terdiri dari sejumlah layanan yang bekerja bersama untuk menyusun sistem yang lebih besar. Masing-masing layanan ini memiliki basis data sendiri dan bertanggung jawab atas penyimpanan dan pemrosesan data yang berkaitan dengan fungsionalitas khususnya. Interaksi antara layanan biasanya menggunakan mekanisme seperti API REST atau gRPC, yang memungkinkan berkomunikasi secara terstruktur dan efisien. Pendekatan microservice juga menggabungkan prinsip loose coupling[23], di mana setiap layanan beroperasi secara independen, sehingga jika salah satu layanan mengalami gangguan, layanan lainnya tetap berfungsi tanpa gangguan. Dalam keseluruhan, pendekatan arsitektur microservice telah menjadi solusi yang semakin populer dalam pengembangan perangkat lunak modern, memberikan keuntungan seperti kemampuan untuk membangun dan mengelola sistem yang kompleks dengan lebih mudah, serta meningkatkan fleksibilitas dan responsivitas aplikasi.

2.4. *GraphQL Generator*

Dengan mengembangkan LMS Pradita dengan pendekatan Microgen, salah satu komponen utama yang digunakan adalah GraphQL Generator. GraphQL Generator adalah alat yang memudahkan pembuatan antarmuka API menggunakan bahasa kueri GraphQL. GraphQL adalah bahasa kueri yang fleksibel dan efisien untuk mengekstraksi data dari server[24], memungkinkan pengembang untuk

mengekstraksi hanya data yang mereka butuhkan tanpa ekstraksi berlebih atau kurang.

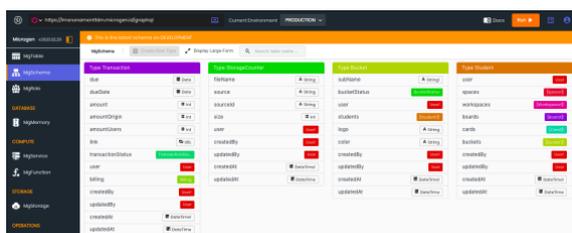
GraphQL Generator menyederhanakan pengembangan antarmuka dengan menyediakan lingkungan interaktif[25] yang disebut GraphQL. GraphQL adalah IDE (lingkungan pengembangan terintegrasi) yang terintegrasi mulus dengan server GraphQL, memungkinkan pengembang untuk menjelajahi dan menguji kueri dan mutasi yang tersedia di API dengan cepat. Dengan antarmuka GraphQL yang responsif, pengembang dapat melihat struktur data yang ada, menjalankan kueri, dan melihat hasil secara real time.

Sebagai bagian dari penelitian ini, GraphQL Generator memainkan peran kunci dalam membangun API yang efisien dan terdokumentasi dengan baik. Dengan menggunakan GraphQL Generator, LMS Pradita dapat menentukan skema GraphQL yang menjelaskan tipe data yang ada dan operasi yang dapat dilakukan.[26] Kemudian, GraphQL memungkinkan pengembang untuk menguji kueri yang akan digunakan oleh UI, memastikan kebenaran dan efisiensi komunikasi antara UI dan UI.[6]

Penerapan GraphQL Generator dalam penelitian ini mencapai tujuan untuk menciptakan antarmuka yang interaktif dan responsif, sekaligus menghindari masalah *overfetching* atau data yang hilang. Menggunakan GraphQL Generator memberikan pendekatan yang lebih terstruktur dan terdokumentasi untuk membangun API[12], yang sangat relevan dalam mengembangkan sistem informasi yang kompleks dan dapat diskalakan seperti Pradita LMS.

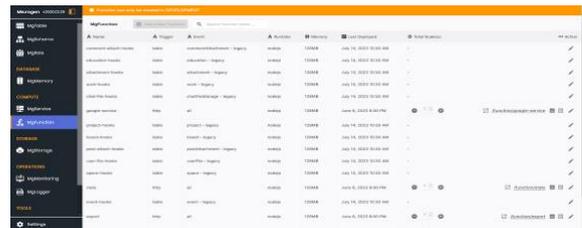
3. HASIL DAN PEMBAHASAN

Dalam studi ini, kami mengeksplorasi penerapan konsep Microgen dalam mengembangkan LMS Pradita yang inovatif menggunakan Microservices dan GraphQL Generator. Penulis telah berhasil membuat LMS Pradita adaptif yang dapat diskalakan dengan antarmuka interaktif. Proses pengembangan diawali dengan perancangan skema database menggunakan visualisasi MgSchema yang memvisualisasikan struktur data yang akan digunakan dalam sistem. Gambar 4 menunjukkan MgSchema memberikan visualisasi yang sangat berguna untuk mendefinisikan dan merencanakan entitas dan hubungan di antara mereka.



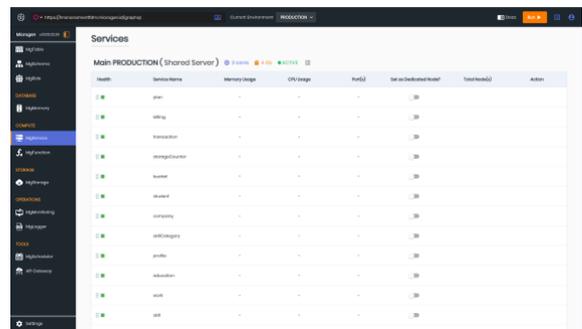
Gambar 4. MgSchema digunakan untuk membuat *table schema* sesuai kebutuhan

Setelah skema database ditentukan, langkah selanjutnya adalah menerapkan fungsi Microgen untuk memanipulasi data. Fungsi Microgen memungkinkan pengembang untuk membuat fungsi pengikatan data dengan cepat dan konsisten. Di LMS Pradita, kami menggunakan fungsi Microgen untuk mengelola proses seperti membuat akun pengguna, menjadwalkan pelajaran, dan melaporkan hasil pembelajaran. Keuntungan utama dari fungsi Microgen adalah fleksibilitasnya yang besar, memungkinkan pengembang menyesuaikan fungsi ini sesuai dengan kebutuhan sistem.



Gambar 5. Pada MgFunction dapat menentukan proses manipulasi data yang ada pada Microgen

Penting untuk dicatat bahwa pada Gambar 5, kita dapat melihat bahwa di *MgFunction* pada Microgen kita dapat menentukan prosedur manipulasi data yang ada. Selain itu, kami berhasil menjelaskan penerapan konsep *microservices* dalam pengembangan LMS Pradita. Setiap fitur LMS Pradita diimplementasikan sebagai layanan mikro independen dengan tanggung jawab khusus. Pendekatan ini menawarkan keuntungan besar dalam hal skalabilitas dan pemeliharaan sistem. Saat mengembangkan fitur baru, kami dapat melakukan perubahan atau peningkatan tanpa memengaruhi komponen sistem lainnya. Pengelolaan proses pengembangan dan pengujian juga ditingkatkan karena setiap layanan mikro dapat diisolasi dan diuji secara independen, sehingga mengurangi risiko potensi komplikasi.

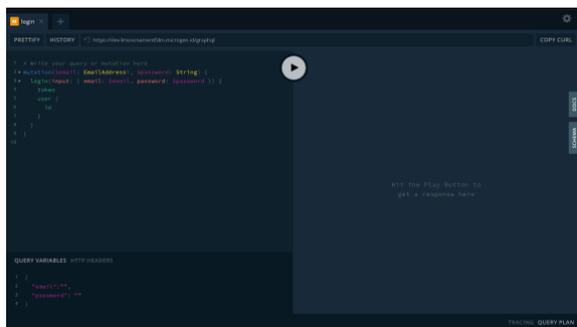


Gambar 6. Microservices pada Microgen

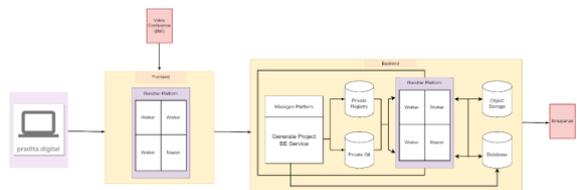
Selanjutnya, pada Gambar 6, terlihat bagaimana *microservices* diaplikasikan dalam konteks Microgen. Selain itu, penggunaan GraphQL Generator juga membawa dampak signifikan dalam proses pengembangan. Dengan memanfaatkan

GraphQL Generator, pengembang dapat berinteraksi secara langsung dengan API yang sudah ada, menguji kueri dan mutasi, serta menerima hasil dengan cepat. Ini berimplikasi pada efisiensi dan responsivitas yang lebih baik dalam proses pengembangan antarmuka, sekaligus mengurangi risiko kesalahan yang mungkin muncul selama komunikasi antara *front-end* dan *back-end*.

Terakhir, pada Gambar 7 dan Gambar 8, secara berurutan menggambarkan GraphQL Generator pada Microgen dan alur infrastruktur LMS Pradita. Dua komponen ini menjadi bagian integral dari penelitian ini, mendukung dan memperkuat efektivitas konsep Microgen, pendekatan *microservices*, serta kemudahan dalam interaksi antarmuka yang dihasilkan. Melalui keseluruhan penjelasan ini, tampak dengan jelas bagaimana teknologi dan metodologi terkini berdampak signifikan dalam menangani kompleksitas pengembangan sistem informasi yang menuntut, seperti halnya dalam pengembangan LMS Pradita.

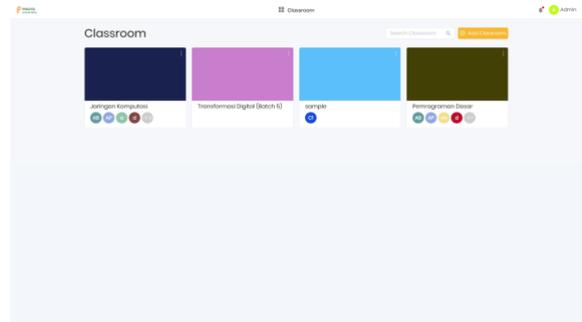


Gambar 7. GraphQL Generator pada Microgen



Gambar 8. Alur Infrastruktur LMS Pradita

Akhirnya, semua sistem LMS Pradita yang dikembangkan berhasil diterapkan. Penggunaan ekstensif konsep Microgen, pendekatan *Microservices*, dan generator GraphQL telah menghasilkan implementasi LMS Pradita yang tidak hanya aman dan terjamin, tetapi juga dapat diadaptasi dan diskalakan sesuai kebutuhan. Keberhasilan tersebut tercermin jelas pada Gambar 9 yang menggambarkan bahwa LMS Pradita berfungsi normal dan sesuai harapan melalui penerapan konsep Microgen. Kondisi ini membuka jalan bagi solusi inovatif dalam pengembangan sistem pembelajaran yang mampu menjawab tantangan dan dinamika pendidikan modern yang terus berkembang.



Gambar 9. LMS Pradita berjalan dengan baik menggunakan konsep Microgen

Hasil penelitian ini dengan jelas menggambarkan kemampuan teknologi dan metode terbaru memiliki dampak positif yang sangat signifikan dalam mengatasi kompleksitas pengembangan sistem informasi seperti LMS. Gambar 9 dengan jelas menunjukkan bahwa LMS Pradita bekerja dengan sempurna berkat penerapan konsep Microgen. Secara keseluruhan, hasil penelitian ini mengonfirmasi bahwa integrasi pendekatan Microgen, *microservices*, dan GraphQL Generator memberikan fondasi yang kokoh untuk mengembangkan LMS yang efisien, adaptif, dan responsif.

Meskipun penelitian ini telah mencapai beberapa keuntungan yang signifikan, penelitian ini juga menghadapi tantangan yang signifikan. Salah satunya adalah bagaimana mengelola sistem yang lebih besar secara optimal dan mengelola kompleksitas integrasi yang muncul dari hubungan antar layanan dalam arsitektur *microservices*. Oleh karena itu, pengujian lebih lanjut dan upaya pengembangan ekstensif diperlukan. Oleh karena itu, akan memastikan keberlangsungan dan kinerja yang optimal dari sistem pengelolaan pembelajaran daring ini, sekaligus menjawab dinamika perubahan di bidang pendidikan digital yang terus berkembang. .

4. DISKUSI

Penelitian ini telah menggambarkan pendekatan yang inovatif dalam mengatasi tantangan pengembangan Learning Management System (LMS) dengan menerapkan konsep Microgen, pendekatan *Microservices*, dan GraphQL Generator. Hasil dari penelitian ini menunjukkan bahwa pendekatan yang menggabungkan Microgen, *Microservices*, dan GraphQL Generator memiliki potensi besar dalam mengembangkan LMS yang responsif, efisien, dan adaptif. Penerapan Microgen dalam merancang skema database menggunakan MgSchema membantu pengembang merencanakan struktur data dengan lebih baik, memberikan visualisasi yang jelas, dan memungkinkan adaptasi yang cepat sesuai persyaratan sistem.

Adopsi fungsi Microgen dalam manipulasi data menghasilkan kemampuan untuk dengan cepat dan

konsisten membuat fungsi pengikatan data. Ini memungkinkan pengembang untuk fokus pada aspek inti pembangunan tanpa harus menghabiskan waktu berlebihan untuk implementasi detail. Penerapan pendekatan *microservices* memungkinkan sistem LMS Pradita terbagi menjadi layanan mikro independen, yang menguntungkan pemeliharaan dan skalabilitas.

Jika menganalisa dari penelitian terdahulu dari Muhammad Fathan Radhiyan dengan judul "Analysis and Design of Microservices Architecture with GraphQL as an API Gateway for Higher Education Information System" yang memiliki fokus pada arsitektur *microservice* dengan GraphQL sebagai gerbang API dalam konteks sistem informasi pendidikan tinggi, menitikberatkan pada desain arsitektur terdistribusi yang memungkinkan integrasi layanan mikro untuk mengelola informasi perguruan tinggi secara efektif, sedangkan dalam penelitian ini fokus pada konstruksi LMS yang adaptif dengan pendekatan Microgen dan GraphQL Generator, menitikberatkan pada antarmuka interaktif dan fleksibilitas pengembangan.

Kemudian dari penelitian yang dilakukan Rizky Faizal Amin dengan judul "Implementasi Restfull Api Menggunakan Arsitektur Microservice Untuk Manajemen Tugas Kuliah" yang lebih menekankan implementasi RESTful API dengan arsitektur mikroserwis untuk manajemen tugas kuliah, menyoroti fleksibilitas dalam mengelola tugas akademik melalui layanan terpisah yang terhubung melalui API.

Dan terakhir penelitian dari Catur Adi Pamungkas dengan judul "Rancang Bangun Learning Management System Berbasis Code Igniter Menggunakan Metode Prototype" berfokus pada pengembangan Learning Management System dengan menggunakan framework Code Igniter dan pendekatan metode prototype, mengeksplorasi langkah-langkah iteratif dalam merancang sistem pembelajaran yang lebih tradisional berbasis web.

Beberapa penelitian sebelumnya mungkin telah fokus pada aspek pengembangan tertentu, seperti penerapan *Microservices* atau GraphQL Generator, tanpa melibatkan aspek lain seperti Microgen. Oleh karena itu, penelitian ini dapat dilihat sebagai kontribusi yang unik dengan pendekatan yang holistik dalam mengatasi kompleksitas pengembangan LMS.

Selain itu, penelitian ini dapat menjadi inspirasi bagi penelitian-penelitian selanjutnya yang ingin menjelajahi potensi dari konsep Microgen dalam pengembangan sistem pendidikan lainnya. Dengan terus menggali dan menerapkan metode-metode inovatif seperti ini, kita dapat terus meningkatkan kualitas sistem pendidikan dan membantu masyarakat dalam menghadapi perubahan pendidikan global yang terus berkembang.

5. KESIMPULAN

Studi ini memberikan wawasan tentang penerapan Microgen dalam membangun Learning Management System (LMS) dengan pendekatan *Microservices* dan GraphQL Generator. Selama eksplorasi kami, kami berhasil mengintegrasikan konsep Microgen untuk merancang database skematis menggunakan gambar MgSchema. Gambar ini menunjukkan bahwa visualisasi skema dapat membantu developer merencanakan struktur data dengan lebih efisien. Menggunakan fungsi Microgen membantu kami mengimplementasikan fungsi yang diperlukan untuk memanipulasi data, memungkinkan adaptasi yang cepat dan konsisten terhadap persyaratan sistem.

Selain itu, penerapan pendekatan *Microservices* di LMS Pradita membuktikan bahwa pendekatan ini memiliki keunggulan nyata dalam mengembangkan sistem yang terukur dan dapat dikelola. Setiap fitur diimplementasikan sebagai layanan mikro mandiri yang memberikan fleksibilitas tinggi selama pengembangan, pengujian, dan pemeliharaan. Kami juga mengamati bagaimana Pembuat GraphQL mempercepat pengembangan antarmuka, dengan menyediakan cara interaktif untuk berinteraksi dengan API yang ada.

Sepanjang penelitian ini, telah ditunjukkan bahwa konsep Microgen dapat berhasil diintegrasikan ke dalam konstruksi LMS yang responsif, efisien, dan berkualitas. Kami percaya bahwa mengadopsi Microgen, pendekatan *Microservices* dan generator GraphQL di LMS Pradita merupakan langkah maju dalam inovasi teknologi pendidikan. Temuan kami menggambarkan potensi besar untuk memecahkan masalah pengembangan sistem yang kompleks dan memberikan dasar untuk solusi yang lebih baik untuk memenuhi perubahan kebutuhan pendidikan modern. Dalam proses pengembangan sistem informasi yang semakin kompleks, seperti LMS, penerapan metode yang canggih dan mutakhir menjadi semakin penting. Hasil dan pengalaman yang kami peroleh dari penelitian ini memberikan panduan berharga bagi pengembang, akademisi, dan praktisi di bidang teknologi pendidikan. Dengan terus mengeksplorasi dan menerapkan metode inovatif seperti Microgen, kami mampu memberikan solusi yang lebih baik dan lebih adaptif untuk memenuhi perubahan kebutuhan pendidikan.

• DAFTAR PUSTAKA

- [1] E. Sukmawati, H. Fitriadi, Y. Pradana, H. Trustisari, and P. A. Wijayanto, *Digitalisasi Sebagai Pengembangan Model Pembelajaran*. Cendikia Mulia Mandiri, 2022.
- [2] N. Hidayat, D. Iskandar, L. Afuan, and ..., "Analysis and Design of Data warehouse Based on SNIKTI Using Data warehouse

- Life Cycle Method at UNSOED Engineering Faculty,” *J. Tek. ...*, vol. 3, no. 3, pp. 797–805, 2022, [Online]. Available: <http://jutif.if.unsoed.ac.id/index.php/jurnal/article/view/514%0Ahttp://jutif.if.unsoed.ac.id/index.php/jurnal/article/download/514/137>
- [3] R. W. SABILAR, “Pemanfaatan Media Online Dalam Peningkatan Pembelajaran Pai Masa Pandemi Covid-19 Di Smp Ma’arif Nu 1 Ajibarang Banyumas,” p. 127, 2022, [Online]. Available: <http://repository.iainpurwokerto.ac.id/id/eprint/13367>
- [4] E. Irawan *et al.*, *Pendidikan Tinggi Di Masa Pandemi: Transformasi, Adaptasi, dan Metamorfosis Menyongsong New Normal*. Zahir Publishing, 2020. [Online]. Available: https://books.google.co.id/books?id=_Kr7DwAAQBAJ
- [5] A. V. Junilla, “Perancangan Ui/Ux Microservice Sistem Informasi Akademik Kampus Dengan Metode Perancangan Five Planes,” *Front. Neurosci.*, vol. 14, no. 1, pp. 1–13, 2021.
- [6] M. F. Radhiyan, D. Khairani, and H. B. Suseno, “Analysis and Design of Microservices Architecture with GraphQL as an API Gateway for Higher Education Information System,” *2022 Int. Conf. Sci. Technol. ICOSTECH 2022*, 2022, doi: 10.1109/ICOSTECH54296.2022.9829090.
- [7] D. Sugiarto, I. Mardianto, M. Najih, D. Adrian, and D. A. Pratama, “Perancangan Dashboard Untuk Visualisasi Harga Dan Pasokan Beras Di Pasar Induk Beras Cipinang,” *J. Teknol. Ind. Pertan.*, vol. 31, no. 1, pp. 12–19, 2021, doi: 10.24961/j.tek.ind.pert.2021.31.1.12.
- [8] M. H. Dehghani, S. Kolahdouz-Rahimi, M. Tisi, and D. Tamzalit, “Facilitating the migration to the microservice architecture via model-driven reverse engineering and reinforcement learning,” *Softw. Syst. Model.*, vol. 21, no. 3, pp. 1115–1133, 2022, doi: 10.1007/s10270-022-00977-3.
- [9] Nasution, “Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online,” *Informatics Eng.*, pp. 1–160, 2021, [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/38607>
- [10] R. P. I. Putri and M. Kamisutara, “Perencanaan Pengembangan Sistem Informasi Pemerintahan Daerah (SIPD) dengan Konsep Service Oriented Architecture (SOA),” *J. Ilmu Komput. dan Bisnis*, vol. 13, no. 1, pp. 174–181, 2022, doi: 10.47927/jikb.v13i1.295.
- [11] A. Tanuwijaya, H. N. Palit, and A. Noertjahyana, “Penerapan Microservices dan Amazon Elastic Container Service untuk Mendukung Scalability,” *J. Infra*, 2021, [Online]. Available: <http://publication.petra.ac.id/index.php/teknik-informatika/article/view/11451%0Ahttps://publication.petra.ac.id/index.php/teknik-informatika/article/viewFile/11451/10061>
- [12] H. Adkins, B. Beyer, P. Blankinship, P. Lewandowski, A. Oprea, and A. Stubblefield, *Building Secure and Reliable Systems - Best Practices for Designing, Implementing and Maintaining Systems*. O’Reilly Media, 2020. [Online]. Available: <https://sre.google/books/>
- [13] N. Wicaksono, S. Sarwosri, and D. Sunaryono, “Rancang Bangun Web Kustom Menggunakan Open Graph Protokol pada Studi Kasus Web Jurusan Teknik Informatika,” *J. Tek. ITS*, vol. 7, no. 1, 2018, [Online]. Available: <http://ejurnal.its.ac.id/index.php/teknik/article/view/29052>
- [14] A. Belhadi, M. Zhang, and A. Arcuri, *Evolutionary-based automated testing for GraphQL APIs*, vol. 1, no. 1. Association for Computing Machinery, 2022. doi: 10.1145/3520304.3528952.
- [15] M. Santoro, L. Vaccari, D. Mavridis, R. Smith, M. Posada, and D. Gattwinkel, *Web application programming interfaces (apis): General purpose standards, terms and European commission initiatives*. 2019. doi: 10.2760/675.
- [16] Pasha Bhimasty and Endah Sudarmilah, “Donation Management Information System in Baitul Maal Mitra Usaha Mandiri Using the Waterfall Method,” *J. Tek. Inform.*, vol. 3, no. 5, pp. 1195–1202, 2022, doi: 10.20884/1.jutif.2022.3.5.291.
- [17] N. Azizah and N. A. Widiastuti, “Implementasi Sistem Informasi Geografis Pencarian Lokasi Umkm Di Kabupaten Jepara,” *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, 2019, doi: 10.36499/jinrpl.v1i1.2768.
- [18] I. KUSUMAWATI, “PERANCANGAN SISTEM INFORMASI PENGAJUAN COLLO APPLICATION FORM (CAF) PADA TOWER BERSAMA GUNA MENINGKATKAN PELAYANAN KEPADA OPERATOR,” *Pap. Knowl. Towar. a Media Hist. Doc.*, no. 2017, pp. 9–29, 2018.
- [19] D. Novita, A. Farisi, and F. P. Sihotang, “Pembuatan Aplikasi Mobile Tanpa Coding Bagi Karyawan PT Dunia Kimia Utama,” *Fordicate*, vol. 2, no. 1, pp. 29–39, 2022, doi: 10.35957/fordicate.v2i1.3460.
- [20] M. P. Mukhid, “DISAIN TEKNOLOGI DAN

- INOVASI PEMBELAJARAN DALAM BUDAYA ORGANISASI DI LEMBAGA PENDIDIKAN.” Pustaka Egaliter. Com, 2023.
- [21] H. Risnanto, “Rancang Bangun Sistem Informasi Layanan Mandiri Perpustakaan Berbasis Arsitektur Microservice,” *Repository.Uinjkt.Ac.Id*, 2022, [Online]. Available: [https://repository.uinjkt.ac.id/dspace/handle/123456789/65036%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/65036/1/HERI RISNANTO-FST.pdf](https://repository.uinjkt.ac.id/dspace/handle/123456789/65036%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/65036/1/HERI%20RISNANTO-FST.pdf)
- [22] N. Siagian, T. E. Tamba, H. H. O. Situmorang, and H. Samosir, “Aplikasi Apotek Berbasis Web Menggunakan Arsitektur Microservices (Studi Kasus Apotek Glen, Kab.Toba),” *J. Appl. Technol. Informatics Indones.*, vol. 1, no. 2, pp. 22–28, 2021, doi: 10.54074/jati.v1i2.35.
- [23] L. Khoirunnisa, “Rancang Bangun Sistem E-Learning Berbasis Microservices dan Domain Driven Design (Studi Kasus Probistek UIN Maulana Malik Ibrahim Malang),” 2019.
- [24] E. Nikolas and N. Santoso, “Pengembangan Aplikasi Forum Komunitas Sepeda berbasis Web,” *J. Pengemb. Teknol. Inf. ...*, vol. 5, no. 9, 2021, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/9803%0Ahttp://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/9803/4388>
- [25] S. Karlsson, A. Causevic, and D. Sundmark, “Automatic Property-based Testing of GraphQL APIs,” *Proc. - 2021 IEEE/ACM Int. Conf. Autom. Softw. Test, AST 2021*, pp. 1–10, 2021, doi: 10.1109/AST52587.2021.00009.
- [26] S. Yangui, I. B. Rodriguez, K. Drira, and Z. Tari, *Service-Oriented Computing: 17th International Conference, ICSOC 2019, Toulouse, France, October 28–31, 2019, Proceedings.* in Lecture Notes in Computer Science. Springer International Publishing, 2019. [Online]. Available: <https://books.google.co.id/books?id=fDy5DwAAQBAJ>

