

# Implementasi Arsitektur Microservices pada Sistem Berbasis Web Menggunakan Metode Scrum

Rahmat Gunawan<sup>1</sup>, Zaenal Alfian<sup>2</sup>, Darmansyah<sup>3</sup>

<sup>1</sup> Program Studi Sistem Informasi- STMIK Rosma

<sup>2,3</sup> Program Studi Teknik Informatika-STMIK Rosma

Jl. Parahyangan Adiarsa Barat – Karawang Jawa Barat - Indonesia

[rahmat@rosma.ac.id](mailto:rahmat@rosma.ac.id), [zaenal.alfian@alumni.rosma.ac.id](mailto:zaenal.alfian@alumni.rosma.ac.id), [darmansyah@rosma.ac.id](mailto:darmansyah@rosma.ac.id)

**Abstrak**— Pengembangan sistem informasi manajemen magang sering kali terhambat oleh keterbatasan arsitektur monolitik tradisional yang kaku, sulit diskalakan, dan berisiko mengalami kegagalan sistem terpusat. Masalah utama yang ditemukan pada sistem manual atau monolitik sebelumnya adalah inefisiensi koordinasi yang menyebabkan data tersebar di berbagai dokumen terpisah, serta penurunan performa signifikan saat beban pengguna meningkat di mana waktu respons dapat melonjak hingga 170 detik pada operasi kritis. Penelitian ini mengusulkan solusi melalui penerapan arsitektur Microservices menggunakan metodologi Scrum untuk meningkatkan fleksibilitas dan ketahanan sistem. Platform dirancang dalam tiga lapisan: presentation layer berbasis React.js, logic layer berbasis Hapi.js dengan proteksi JWT, dan data layer menggunakan PostgreSQL. Metode Scrum digunakan untuk memastikan iterasi pengembangan yang adaptif melalui siklus sprint. Hasil implementasi menunjukkan bahwa arsitektur Microservices mampu meningkatkan efisiensi operasional dan skalabilitas secara signifikan. Pengujian teknis membuktikan stabilitas sistem dengan tingkat keberhasilan fungsionalitas 100% dan waktu respons yang stabil pada kisaran 399 ms meskipun di bawah beban pengguna yang tinggi. Dengan pembagian layanan yang independen, sistem ini menawarkan isolasi kesalahan yang lebih baik dibandingkan pendekatan monolitik, sehingga sangat layak diadopsi untuk mendukung digitalisasi proses manajemen magang di lingkungan pendidikan vokasi dan industri.

**Kata Kunci**— Microservices, Application Programming Interface, Magang, React.js, Hapi.js.

**Abstract**— The development of internship management information systems is often hindered by the limitations of traditional monolithic architectures, which are rigid, difficult to scale, and prone to centralized failures. The primary issues identified in previous manual or monolithic systems include coordination inefficiencies causing scattered data across separate documents and significant performance degradation during peak user loads, where response times could spike to 170 seconds for critical operations. This research proposes a solution through the implementation of Microservices architecture using the Scrum methodology to enhance system flexibility and resilience. The platform is designed in three layers: a React.js-based presentation layer, a Hapi.js-based logic layer with JWT protection, and a PostgreSQL data layer. The Scrum method is applied to ensure adaptive development iterations through sprint cycles. Implementation results demonstrate that Microservices architecture significantly improves operational efficiency and

*scalability. Technical testing confirms system stability with a 100% functional success rate and a stable response time of approximately 399 ms, even under high user load. By utilizing independent service division, this system offers superior fault isolation compared to monolithic approaches, making it highly suitable for supporting the digitalization of internship management processes in vocational education and industrial environments.*

**Keywords**— Microservices, Application Programming Interface, Internship, React.js, Hapi.js.

## I. PENDAHULUAN

Di era transformasi digital, koordinasi program magang menjadi jembatan krusial antara dunia pendidikan dan praktik nyata di industri. Pengelolaan magang menuntut sinkronisasi yang efektif antara peserta, mentor, dan perusahaan penerima guna memfasilitasi seluruh siklus kegiatan, mulai dari proses seleksi, pendaftaran, pemantauan harian, hingga penilaian akhir dan penerbitan sertifikat. Kebutuhan akan platform terintegrasi yang mampu menyediakan data secara real-time dan terukur menjadi faktor penentu keberhasilan tujuan pembelajaran magang dalam memenuhi standar kompetensi industri [1].

Namun, realitas di lapangan menunjukkan bahwa banyak institusi masih mengandalkan proses manual atau arsitektur monolitik tradisional yang memiliki keterbatasan dalam skalabilitas dan risiko kegagalan sistem terpusat. Sistem monolitik [2], [3] sering kali mengalami kendala teknis berupa lonjakan waktu respons yang bisa mencapai 170 detik pada operasi kritis saat menghadapi beban pengguna tinggi, yang berujung pada kegagalan fungsional. Selain itu, pengelolaan data yang tersebar di berbagai dokumen terpisah meningkatkan beban administratif dan memperlambat proses pengambilan keputusan bagi sekolah maupun perusahaan mitra.

Berdasarkan *State of the Art (SOTA)* dalam literatur rekayasa perangkat lunak, penelitian sebelumnya mengenai sistem informasi magang umumnya masih berfokus pada model monolitik yang sulit untuk dikembangkan secara independent [4]. Justifikasi pemilihan arsitektur *microservices* dalam penelitian ini didasarkan pada kemampuannya untuk melakukan isolasi kesalahan (*fault isolation*), di mana kegagalan pada satu layanan tidak akan meruntuhkan keseluruhan sistem. Secara empiris, pendekatan *microservices*

terbukti jauh lebih efisien dengan mampu mempertahankan stabilitas waktu respons pada kisaran 399 ms untuk operasi krusial, memberikan keunggulan kompetitif dibandingkan arsitektur tradisional.

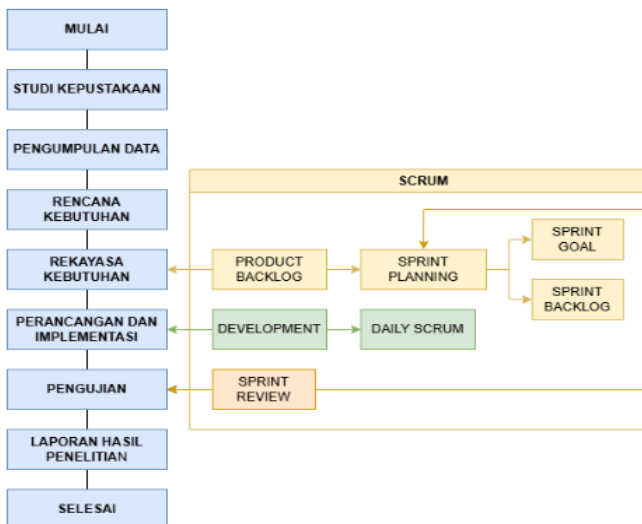
Atas dasar tantangan tersebut, penelitian ini mengembangkan aplikasi manajemen magang berbasis web yang mengintegrasikan arsitektur *microservices* dengan metodologi Scrum untuk menjamin proses pengembangan yang adaptif dan iteratif. Solusi yang diusulkan memanfaatkan React.js pada presentation layer untuk antarmuka yang responsif, Hapi.js sebagai kerangka kerja *microservices* yang modular dengan proteksi JWT untuk keamanan, serta PostgreSQL yang diisolasi per layanan guna menjamin integritas data. Implementasi sistem ini diharapkan dapat menyederhanakan alur administratif, meningkatkan ketahanan sistem terhadap lonjakan beban, serta memberikan pengalaman pengguna yang optimal bagi seluruh pemangku kepentingan.

## II. METODE PENELITIAN

Metodologi Penelitian ini menjelaskan kerangka kerja serta langkah-langkah sistematis yang digunakan dalam pengembangan dan pengujian aplikasi magang berbasis web untuk siswa Sekolah Menengah Kejuruan (SMK).

### 2.1. Tahapan Penelitian

Penelitian ini menggunakan desain eksperimental [5] yang didasarkan pada uji kelayakan untuk menentukan sejauh mana aplikasi magang siswa SMK yang dibangun dengan arsitektur *microservices* layak digunakan. Pendekatan ini difokuskan pada pembuktian kelayakan aplikasi dari aspek fungsionalitas, kinerja, kegunaan, dan keamanan dasar, tanpa menekankan pada perbandingan statistik langsung dengan aplikasi monolitik [6]. Dengan menerapkan eksperimen terkontrol, peneliti bertujuan untuk memastikan bahwa solusi *microservices* dapat diimplementasikan secara praktis serta mampu memenuhi kebutuhan pengguna di lingkungan pendidikan maupun industri.



Gbr 1. Alur Penelitian

### 2.2. Metodologi Pengembangan Sistem

Pengembangan aplikasi ini mengadopsi kerangka kerja Scrum, yang merupakan salah satu metodologi agile [6], [7]. Scrum dipilih untuk memungkinkan proses pengembangan yang fleksibel, iteratif, dan adaptif terhadap perubahan. Alur kerja diorganisasikan dalam siklus singkat yang disebut sprint.

Kerangka kerja Scrum dalam penelitian ini melibatkan tiga peran utama:

- **Product Owner:** Bertanggung jawab mendefinisikan visi produk, mengelola, dan memprioritaskan product backlog.
- **Scrum Master:** Berperan sebagai fasilitator untuk memastikan tim mematuhi praktik Scrum serta membantu menyelesaikan hambatan yang muncul.
- **Development Team:** Bertugas mengembangkan, menguji, dan menghasilkan increment produk yang fungsional pada akhir setiap sprint.
- **Manajemen proyek** dilakukan menggunakan artefak Scrum seperti product backlog (daftar kebutuhan pengguna) dan sprint backlog (tugas yang harus diselesaikan dalam satu sprint).

### 2.3. Teknik Pengumpulan Data

Pengumpulan data dalam penelitian ini difokuskan pada data sekunder untuk merumuskan kebutuhan sistem. Teknik yang digunakan meliputi:

- **Studi Literatur:** Melakukan kajian mendalam terhadap jurnal ilmiah, buku, dan repositori daring (seperti Google Scholar dan Mendeley) untuk memperoleh landasan teori mengenai arsitektur *microservices*, metodologi Scrum, serta teknik pengujian aplikasi.
- **Analisis Dokumen:** Menelaah dokumen Standard Operating Procedure (SOP) dan panduan terkait program magang di SMK. Data dari dokumen ini digunakan langsung untuk memetakan kebutuhan fungsional dan menyusun product backlog tanpa melakukan survei atau wawancara dengan pengguna [8].

### 2.4. Teknik Analisis dan Perancangan Sistem

Tahap analisis dan perancangan sistem menggunakan pemodelan visual untuk memastikan bahwa kebutuhan fungsional dan arsitektur sistem terdefinisi dengan baik [7], [8].

### 2.5. Rekayasa Kebutuhan (Requirements Engineering)

Untuk memodelkan kebutuhan sistem digunakan Unified Modeling Language (UML) [9], yang terdiri dari:

- **Use Case Diagram:** Untuk memetakan interaksi antara aktor (admin, siswa, pembimbing, perusahaan) dengan fungsionalitas utama sistem.
- **Activity Diagram:** Untuk menggambarkan alur kerja dinamis dari proses bisnis utama, seperti alur pendaftaran magang.

- *Sequence Diagram*: Untuk merinci urutan interaksi dan komunikasi antar komponen dalam arsitektur microservices.
- *Class Diagram*: Untuk mendefinisikan struktur data, atribut, serta hubungan antar entitas dalam sistem, yang menjadi cetak biru perancangan basis data.

**2.6. Desain Arsitektur**

Aplikasi dirancang menggunakan arsitektur microservices untuk memastikan skalabilitas dan kemudahan pemeliharaan. Setiap fungsionalitas bisnis dipecah menjadi layanan independen, seperti *Auth Service, Student Service, Company Service*, dan lainnya, yang berkomunikasi melalui API Gateway. Setiap layanan memiliki basis data PostgreSQL tersendiri untuk mendukung independensi data.

**2.7. Teknik Pengujian Sistem**

Untuk memastikan kualitas dan kelayakan aplikasi [9], [10], [11], [12], dilakukan dua jenis pengujian secara komprehensif:

- *Black-Box Testing* : Pengujian ini dilakukan untuk mengevaluasi fungsionalitas sistem dari perspektif pengguna tanpa melihat struktur kode internal. Metode yang digunakan meliputi:
- *Functional Testing*: Memverifikasi setiap endpoint REST API menggunakan Postman untuk memastikan kesesuaian respons HTTP dan data yang dikirim.
- *Usability Testing*: Melibatkan 14 partisipan untuk menguji alur kerja dan antarmuka pengguna, memastikan navigasi dan interaksi berjalan sesuai harapan.

TABEL I.  
PRODUCT BACKLOG

No	Peran	User Story	Prioritas
1	Admin	Sebagai Admin, saya ingin dapat masuk ( <i>login</i> ) ke sistem sehingga saya dapat mengakses <i>dashboard</i> admin	Tinggi
2	Admin	Sebagai Admin, saya ingin dapat melihat semua data pengguna (admin, siswa, perusahaan, pembimbing) sehingga saya dapat memantau seluruh pengguna sistem	Tinggi
3	Admin	Sebagai Admin, saya ingin dapat menambah, mengubah, dan menghapus data admin sehingga saya dapat mengelola akses admin	Tinggi
4	Admin	Sebagai Admin, saya ingin dapat menambah, mengubah, dan menghapus data perusahaan sehingga saya dapat mengelola perusahaan mitra	Tinggi
5	Admin	Sebagai Admin, saya ingin dapat menambah, mengubah, dan menghapus data siswa sehingga saya dapat mengelola peserta magang	Tinggi
6	Admin	Sebagai Admin, saya ingin dapat menambah, mengubah, dan menghapus data pembimbing sehingga saya dapat mengelola pembimbing	Tinggi
7	Admin	Sebagai Admin, saya ingin dapat melihat semua program magang sehingga saya dapat memantau program yang tersedia	Sedang

III. HASIL DAN PEMBAHASAN

**3.1 Product Backlog**

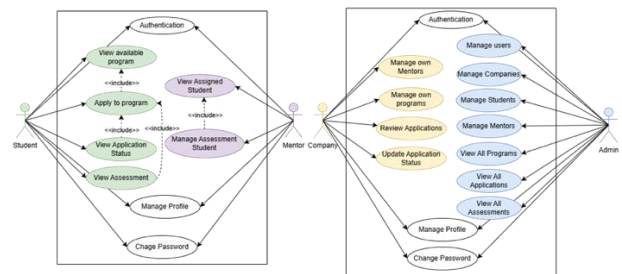
Pada subbagian ini, *Product Backlog* disajikan sebagai kumpulan *user stories* yang menggambarkan kebutuhan fungsional dari sistem magang SMK (*SMK Internship System*) berdasarkan sudut pandang empat aktor utama (Admin, Perusahaan, Pembimbing, dan Siswa). Setiap *user story* diformulasikan menggunakan format: "*Sebagai <peran>, saya ingin <fitur> sehingga saya dapat <tujuan>*" dan diklasifikasikan berdasarkan tingkat prioritas (Tinggi, Sedang) sesuai dengan urgensi serta dampaknya terhadap keberhasilan implementasi sistem.

*Product Backlog* ini menjadi dasar utama dalam proses *Sprint Planning* untuk menentukan fitur mana yang akan dikembangkan pada setiap iterasi pengembangan, sehingga kebutuhan pengguna dapat dipenuhi secara bertahap dan sistematis sesuai dengan prinsip pengembangan *agile*.

**Penjelasan Prioritas:**

- **Tinggi (High)**: Fitur kritis yang wajib ada agar sistem dapat berfungsi.
- **Sedang (Medium)**: Fitur penting untuk meningkatkan pengalaman pengguna.
- **Rendah (Low)**: Fitur tambahan untuk mempermudah penggunaan (tidak ditampilkan pada tabel ini).

**3.2 Use Case Diagram**



Gbr 2. Rancangan Use Case Diagram

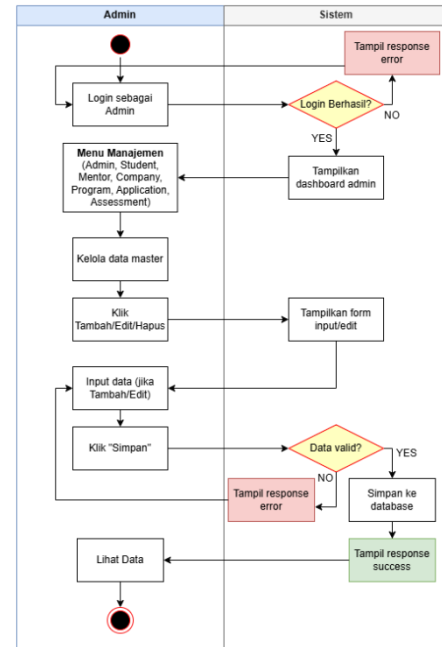
TABEL II  
DESKRIPSI USE CASE

Aktor	Use Case Utama
<b>Siswa (Student)</b>	<ol style="list-style-type: none"> <li>1. Autentikasi – Login/logout untuk siswa yang terdaftar.</li> <li>2. Dashboard – Akses ringkasan data/menu cepat.</li> <li>3. Jelajahi Program – Melihat peluang magang yang tersedia beserta persyaratan dan deskripsi.</li> <li>4. Pengajuan Aplikasi – Mendaftar ke program yang dipilih dengan unggah dokumen dan validasi.</li> <li>5. Pelacakan Status – Memantau status aplikasi (dalam proses/diterima/ditolak).</li> <li>6. Akses Hasil Evaluasi – Melihat nilai akhir dan umpan balik pembimbing setelah magang.</li> <li>7. Manajemen Profil – Memperbarui data pribadi (nama, kontak, foto).</li> <li>8. Ubah Kata Sandi – Memperbarui kredensial login untuk keamanan akun.</li> </ol>
<b>Pembimbing (Mentor)</b>	<ol style="list-style-type: none"> <li>1. Autentikasi – Login dengan hak akses pembimbing.</li> <li>2. Dashboard – Akses ringkasan data/menu cepat.</li> <li>3. Daftar Siswa Binaan – Melihat daftar siswa di bawah bimbingan.</li> <li>4. Evaluasi Siswa – Memberikan nilai, umpan balik, dan status kelulusan.</li> <li>5. Manajemen Profil – Memperbarui informasi pribadi (jabatan, departemen, kontak).</li> </ol>
<b>Perusahaan (Company)</b>	<ol style="list-style-type: none"> <li>1. Autentikasi – Login sebagai perwakilan perusahaan.</li> <li>2. Dashboard – Akses ringkasan data/menu cepat.</li> <li>3. Manajemen Pembimbing – Menambah, mengubah, atau menghapus data pembimbing terafiliasi.</li> <li>4. Manajemen Program – Menentukan, memperbarui, atau menutup lowongan magang.</li> <li>5. Peninjauan Aplikasi – Meninjau aplikasi siswa dan memutuskan (diterima/ditolak).</li> <li>6. Pembaruan Status Aplikasi – Mengubah status dan memberi umpan balik kepada siswa.</li> <li>7. Manajemen Profil – Memperbarui detail perusahaan.</li> </ol>
<b>Admin</b>	<ol style="list-style-type: none"> <li>1. Autentikasi – Login sebagai administrator sistem.</li> <li>2. Dashboard – Akses ringkasan data/menu cepat.</li> <li>3. Manajemen Pengguna – CRUD akun untuk Admin, Perusahaan, Pembimbing, dan Siswa.</li> <li>4. Manajemen Perusahaan – CRUD data perusahaan mitra.</li> <li>5. Manajemen Siswa &amp; Pembimbing – CRUD data siswa dan pembimbing.</li> <li>6. Manajemen Program – CRUD seluruh program magang.</li> </ol>

7. Monitoring Aplikasi & Evaluasi – Melihat aplikasi dan hasil evaluasi.
8. Ubah Kata Sandi – Memperbarui keamanan akun admin.

### 3.3 Activity Diagram

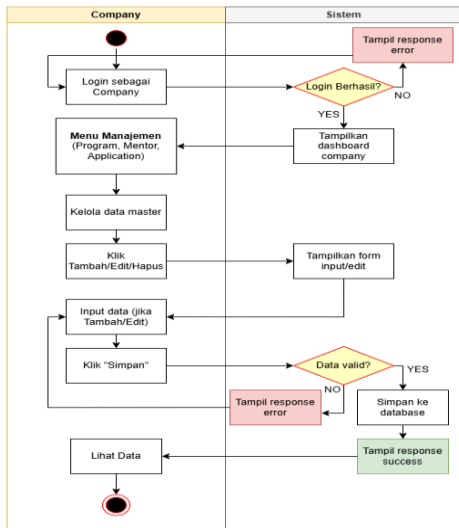
#### 1. Alur Admin Operations



Gbr 3. Admin Activity Diagram

Activity Diagram Admin menggambarkan alur kerja seorang admin dalam mengelola data master pada sistem magang. Proses dimulai ketika admin melakukan login ke dalam sistem. Setelah proses autentikasi berhasil, sistem akan menampilkan dashboard yang menyediakan menu pengelolaan berbagai entitas, seperti admin, siswa, mentor, perusahaan, program, pendaftaran, dan penilaian.

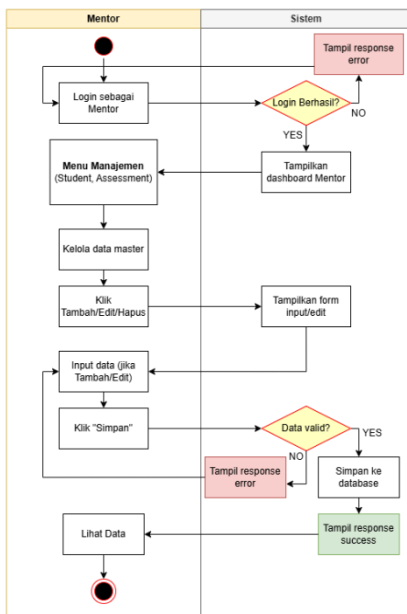
#### 2. Alur Company Manage Programs & Mentors



Gbr 4. Company Activity Diagram

Activity Diagram ini menggambarkan proses bisnis yang dijalankan oleh pengguna dengan peran sebagai perusahaan dalam sistem magang. Proses dimulai dengan autentikasi, di mana pihak perusahaan melakukan login. Jika autentikasi berhasil, sistem akan menampilkan dashboard khusus perusahaan yang menyediakan menu pengelolaan entitas program, mentor, dan pendaftaran magang.

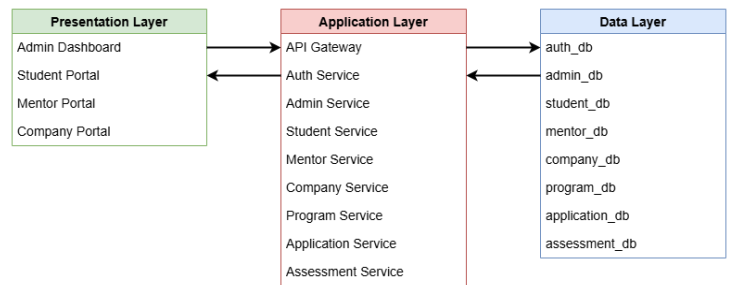
3. Activity Diagram Management Assesment



Gbr 5. Activity Diagram Management Assesment Diagram aktivitas ini menggambarkan alur kegiatan yang dilakukan oleh pengguna dengan peran sebagai mentor dalam sistem magang. Proses dimulai pada tahap login, di mana mentor harus melakukan autentikasi untuk dapat mengakses sistem. Setelah berhasil login, sistem akan menampilkan dashboard

khusus mentor yang menyediakan menu pengelolaan entitas siswa dan penilaian.

3.4 Implementasi Arsitektur



Gbr 6. Implementasi Arsitektur

Pada Presentation Layer, seluruh interaksi pengguna dengan sistem difasilitasi melalui antarmuka web dan API gateway, yang berfungsi sebagai titik masuk permintaan klien ke layanan backend.

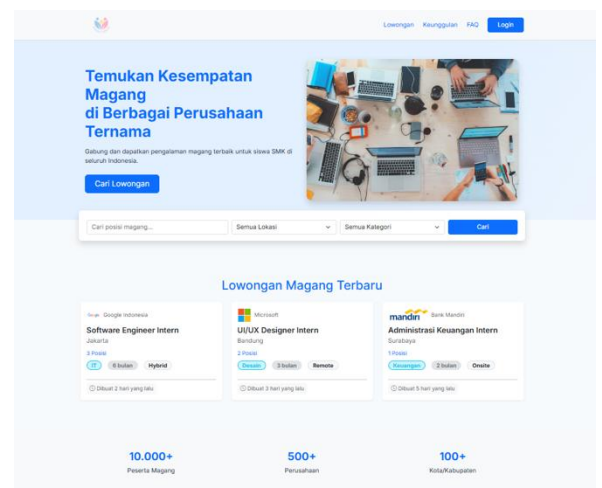
Application Layer berperan sebagai inti logika bisnis, di mana setiap microservice—seperti layanan autentikasi, manajemen pengguna, program magang, pendaftaran, dan penilaian—berjalan secara independen dan saling berkomunikasi melalui API. Lapisan ini memastikan bahwa pemrosesan data dan aturan bisnis dijalankan secara terpisah sesuai dengan domain masing-masing layanan.

Sementara itu, Data Layer bertanggung jawab atas pengelolaan dan penyimpanan data, di mana setiap microservice dapat memiliki basis data yang terisolasi. Hal ini mendukung skalabilitas, keamanan, dan konsistensi data.

3.5 Implementasi Sistem

Implementasi tampilan menu web aplikasi ini memanfaatkan konsep SPA pada React dan terdiri dari 1 *index.html* tetapi bisa memiliki lebih dari 1 page dan komponen.

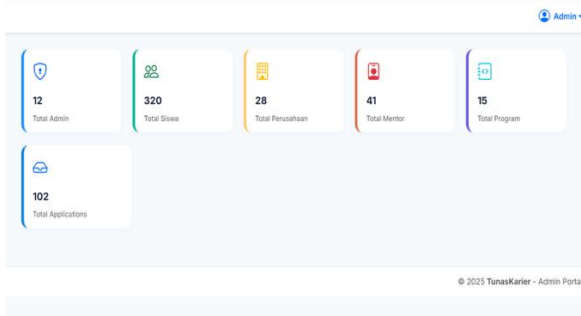
1. Tampilan HomePage



Gbr 7. Homepage

Halaman homepage pada aplikasi React.js ini dirancang sebagai antarmuka utama yang menampilkan informasi peluang magang bagi siswa SMK secara interaktif dan responsif. Melalui integrasi komponen modular, pengguna dapat mengakses fitur pencarian lowongan, melihat daftar program magang terbaru, serta memperoleh informasi statistik dan keunggulan platform secara terstruktur.

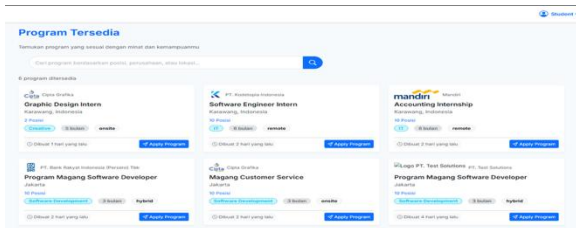
2. Tampilan *Dashboard Admin*



Gbr 8. *Dashboard Admin*

*Dashboard Admin* pada aplikasi React.js ini menampilkan rekap data dan menu penting seperti jumlah admin, siswa, perusahaan, mentor, *programs*, serta *applications*. Data di-*fetch* langsung dari API yang dikembangkan dengan Hapi.js dan PostgreSQL.

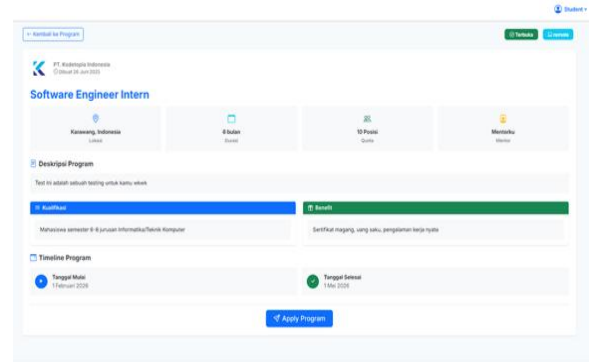
3. Tampilan *Student Portal*



Gbr 9. *Student Portal*

*Student Portal* memungkinkan siswa mencari dan mendaftar program magang melalui antarmuka React.js yang responsif. *Request* pengguna diproses oleh *backend* Hapi.js, sementara data program dan aplikasi disimpan secara terstruktur di PostgreSQL. Integrasi ketiga *layer* ini memastikan pencarian dan pendaftaran program berjalan efisien.

4. Tampilan Detail Program



Gbr 10. *Detail Apply Program*

Tampilan detail program menyediakan informasi lengkap terkait detail informasi program magang yang bisa dilamar oleh siswa. Kemudian juga program magang ini bisa terindeks oleh siswa berdasarkan statusnya apabila masih “*open*” atau dibuka. Ketika statusnya berubah menjadi *closed* atau *draft* maka otomatis siswa tidak akan bisa melihat program magang-nya sama sekali. Walaupun programnya masih tersedia atau belum dihapus oleh perusahaannya.

3.6 Pengujian

Hasil pengujian menunjukkan bahwa implementasi arsitektur microservices pada aplikasi magang SMK berbasis web dengan React.js, Hapi.js, dan PostgreSQL, telah berhasil memenuhi seluruh standar pengujian yang telah ditetapkan. Pengujian mencakup delapan modul utama dengan total 49 test case yang seluruhnya menunjukkan tingkat keberhasilan 100 persen.

Modul autentikasi dan otorisasi berhasil memverifikasi mekanisme login, validasi kredensial, serta pengaturan hak akses berbasis peran. Sistem manajemen data pengguna meliputi admin, siswa, perusahaan, dan mentor menunjukkan kemampuan operasi CRUD yang optimal dengan validasi input ketat. Modul program dan lamaran magang berfungsi dengan baik dalam menangani proses pendaftaran, seleksi, dan pengelolaan dokumen pendukung.

Aspek usability dan antarmuka pengguna memperoleh skor tinggi dalam hal kemudahan penggunaan, responsivitas, dan kepuasan pengguna. Pengujian arsitektur microservices mengonfirmasi efektivitas pemisahan layanan, isolasi database, dan komunikasi antar komponen yang stabil. Hasil pengujian ini membuktikan bahwa sistem yang dikembangkan tidak hanya memenuhi kebutuhan fungsional, tetapi juga mengadopsi prinsip-prinsip desain yang modern dan scalable untuk mendukung digitalisasi proses magang di lingkungan pendidikan vokasi.

TABEL III.  
HASIL PENGUJIAN

No	Modul Pengujian	Jumlah Test Case	Berhasil	Gagal	Tingkat Keberhasilan (%)	Status
1	Autentikasi & Otorisasi	5	5	0	100	OK
2	Manajemen Admin	5	5	0	100	OK
3	Manajemen Siswa	5	5	0	100	OK
4	Manajemen Perusahaan	5	5	0	100	OK
5	Manajemen Mentor	5	5	0	100	OK
6	Manajemen Program & Lamaran	10	10	0	100	OK
7	Usability & UI/UX	8	8	0	100	OK
8	Arsitektur <i>Microservices</i>	6	6	0	100	OK
<b>Total</b>		49	49	0		

#### IV. KESIMPULAN DAN SARAN

Penerapan arsitektur *microservices* pada aplikasi magang siswa SMK berbasis web berhasil meningkatkan efisiensi, skalabilitas, dan kemudahan pemeliharaan sistem. Aplikasi ini mampu memfasilitasi interaksi dan manajemen program magang secara terintegrasi antara admin, perusahaan, mentor, dan siswa, dengan fitur yang sesuai kebutuhan masing-masing peran. Penggunaan React.js dan Bootstrap pada frontend menghasilkan antarmuka yang responsif dan intuitif, sedangkan backend berbasis Hapi.js dengan proteksi JWT serta integrasi database PostgreSQL memastikan keamanan, kinerja, dan pengelolaan data yang andal. Pendekatan pengembangan bertahap menggunakan metode Scrum juga terbukti efektif dalam menyesuaikan aplikasi dengan kebutuhan pengguna, sehingga solusi yang dihasilkan layak diimplementasikan di lingkungan pendidikan dan industri.

#### DAFTAR PUSTAKA

- [1] A. W. Khurniawan and G. Erda, "Peluang Kerja Lulusan SMK Dalam Menghadapi Revolusi Industri 4.0 dan Bonus Demografi Tahun 2030," *Vocational Education Policy, White Paper*, vol. 1, no. 11, 2019.
- [2] M. Susanto and M. H. Hilman, "ABizLSM: Kerangka Migrasi Sistem Lawas Genting Arsitektur Monolitik ke *Microservices* untuk Perusahaan Startup dengan Dinamika Bisnis Tinggi," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 11, no. 1, 2025, doi: 10.25077/teknosi.v11i01.2025.27-36.
- [3] S. Waruwu, I. Kadek, and D. Nuryana, "Implementasi Arsitektur Monolitik Pada Rancang Bangun Sistem Informasi," *Journal of Informatics and Computer Science*, vol. 4, no. 4, 2023.
- [4] C. Salazar-Guaña and R. M. Toasa, "Proposal of a Web Application for the Efficient Management of Internship Projects in Higher Education: An Innovative Approach to Efficiently Manage an Ecosystem of Distributed Services," in *Lecture Notes in Networks and Systems*, 2025, doi: 10.1007/978-3-031-93106-2\_2.
- [5] H. L. Tong *et al.*, "A personalized mobile app for physical activity: An experimental mixed-methods study," *Digit. Health*, vol. 8, 2022, doi: 10.1177/20552076221115017.
- [6] M. Y. Ng *et al.*, "Participant Evaluation of Blockchain-Enhanced Women's Health Research Apps: Mixed Methods Experimental Study," *JMIR Mhealth Uhealth*, vol. 13, 2025, doi: 10.2196/65747.
- [7] S. Mulyani, *Metode Analisis dan Perancangan sistem*. Bandung: Abdi Sistemika, 2016.
- [8] F. H. Zulfallah and S. Hidayatuloh, "Analisis dan Perancangan Sistem Informasi Pendaftaran Magang pada Inspektorat Jendral Kementerian Pendidikan dan Kebudayaan," *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi dan Sistem Komputer*, vol. 5, no. 1, 2022, doi: 10.55886/infokom.v5i1.294.
- [9] O. Loyola-Gonzalez, "Black-box vs. White-Box: Understanding their advantages and weaknesses from a practical point of view," 2019. doi: 10.1109/ACCESS.2019.2949286.
- [10] M. S. Baghini, M. Mohammadi, and N. Norouzkhani, "Usability Testing: A Bibliometric Analysis Based on WoS Data," *Journal of Scientometric Research*, vol. 13, no. 1, 2024, doi: 10.5530/jscires.13.1.2.
- [11] E. D. Rinawiyanti, R. M. Surjani, M. Hartono, E. P. Permatasari, and A. C. V. Chrisma, "Mixed Method Usability Testing for User Experience and User Interface of AI-Based Supermarket Applications," *Journal of Applied Data Sciences*, vol. 6, no. 1, 2025, doi: 10.47738/jads.v6i1.453.
- [12] L. Giamattei, A. Guerriero, R. Pietrantonio, and S. Russo, "Automated functional and robustness testing of microservice architectures," *Journal of Systems and Software*, vol. 207, 2024, doi: 10.1016/j.jss.2023.111857.