

Arsitektur Mikroservis untuk Interoperabilitas Data: Transformasi Integrasi Layanan Smart Village ke Smart City

Esa Firmansyah^{1*}, Alif Gumelar Syah Moeslim², Rani Rahmayani³

¹School of Science and Technology, Asia e University, Selangor, Malaysia

²Informatika, Fakultas Teknologi Informasi, Universitas Sebelas April, Sumedang, Indonesia

³Manajemen, Ekonomi, Universitas ARS, Bandung, Indonesia

C70109170013@aeu.edu.my, 220660121161@student.unsap.ac.id, rani@ars.ac.id

Submit: 09-01-2026, Revisi: 19-02-2026, Diterima: 28-01-2026, Publikasi: 10-02-2026

Abstract— *Smart Village implementations often face data silo challenges due to the lack of efficient integration between village-level systems or city government systems (Smart City), limiting cross-level data synchronization and utilization. This study aims to design a flexible, scalable, and reliable Smart Village to Smart City data interoperability architecture at the architectural integration level to support population administration, licensing, and social assistance services. The proposed methodology adopts a software engineering approach based on the Software Development Life Cycle (SDLC), focusing on requirements analysis, domain-based service decomposition, and microservices architecture design grounded in modern Service-Oriented Architecture (SOA) principles. The proposed architecture integrates synchronous communication through REST APIs managed by an API Gateway and asynchronous communication through Apache Kafka as a message broker to enable real-time or near real-time data update events in distributed environments. The design results indicate that the microservices-based approach with API Gateway and Kafka enhances interoperability, reduces direct coupling between systems, and provides greater flexibility for village service development without disrupting the stability of central systems at the district or city level. Therefore, this architecture has the potential to serve as an adaptive integration framework for the digital public service transformation from Smart Village to Smart City.*

Keywords: *API Gateway, Interoperabilitas Data, Microservice, Smart City, Smart Village.*

I. PENDAHULUAN

Transformasi digital di tingkat desa merupakan salah satu langkah strategis dalam mendukung pengembangan ekosistem *Smart City*[1-4]. Implementasi konsep *Smart*

Village mendorong pemanfaatan teknologi informasi untuk meningkatkan kualitas layanan publik desa, seperti layanan kependudukan, perizinan, dan bantuan sosial [5-8]. Namun, dalam praktiknya, sistem informasi yang dikembangkan di tingkat desa sering kali berdiri sendiri dan tidak terintegrasi dengan sistem pemerintah kabupaten atau kota sebagai pusat pengelolaan data *Smart City* [9]. Kondisi ini menyebabkan terjadinya terpisahnya data (*silo data*), di mana data desa sulit diakses, disinkronkan, dan dimanfaatkan secara optimal oleh sistem di tingkat daerah [10]. Interoperabilitas data dalam konteks ini merujuk pada kemampuan berbagai sistem informasi yang berbeda untuk saling bertukar data, menafsirkan makna data yang dipertukarkan secara konsisten, serta memanfaatkan data tersebut secara efektif tanpa memerlukan proses integrasi ulang yang kompleks. Berbagai pendekatan telah digunakan untuk mengintegrasikan sistem informasi antar instansi pemerintahan, di antaranya adalah arsitektur monolitik terpusat, integrasi berbasis basis data, serta pendekatan *Service-Oriented Architecture* (SOA) [11]. Pendekatan tersebut umumnya mengandalkan mekanisme komunikasi sinkron melalui protokol tertentu untuk melakukan pertukaran data antar sistem. Selain itu, beberapa sistem juga memanfaatkan layanan web berbasis REST untuk menyediakan akses data antar aplikasi [12]. Pendekatan integrasi konvensional memiliki kelebihan dalam kemudahan implementasi awal dan kesederhanaan pengelolaan sistem. Arsitektur monolitik dan integrasi berbasis *Enterprise Service Bus* (ESB) memungkinkan pengelolaan sistem secara terpusat dan terkontrol[13]. Namun, metode tersebut memiliki sejumlah kelemahan,

antara lain tingkat ketergantungan (*coupling*) yang tinggi antar sistem, keterbatasan skalabilitas, meningkatnya beban server pusat ketika volume data dan jumlah layanan bertambah, risiko di mana perubahan pada satu integrasi dapat mengganggu kestabilan integrasi lainnya, serta potensi *bottleneck* akibat sifat ESB yang terpusat. [13-15]. Selain itu, komunikasi sinkron antar sistem berpotensi menimbulkan latensi tinggi dan kegagalan sistem secara menyeluruh apabila salah satu komponen mengalami gangguan [13-16].

Dalam konteks integrasi *Smart Village* ke *Smart City*, penggunaan komunikasi sinkron berbasis REST API tanpa mekanisme asinkron yang memadai menimbulkan permasalahan pada ketersediaan dan keandalan sistem akibat adanya keterikatan waktu (*temporal coupling*) dan risiko kegagalan berantai (*cascading failures*) [17,18]. Proses sinkronisasi data secara langsung antara sistem desa dan sistem kabupaten dapat memicu beban server yang sangat tinggi, karena setiap pembaruan data yang dilakukan secara bersamaan dari banyak sumber sering kali mengakibatkan kemacetan kinerja (*performance bottleneck*). Meskipun sinkronisasi *real-time* sangat penting untuk menjaga integritas dan kebaruan informasi, ketergantungan pada metode komunikasi sinkron ini dapat meningkatkan latensi secara signifikan. Selain itu, kegagalan pada sistem pusat berpotensi menghambat layanan di tingkat desa, sehingga mengurangi fleksibilitas pengembangan layanan publik digital. Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan perancangan arsitektur interoperabilitas berbasis mikroservis dengan pendekatan SOA modern yang dikombinasikan dengan penggunaan *API Gateway* dan *message broker* Apache Kafka. Arsitektur mikroservis memungkinkan setiap layanan desa dikembangkan dan dikelola secara independen, Sementara *API Gateway* berperan sebagai pengelola lalu lintas komunikasi antar layanan sekaligus bertindak sebagai pintu masuk terpusat bagi seluruh permintaan dari klien ke sistem mikroservis [18-21]. Apache Kafka digunakan sebagai *message broker* untuk mendukung distribusi data yang dipicu oleh peristiwa (*event-driven*) secara asinkron dan *real-time* [22-24]. Pendekatan ini diharapkan mampu mengurangi ketergantungan langsung antar sistem, meningkatkan skalabilitas, serta menjaga stabilitas sistem pusat Smart

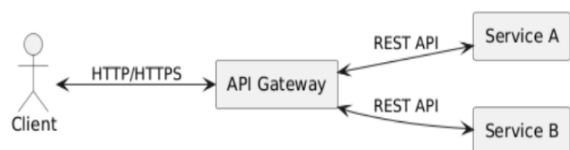
City meskipun terjadi peningkatan aktivitas layanan di tingkat desa. Oleh karena itu, penelitian ini memposisikan interoperabilitas data sebagai fokus utama, sementara aspek skalabilitas dan ketahanan sistem diperlakukan sebagai konsekuensi arsitektural dari desain interoperabilitas yang diusulkan.

Berdasarkan permasalahan tersebut, tujuan penelitian ini adalah merancang sebuah framework arsitektural integrasi layanan *Smart Village* ke *Smart City* menggunakan arsitektur mikroservis berbasis SOA modern dengan dukungan *API Gateway* dan Apache Kafka sebagai *message broker*. Penelitian ini bertujuan untuk menghasilkan rancangan arsitektur interoperabilitas data yang didukung oleh fleksibilitas, skalabilitas, dan keandalan sistem sehingga pengembangan layanan publik digital di tingkat desa dapat dilakukan tanpa mengganggu stabilitas sistem pusat di tingkat kabupaten atau kota.

II. TINJAUAN PUSTAKA

2.1 Application Programming Interface

Application Programming Interface (API) adalah antarmuka yang memungkinkan layanan sistem perangkat lunak berkomunikasi secara terstandarisasi untuk bertukar data dan fungsi tanpa mengungkap detail internal sistem. Dalam konteks interoperabilitas data, API berperan sebagai mekanisme utama pertukaran data lintas sistem yang bersifat heterogen, baik dari sisi platform, bahasa pemrograman, maupun domain organisasi. Dalam arsitektur mikroservis, API memainkan peran sebagai mekanisme *loose coupling* antar layanan yang bersifat independen, sehingga memungkinkan integrasi data antar layanan dilakukan tanpa ketergantungan langsung pada implementasi internal masing-masing layanan. RESTful API adalah pendekatan API yang umum dipakai dalam komunikasi mikroservis dengan protokol HTTP karena kesederhanaannya dan dukungan luas pada sistem terdistribusi modern[25]. Selain *Representational State Transfer* (REST), API dapat dikembangkan dengan pendekatan lain seperti GraphQL atau gRPC untuk kebutuhan performa dan interoperabilitas yang lebih kompleks[26]. Namun, penelitian ini memfokuskan penggunaan REST API sebagai mekanisme interoperabilitas data yang paling umum diadopsi pada sistem pemerintahan dan layanan publik digital.



Gambar 1. Ilustrasi API sebagai antarmuka komunikasi layanan mikroservis

Gambar 1 menggambarkan peran *API Gateway* sebagai mediator interoperabilitas data antara klien eksternal dan layanan mikroservis. Klien mengirimkan permintaan melalui protokol HTTP/HTTPS ke *API Gateway*, yang kemudian meneruskan permintaan tersebut ke layanan-layanan mikro seperti *Service A* dan *Service B* melalui REST API. *API Gateway* berfungsi sebagai titik integrasi terpusat untuk mengelola pertukaran data lintas layanan, sekaligus menangani fungsi *routing*, autentikasi, dan agregasi layanan. Pendekatan ini menyederhanakan interoperabilitas data antar sistem eksternal dan internal tanpa mengekspos struktur layanan mikro secara langsung, sehingga mendukung integrasi yang lebih terkontrol dalam ekosistem mikroservis yang kompleks.

2.2 Service-Oriented Architecture dan Mikroservis

Service-Oriented Architecture (SOA) adalah model arsitektur *software* yang fokus pada layanan independen yang dapat berinteraksi melalui protokol standar dalam jaringan. SOA dirancang untuk mendukung interoperabilitas antar sistem melalui standarisasi layanan dan kontrak komunikasi. Mikroservis dipandang sebagai evolusi dari prinsip SOA yang memecah sistem besar menjadi layanan kecil yang otonom dan mudah diimplementasikan secara independen. Keunggulan mikroservis dibanding model monolitik yaitu skalabilitas, toleransi kesalahan yang lebih tinggi, dan siklus pengembangan yang lebih cepat [27-28].

Beberapa penelitian membahas perbedaan antara SOA dan *Microservices Architecture*. Namun, penelitian ini tidak bertujuan membandingkan kedua pendekatan tersebut, melainkan memanfaatkan prinsip SOA sebagai dasar interoperabilitas layanan dan mengimplementasikannya melalui arsitektur mikroservis yang lebih ringan dan terdistribusi untuk mendukung integrasi data *Smart Village* ke *Smart City* [29].

Tabel 1. Perbandingan Konseptual SOA dan Mikroservis berdasarkan Studi Literatur [28].

Aspek	SOA	Mikroservis
Granularitas layanan	Menengah	Lebih kecil
Deployment	Agregat layanan	Independen per layanan
Komunikasi	Banyak bergantung pada ESB	<i>API asynchronous messaging</i> +
Skalabilitas	Terbatas	Tinggi
Integrasi	Terpusat	Terdistribusi

Meskipun penelitian ini tidak berfokus pada analisis komparatif mendalam antara SOA dan mikroservis, Tabel 1 disajikan untuk menjustifikasi pemilihan arsitektur mikroservis sebagai pendekatan yang lebih adaptif dalam mendukung interoperabilitas data lintas sistem. Mikroservis menawarkan *granularitas* layanan yang lebih kecil dan integrasi yang lebih terdistribusi, sehingga memudahkan pemisahan domain data dan integrasi antar layanan *Smart Village* dan *Smart City* tanpa ketergantungan terpusat.

2.3 API Gateway

API Gateway merupakan komponen *middleware* yang berfungsi sebagai pintu gerbang tunggal untuk seluruh permintaan eksternal menuju kumpulan layanan mikro. Dalam konteks interoperabilitas data, *API Gateway* berperan sebagai mediator komunikasi lintas sistem yang mengabstraksikan kompleksitas layanan internal. Fungsi utama *API Gateway* meliputi *routing* permintaan, autentikasi dan otorisasi, penerapan *rate limiting* dan *throttling*, serta agregasi respons dari beberapa layanan. Dengan pendekatan ini, integrasi data antar sistem eksternal dan internal dapat dilakukan melalui satu titik akses terstandarisasi, sehingga mengurangi kompleksitas interoperabilitas seiring bertambahnya jumlah layanan mikro.

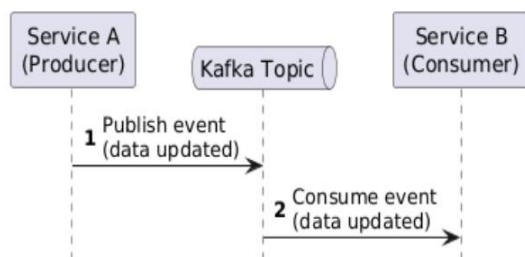
2.4 Apache Kafka sebagai Message Broker

Apache Kafka merupakan platform *distributed event streaming* yang mendukung komunikasi asinkron antar layanan mikro[29]. Dalam penelitian ini, Kafka diposisikan sebagai mekanisme pertukaran data lintas

domain secara *event-driven*, bukan sebagai fokus utama optimasi performa sistem.

Sebagai *message broker*, Kafka memungkinkan layanan untuk menerbitkan dan mengonsumsi peristiwa pembaruan data secara *real-time* atau *near real-time*. Pendekatan ini mendukung interoperabilitas data tanpa ketergantungan langsung antar layanan, karena produsen dan konsumen data tidak perlu saling mengetahui status atau ketersediaan satu sama lain.

Kafka juga mendukung berbagai pola komunikasi seperti *event streaming*, *Command Query Responsibility Segregation* (CQRS), dan *saga pattern* [29]. Namun, penelitian ini memanfaatkan Kafka terutama sebagai *enabler* integrasi data asinkron antara sistem Smart Village dan Smart City, guna mengurangi *temporal coupling* dan meningkatkan ketahanan integrasi.



Gambar 2. Pola Event-Driven dengan Kafka

Gambar 2 merupakan ilustrasi pola *event-driven* menggunakan Apache Kafka, yang menunjukkan alur komunikasi asinkron antara dua layanan mikro. Dalam diagram tersebut, *Service A* berperan sebagai *producer* yang menerbitkan *event* bertipe "data updated" ke dalam *Kafka Topic*, sementara *Service B* bertindak sebagai *consumer* yang mengonsumsi *event* tersebut dari topik yang sama. Pola ini memungkinkan pertukaran data lintas layanan dilakukan secara asinkron dan terdistribusi, sehingga meningkatkan fleksibilitas dan interoperabilitas dalam sistem pemerintahan yang kompleks.

2.5 Komunikasi Sinkron dan Asinkron

Komunikasi sinkron merupakan pola interaksi di mana pengirim menunggu respons langsung dari penerima sebelum melanjutkan proses berikutnya. Pola ini sesuai untuk skenario *request-response*, namun berpotensi

menimbulkan latensi tinggi dan keterikatan waktu yang dapat menghambat interoperabilitas pada sistem terdistribusi. Sebaliknya, komunikasi asinkron memungkinkan pengiriman pesan atau *event* tanpa menunggu respons langsung. Pola ini lebih sesuai untuk pertukaran data lintas sistem yang bersifat heterogen, karena mendukung *decoupling* antar layanan dan toleransi terhadap gangguan[29-33]. Kombinasi komunikasi sinkron dan asinkron digunakan untuk menyeimbangkan kebutuhan akses data langsung dan distribusi pembaruan data lintas level pemerintahan.

2.6 Metrik Kinerja pada Arsitektur Mikroservis

Dalam evaluasi arsitektur terdistribusi, metrik kinerja yang umum digunakan meliputi *throughput*, *latency*, dan *error rate*[34-37]. *Throughput* merepresentasikan jumlah pesan atau permintaan yang dapat diproses sistem dalam satuan waktu tertentu, sedangkan *latency* menunjukkan waktu respons yang dibutuhkan untuk menyelesaikan suatu permintaan[38]. *Error rate* menggambarkan persentase permintaan yang gagal diproses dan sering digunakan sebagai indikator ketahanan sistem (*fault tolerance*). Metrik-metrik ini digunakan dalam penelitian sebagai dasar analisis implikasi arsitektural terhadap interoperabilitas data, khususnya dalam konteks penggunaan *API Gateway* dan Apache Kafka. Evaluasi difokuskan pada bagaimana arsitektur mampu menjaga pertukaran data lintas layanan secara andal tanpa meningkatkan ketergantungan langsung antar komponen sistem.

III. METODE PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan metode *System Development Life Cycle* (SDLC) untuk menghasilkan rancangan arsitektur interoperabilitas data *Smart Village* ke *Smart City* berbasis mikroservis yang berfokus pada mekanisme pertukaran data lintas sistem pemerintahan secara terstandarisasi dan terdistribusi [38-40].

3.1 Desain Penelitian

Desain penelitian bersifat *design science* pada domain rekayasa perangkat lunak, dengan keluaran utama berupa artefak rancangan arsitektur yaitu *high-level architecture*,

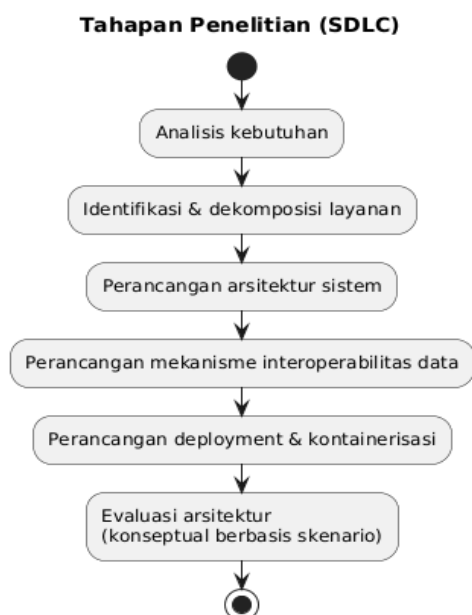
rancangan komponen, rancangan *deployment*, dan rancangan mekanisme pertukaran data sinkron–asinkron.

3.2 Objek dan Ruang Lingkup

Objek penelitian adalah layanan publik digital pada tingkat *Smart Village* yang mencakup tiga domain layanan yaitu (1) kependudukan, (2) perizinan, dan (3) bantuan sosial, serta integrasinya ke sistem *Smart City* di tingkat kabupaten/kota dalam konteks pertukaran dan sinkronisasi data antar sistem yang bersifat heterogen. Ruang lingkup dibatasi pada perancangan arsitektur sistem dan mekanisme interoperabilitas data baik secara sinkron maupun asinkron, tanpa membahas implementasi antarmuka pengguna (*front-end*) secara detail.

3.3 Tahapan Penelitian (SDLC)

Tahapan penelitian disusun berdasarkan SDLC yang disesuaikan untuk kebutuhan arsitektur mikroservis dalam mendukung interoperabilitas data lintas sistem *Smart Village* dan *Smart City*. Berikut merupakan penjelasan gambar 3:



Gambar 3. UML Activity Diagram SDLC

Tahap pertama adalah analisis kebutuhan, yang mencakup identifikasi kebutuhan fungsional seperti akses data, sinkronisasi, dan monitoring, serta kebutuhan non-

fungsional seperti skalabilitas, ketahanan terhadap gangguan, keamanan akses, dan kemudahan evolusi layanan. Selain itu, tahap ini menetapkan batas sistem antara *Smart Village* dan *Smart City* serta mendefinisikan kebutuhan pertukaran data antar keduanya sebagai dasar interoperabilitas.

Tahap kedua adalah identifikasi dan dekomposisi layanan, yang dilakukan dengan pendekatan *domain-driven decomposition*[41]. Layanan *Smart Village* diuraikan menjadi mikroservis berdasarkan domain bisnis, yaitu *Population Service*, *Licensing Service*, dan *Social Aid Service*. Prinsip kemandirian layanan (*independent deployability*) dan pemisahan data (*database per service*) diterapkan untuk mengurangi tingkat keterikatan antar layanan (*coupling*), sehingga setiap domain data dapat diintegrasikan secara fleksibel tanpa ketergantungan langsung antar sistem.

Tahap ketiga adalah perancangan arsitektur sistem, yang mencakup penyusunan *microservices layer*, *integration layer*, dan komponen konsumen dari *Smart City*. Dalam arsitektur ini, *API Gateway* ditetapkan sebagai titik akses tunggal untuk komunikasi sinkron sebagai mediator interoperabilitas data, sementara Apache Kafka digunakan sebagai infrastruktur pertukaran *event* untuk mendukung distribusi data lintas sistem secara asinkron dan terdistribusi. Kombinasi kedua komponen ini memungkinkan sistem untuk menangani berbagai pola komunikasi secara efisien dan fleksibel.

Tahap keempat adalah perancangan mekanisme interoperabilitas data, yang mencakup dua pendekatan utama. Pola sinkron berbasis RESTful API (*request–response*) dirancang untuk kebutuhan *query* atau validasi data secara langsung antar sistem *Smart Village* dan *Smart City*. Sementara itu, pola asinkron berbasis *event* (*publish–subscribe*) digunakan untuk pembaruan data secara *real-time* atau *near real-time*, seperti *event* “data updated”. Pendekatan ini memastikan bahwa sistem *Smart Village* tetap dapat beroperasi meskipun sistem *Smart City* sedang tidak tersedia, sehingga interoperabilitas data tidak bergantung pada ketersediaan seluruh komponen sistem.

Tahap kelima adalah perancangan *deployment* dan containerisasi, yang dilakukan dengan menggunakan teknologi kontainer seperti Docker. Pendekatan ini

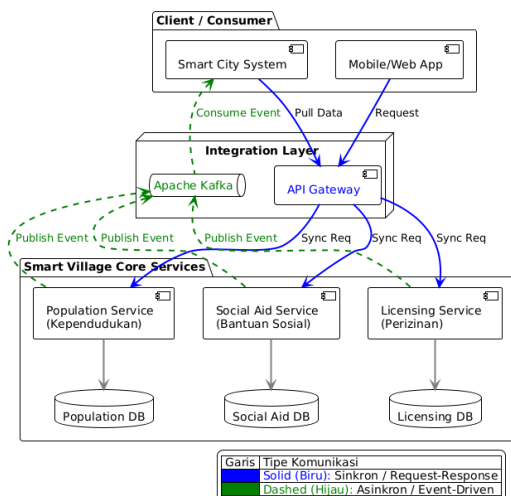
bertujuan untuk menjamin konsistensi lingkungan eksekusi, memudahkan replikasi, dan mendukung skalabilitas sistem dalam konteks integrasi dan pertukaran data lintas layanan. Selain itu, tahap ini juga menetapkan keterhubungan layanan ke Kafka *cluster* serta jalur komunikasi melalui *API Gateway*, sehingga seluruh komponen sistem dapat berinteraksi secara optimal dalam lingkungan terdistribusi.

Tahap terakhir adalah evaluasi arsitektur secara konseptual berbasis skenario penggunaan untuk menilai kesiapan interoperabilitas data yang diusulkan. Evaluasi dilakukan terhadap tiga skenario utama yaitu (a) permintaan data sinkron dari Smart City, (b) pembaruan data desa yang memicu *event*, dan (c) kondisi gangguan sementara pada komponen konsumen. Metrik evaluasi yang dianalisis meliputi latensi komunikasi (baik REST maupun propagasi *event*), *throughput* pertukaran data, serta ketahanan terhadap kegagalan layanan (*fault tolerance*).

IV. HASIL DAN PEMBAHASAN

4.1 Desain Arsitektur

Desain Arsitektur Mikroservis Smart Village

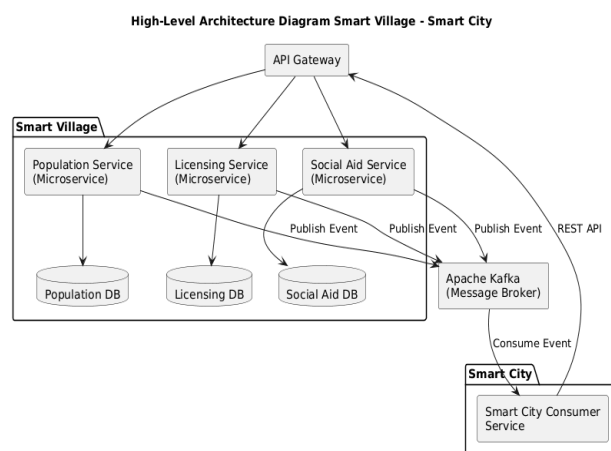


Gambar 4. Desain Arsitektur Mikroservis

Hasil perancangan pada gambar 4 menunjukkan bahwa sistem Smart Village dipecah menjadi tiga layanan inti berbasis domain, yaitu *Population Service* (kependudukan), *Licensing Service* (perizinan), dan *Social Aid Service* (bantuan sosial). Setiap layanan dirancang

sebagai mikroservis mandiri dengan tanggung jawab spesifik untuk memisahkan domain data dan mendukung interoperabilitas lintas layanan, sekaligus menekan *coupling* dan memudahkan pengembangan atau skalabilitas per layanan.

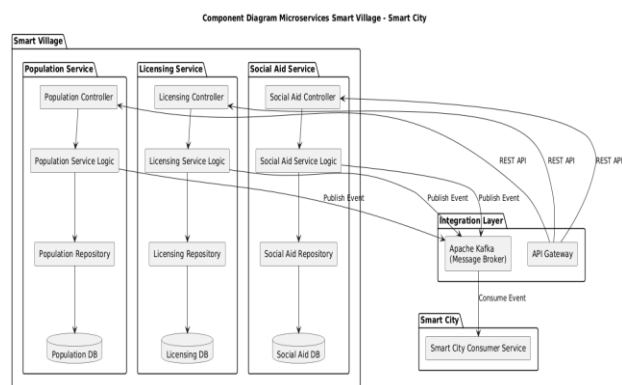
Di sisi integrasi, arsitektur menambahkan *integration layer* yang terdiri dari *API Gateway* dan Apache Kafka untuk melayani dua kebutuhan utama yaitu akses data sinkron (*request-response*) dan pertukaran data asinkron (*event-driven*). Pola ini dipilih agar sistem Smart City dapat melakukan konsumsi data secara interoperabel tanpa ketergantungan langsung dan tanpa membebani layanan desa melalui pemanggilan sinkron terus menerus, sekaligus menjaga ketersediaan ketika terjadi gangguan sementara pada salah satu komponen.



Gambar 5. High-Level Architecture Smart Village ke Smart City

Gambar 5 merupakan representasi arsitektur tingkat tinggi (*high-level architecture*) yang menggambarkan integrasi sistem Smart Village dengan Smart City berbasis pendekatan mikroservis dalam konteks pertukaran data lintas level pemerintahan. Dalam arsitektur ini, Smart Village terdiri dari tiga layanan utama yang terdekomporsi berdasarkan domain bisnis, yaitu *Population Service*, *Licensing Service*, dan *Social Aid Service*. Masing-masing layanan dirancang sebagai mikroservis yang berdiri sendiri dan terhubung ke basis data terpisah, sehingga mendukung prinsip *independent deployability* dan mengurangi *coupling* antar komponen dalam proses interoperabilitas data.

Untuk mendukung interoperabilitas, arsitektur ini mengimplementasikan *integration layer* yang terdiri dari *API Gateway* dan Apache Kafka. *API Gateway* berfungsi sebagai titik akses tunggal untuk komunikasi sinkron berbasis REST API sebagai mediator interoperabilitas data, memungkinkan klien eksternal seperti sistem Smart City melakukan permintaan data secara langsung. Sementara itu, Apache Kafka berperan sebagai *message broker* yang menerima *event* dari masing-masing layanan mikroservis dan mendistribusikannya ke *Smart City Consumer Service* secara asinkron sebagai mekanisme pertukaran data lintas domain layanan.



Gambar 6. Component Diagram Microservices

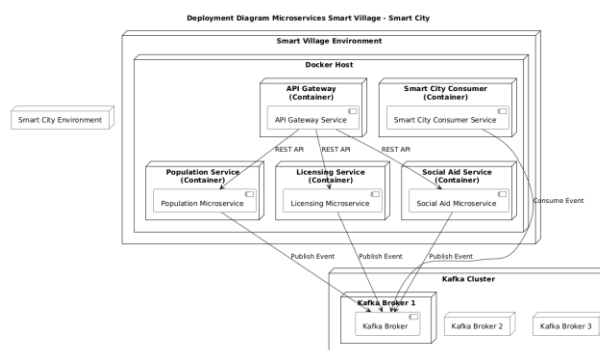
Gambar 6 merupakan *Component Diagram* yang menggambarkan struktur internal arsitektur mikroservis untuk integrasi Smart Village ke Smart City. Diagram ini terbagi menjadi tiga lapisan utama yaitu Smart Village, *Integration Layer*, dan Smart City, yang masing-masing memiliki peran spesifik dalam mendukung interoperabilitas data secara modular dan terdistribusi.

Pada lapisan Smart Village, terdapat tiga layanan mikro yang terdekomposisi berdasarkan domain bisnis, yaitu *Population Service*, *Licensing Service*, dan *Social Aid Service*. Setiap layanan terdiri dari komponen *Controller*, *Service Logic*, *Repository*, dan *Database* yang berdiri sendiri. Struktur ini mencerminkan prinsip *separation of concerns* dan *independent deployability*, sehingga setiap domain data dapat diintegrasikan tanpa ketergantungan langsung antar layanan.

Lapisan *Integration Layer* berfungsi sebagai penghubung antara layanan mikro dan sistem eksternal untuk

memfasilitasi interoperabilitas data lintas sistem. Di dalamnya terdapat *API Gateway* yang menangani komunikasi sinkron berbasis REST API, serta *Apache Kafka* sebagai *message broker* untuk komunikasi asinkron berbasis *event*. *Controller* dari masing-masing layanan mikro berinteraksi dengan *API Gateway* untuk permintaan langsung, sementara *event* yang dihasilkan oleh layanan dipublikasikan ke Kafka untuk didistribusikan ke konsumen.

Lapisan terakhir, Smart City, terdiri dari *Smart City Consumer Service* yang berfungsi sebagai penerima event dari Kafka. Dengan pola ini, sistem Smart City dapat mengonsumsi data secara *real-time* secara interoperabel tanpa membebani layanan desa dengan permintaan sinkron yang berulang. Selain itu, pendekatan ini meningkatkan ketahanan sistem terhadap gangguan sementara dan mendukung skalabilitas horizontal.



Gambar 7. Deployment Diagram Microservices

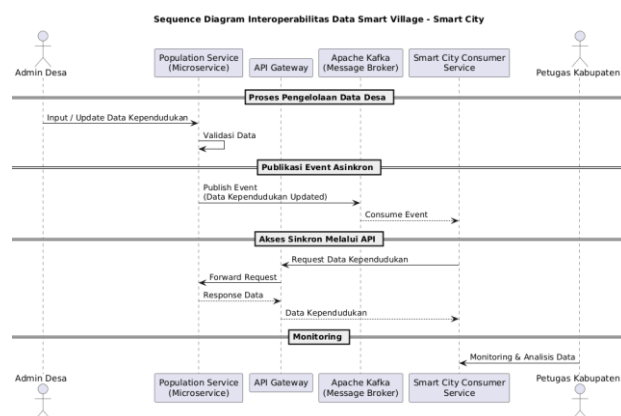
Gambar 7 merupakan *Deployment Diagram* yang menggambarkan bagaimana komponen-komponen mikroservis Smart Village dan Smart City diimplementasikan dalam lingkungan kontainerisasi untuk mendukung interoperabilitas data dalam lingkungan terdistribusi. Diagram ini menunjukkan bahwa seluruh layanan dijalankan dalam kontainer yang dikelola oleh *Docker Host* di lingkungan Smart Village.

Layanan yang dideploy meliputi *Population Service*, *Licensing Service*, dan *Social Aid Service*, masing-masing sebagai mikroservis independen. Selain itu, terdapat *API Gateway* sebagai titik akses sinkron dan *Smart City Consumer Service* sebagai komponen yang mengonsumsi *event* dari sistem desa. Komunikasi antar layanan

dilakukan melalui REST API, sementara pertukaran data asinkron difasilitasi oleh *Kafka Cluster* yang terdiri dari tiga *broker*, *Kafka Broker* 1, 2, dan 3. Setiap layanan mikroservis mempublikasikan *event* ke Kafka, yang kemudian didistribusikan ke konsumen Smart City. Diagram ini menekankan pentingnya arsitektur berbasis kontainer dan *event-driven* dalam menjaga keberlanjutan interoperabilitas antar wilayah administratif.

4.2 Mekanisme Interoperabilitas Data

Mekanisme interoperabilitas dirancang menggunakan dua mode komunikasi yang saling melengkapi yaitu sinkron melalui REST API dan asinkron melalui *event streaming*. Pada komunikasi sinkron, Smart City mengakses layanan desa melalui *API Gateway* sebagai *single entry point* untuk menjamin pertukaran data yang terkontrol dan terstandarisasi. Pada komunikasi asinkron, setiap pembaruan data penting di layanan desa memicu publikasi *event* ke Apache Kafka, dan Smart City Consumer Service mengambil *event* tersebut sebagai bentuk interoperabilitas data berbasis peristiwa.



Gambar 8. Sequence Diagram Interoperabilitas Data

Gambar 8 merupakan *Sequence Diagram* yang menggambarkan alur interoperabilitas data antara sistem Smart Village dan Smart City melalui kombinasi komunikasi sinkron dan asinkron. Diagram ini memvisualisasikan interaksi antar komponen utama, yaitu *Admin Desa*, *Population Service*, *API Gateway*, *Apache Kafka*, *Smart City Consumer Service*, dan *Petugas Kabupaten*, dalam empat proses terstruktur.

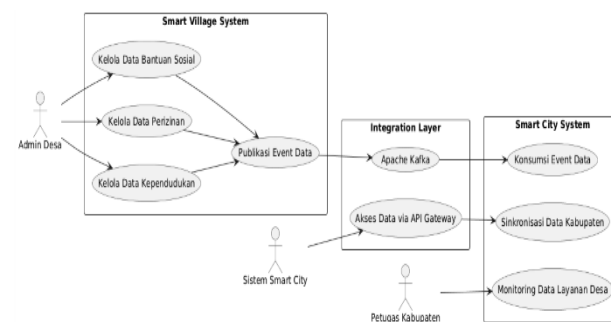
Pada proses pertama, *Pengelolaan Data Desa*, Admin Desa melakukan *input* atau pembaruan data kependudukan melalui *Population Service*, yang kemudian melakukan validasi terhadap data tersebut. Proses ini mencerminkan aktivitas lokal yang memicu alur integrasi data.

Proses kedua adalah *Publikasi Event Asinkron*, di mana *Population Service* menerbitkan *event* “data updated” ke Apache Kafka sebagai *message broker*. Kafka kemudian mendistribusikan event tersebut ke *Smart City Consumer Service*, memungkinkan sistem Smart City menerima pembaruan data secara *real-time* tanpa harus melakukan permintaan langsung secara terus-menerus.

Proses ketiga, *Akses Sinkron Melalui API*, menunjukkan bagaimana Petugas Kabupaten dapat mengakses data kependudukan secara langsung melalui *API Gateway*. Permintaan dikirim ke *Population Service*, yang merespons dengan data yang diminta, lalu dikembalikan ke *Petugas Kabupaten*. Pola ini mendukung kebutuhan *query* atau validasi data secara eksplisit dan cepat.

Proses terakhir adalah *Monitoring*, di mana *Smart City Consumer Service* melakukan pemantauan dan analisis terhadap data yang diterima melalui *event*. Proses ini penting untuk mendukung pengambilan keputusan berbasis data di tingkat kabupaten atau kota.

4.3 Pembahasan Artefak Perancangan



Gambar 9. Use Case Diagram

Gambar 9 merupakan *Use Case Diagram* yang menggambarkan interaksi antara aktor utama dan sistem dalam ekosistem Smart Village ke Smart City, dengan fokus pada kebutuhan integrasi data dan interoperabilitas layanan. Diagram ini menampilkan dua aktor utama, yaitu

Admin Desa dan *Petugas Kabupaten*, serta entitas sistem Smart City, yang berinteraksi dengan layanan inti Smart Village melalui lapisan integrasi.

Admin Desa memiliki peran dalam mengelola tiga jenis data utama yaitu *Data Kependudukan*, *Data Perizinan*, dan *Data Bantuan Sosial*. Setiap aktivitas pengelolaan data ini menghasilkan *event* yang dipublikasikan melalui Apache Kafka, memungkinkan sistem Smart City untuk menerima pembaruan secara asinkron. Di sisi lain, *Petugas Kabupaten* dapat mengakses data secara sinkron melalui *API Gateway*, misalnya untuk keperluan validasi atau monitoring langsung.

Lapisan integrasi yang terdiri dari *API Gateway* dan Apache Kafka berfungsi sebagai penghubung antara sistem Smart Village dan Smart City, mendukung dua pola komunikasi yaitu *request-response* untuk akses langsung, dan *publish-subscribe* untuk pertukaran data *real-time*. Sistem Smart City kemudian memanfaatkan data yang diterima untuk melakukan *sinkronisasi*, *monitoring*, dan *analisis layanan desa*.

Rancangan *use case* ini tidak berfokus pada antarmuka pengguna (UI), melainkan pada justifikasi ruang lingkup layanan inti dan kebutuhan integrasi data. Diagram ini memperjelas aktor utama (*Admin Desa* dan *pihak kabupaten*) serta cakupan layanan Smart Village yang terintegrasi (kependudukan, perizinan, bantuan sosial). Artefak ini digunakan untuk menjustifikasi batas ruang lingkup penelitian pada layanan inti dan kebutuhan integrasi data, bukan implementasi UI aplikasi.

4.4 Evaluasi Konseptual

Evaluasi dilakukan secara konseptual berbasis skenario untuk menilai kesiapan arsitektur dalam mendukung interoperabilitas, skalabilitas, dan ketahanan layanan dalam konteks pertukaran data lintas sistem Smart Village dan Smart City. Skenario yang digunakan meliputi (a) permintaan data sinkron dari Smart City melalui *API Gateway* sebagai mekanisme interoperabilitas sinkron, (b) pembaruan data di Smart Village yang memicu *event* dan diproses *consumer Smart City* sebagai bentuk interoperabilitas data berbasis *event*, serta (c) gangguan sementara pada komponen *consumer* yang menguji kemampuan sistem tetap berjalan tanpa kegagalan

menyeluruh dalam skenario integrasi lintas level pemerintahan.

Ringkasan hasil evaluasi konseptual berdasarkan skenario dan metrik arsitektural ditunjukkan pada Tabel 2.

Tabel 2. Evaluasi Konseptual Arsitektur Berbasis Skenario

Aspek Evaluasi	Pendekatan Sinkron (REST)	Pendekatan Event-Driven (Kafka)
Latensi	Rendah untuk permintaan langsung	Tergantung propagasi event
Throughput	Terbatas oleh kapasitas layanan	Lebih fleksibel dan terdistribusi
Ketahanan Sistem	Rentan jika consumer gagal	Lebih tahan terhadap kegagalan
Beban Server	Tinggi saat trafik meningkat	Lebih terdistribusi
Interoperabilitas	Terbatas sinkron	Tinggi lintas sistem

Berdasarkan Tabel 2, pendekatan *event-driven* berbasis Apache Kafka menunjukkan keunggulan pada aspek interoperabilitas dan ketahanan sistem dalam mendukung pertukaran data lintas sistem yang heterogen dibandingkan pendekatan sinkron murni. Sementara komunikasi REST tetap relevan untuk permintaan data langsung dengan latensi rendah pada skenario interoperabilitas sinkron, mekanisme *event-driven* memberikan fleksibilitas dan distribusi beban yang lebih baik pada skenario integrasi lintas sistem Smart Village ke Smart City.

Metrik yang dianalisis pada evaluasi ini mencakup latensi komunikasi (REST dan propagasi *event*), *throughput* pertukaran data, serta ketahanan terhadap kegagalan layanan (*fault tolerance*) sebagai indikator kesiapan interoperabilitas arsitektur yang diusulkan. Secara desain, kombinasi REST dengan *event-driven* berpotensi mengurangi beban integrasi sinkron langsung karena pembaruan rutin dapat dialihkan menjadi *event* yang mendukung pertukaran data lintas sistem secara tidak

bergantung waktu (*temporal decoupling*) dan dikonsumsi sesuai kapasitas sistem kabupaten.

Implikasi dari rancangan ini adalah berkurangnya beban permintaan sinkron langsung ke layanan Smart Village karena pembaruan rutin dapat dialirkan melalui mekanisme *event-driven*, sehingga konsumsi data oleh Smart City menjadi lebih efisien pada kondisi trafik tinggi. Selain itu, penggunaan komunikasi asinkron meningkatkan ketahanan sistem karena layanan desa tetap dapat beroperasi ketika komponen di sisi kabupaten (*consumer*) mengalami gangguan sementara, lalu pemrosesan dapat dilanjutkan saat layanan kembali normal tanpa mengganggu kesinambungan pertukaran data antar sistem. Dari sisi pengembangan, pemisahan layanan menjadi mikroservis serta penggunaan *API Gateway* memudahkan evolusi layanan dalam konteks integrasi dan interoperabilitas data, baik dalam penambahan maupun perubahan *endpoint* serta peningkatan skala tiap layanan, tanpa harus mengubah keseluruhan sistem secara monolitik. Dengan demikian, adaptasi terhadap kebutuhan layanan publik desa dapat dilakukan lebih cepat.

Rancangan interoperabilitas yang diusulkan bersifat generik pada level arsitektural dan dapat dipetakan ke berbagai standar interoperabilitas pada tahap implementasi sistem. Namun, integrasi standar tersebut berada di luar ruang lingkup penelitian ini yang berfokus pada perancangan dan evaluasi konseptual arsitektur.

Evaluasi pada penelitian ini difokuskan pada analisis konseptual dan implikasi arsitektural terhadap interoperabilitas data, sehingga pengukuran kuantitatif performa menjadi ruang penelitian lanjutan pada tahap implementasi sistem.

V. KESIMPULAN

Penelitian ini merancang arsitektur interoperabilitas data Smart Village ke Smart City berbasis mikroservis yang memecah layanan desa menjadi tiga layanan inti (kependudukan, perizinan, bantuan sosial) serta *integration layer* yang memanfaatkan *API Gateway* dan Apache Kafka sebagai *enabler* interoperabilitas data. Mekanisme interoperabilitas menggabungkan komunikasi sinkron melalui REST API untuk kebutuhan *request-*

response pada akses data langsung lintas sistem dan komunikasi asinkron berbasis *publish-subscribe event* untuk pembaruan data *real-time* atau *near real-time* pada skenario pertukaran data terdistribusi, sehingga ketergantungan langsung antar sistem dapat dikurangi.

Secara konseptual, rancangan ini berpotensi meningkatkan fleksibilitas pengembangan dan skalabilitas karena tiap layanan dapat dikembangkan serta diskalakan secara independen tanpa mengganggu mekanisme interoperabilitas data, sekaligus meningkatkan ketahanan integrasi melalui pemanfaatan pertukaran data berbasis *event* yang mendukung komunikasi lintas sistem secara *loosely coupled*. Evaluasi berbasis skenario dengan indikator latensi, *throughput*, dan *fault tolerance* mengindikasikan adanya peluang penurunan beban integrasi sinkron langsung di sisi layanan desa melalui pemanfaatan interoperabilitas asinkron, serta menjaga keberlangsungan layanan ketika konsumen di tingkat kabupaten mengalami gangguan sementara tanpa menghambat alur pertukaran data antar sistem.

Keterbatasan penelitian ini adalah fokus pada perancangan dan evaluasi konseptual terhadap interoperabilitas data, sehingga performa arsitektur belum divalidasi melalui uji empiris pada lingkungan implementasi. Rekomendasi penelitian lanjutan adalah membangun prototipe dan melakukan pengukuran kuantitatif (latensi *end-to-end*, *throughput* Kafka, dan reliabilitas saat kegagalan parsial) sebagai validasi interoperabilitas secara operasional, serta memperkuat tata kelola API (*versioning*, otorisasi, dan *audit log*) agar rancangan siap dioperasikan pada skala pemerintah daerah dalam ekosistem Smart City yang terintegrasi.

DAFTAR PUSTAKA

- [1] A. P. E. Susilowati, R. Rachmawati, and R. Rijanta, "Analysis of Smart Village Development in Supporting Smart City in Indonesia: A Systematic Review," Dec. 01, 2024, Muhammadiyah University of Surakarta. doi: 10.23917/forgeo.v38i3.4790.
- [2] T. Cahya, U. Aris, R. Pranacitra, S. Suriyanto, and W. Dewanto, "Smart Governance: Program Transformasional Digital Nasional Melalui Desa,

- Siapakah Indonesia?,” *IBLAM LAW REVIEW*, vol. 4, no. 2, pp. 217–223, May 2024, doi: 10.52249/ilr.v4i2.528.
- [3] D. Iswanto and Miskan, “From Smart City to Smart Village: Akselerasi Transformasi Digital dalam Optimalisasi Pelayanan Publik di Desa Kepatihan Kabupaten Gresik,” *Contemporary Public Administration Review*, vol. 2, no. 2, pp. 170–186, Feb. 2025, doi: 10.26593/copar.v2i2.8794.170-186.
- [4] E. Firmansyah and M. Agreindra Helmiawan, “Pengukuran Kesiapan Transformasi Digital Desa Kaduwulung Menuju Desa Cerdas Berbasis SNI ISO 37122:2019 Melalui Pemetaan Data Desa,” *Jurnal Ilmu-ilmu Informatika dan Manajemen*, vol. 19, no. 1, 2025, doi: 10.33481/infomans.vxxixx.
- [5] M. Aidin, “Transformasi Digital Administrasi Desa Melalui Sistem Informasi Desa: Kajian Pustaka Tentang Faktor Pendukung Dan Penghambat,” *Nusantara Journal of Multidisciplinary Science*, vol. 2, no. 8, 2025, [Online]. Available: <https://jurnal.intekom.id/index.php/njms>
- [6] I. Z. Adnan, Z. Adnan, M. I. Fadhlurrohman, and H. S. Fauzan, “Implementation of e-Government through System Village Information (SID) (Case Study of Talaga Jaya Village, Garut Regency),” *Sinergi International Journal of Communication Sciences*, vol. 2, no. 2, pp. 106–121, May 2024, doi: 10.61194/ijcs.v2i2.296.
- [7] M. Zidan nasution, I. Diah Delima, and E. Prasetyo, “Implementasi Smart Village Melalui Sistem Smart Desa Digital Dalam Meningkatkan Kualitas Pelayanan Di Desa Pasir Ampo Kecamatan Kresek Kab. Tangerang,” *Jurnal Ilmiah Wahana Pendidikan*, vol. 10, no. 7, pp. 752–776, Apr. 2024, doi: 10.5281/zenodo.11171231.
- [8] N. P. N. Prameswari, Ni Putu Anik Prabawati, and I Dewa Ayu Putri Wirantari, “Kualitas Pelayanan Publik Berbasis Elektronik Melalui Aplikasi SIPADU Dalam Mewujudkan Smart Village di Desa Dangin Puri Kangin Kota Denpasar,” *Socio-political Communication and Policy Review*, Jul. 2025, doi: 10.61292/shkr.271.
- [9] S. Dwi Putri Siregar and M. Irwan Padli Nasution, “Peran Data Integration Dalam Mewujudkan Interoperabilitas Sistem Informasi,” *Jurnal Penelitian Ilmu-Ilmu Sosial*, May 2025, doi: 10.5281/zenodo.15683326.
- [10] A. Hapsari and I. Permana, “The Implementation of Data Interoperability Services in the Government of West Sumatra Province,” *The Future of Education Journal*, vol. 4, p. Page, 2025, [Online]. Available: <https://journal.tofedu.or.id/index.php/journal/index>
- [11] E. K. Dewi, Y. Muhaemin, G. A. Sitompul, R. Y. Parinduri, and M. Rika, “Digital Transformation in Public Services: Evaluation of E-Government Implementation in Indonesia,” *Socius Jurnal Penelitian Ilmu-Ilmu Sosial*, 2025, doi: 10.5281/zenodo.17475222.
- [12] A. Yulianjani, G. Dito April Yanto Wijaya, and A. Cristopher, “Implementasi of E-Government With The RESTFUL API Method on the Population Data Management System in Pasar Kemis District,” *Jurnal Sensi*, 2025.
- [13] M. Manoj and V. Mohan, “Multi Agent Systems,” *International Journal for Multidisciplinary Research*, 2025, [Online]. Available: www.ijfmr.com
- [14] R. Bernsteiner, M. Blasisker, M. Kohlegger, C. Ploder, and S. Schlögl, “Performance Comparison between a Monolithic and a Microservice Application,” *SQAMIA*, 2025.
- [15] M. Kaloudis, “Evolving Software Architectures from Monolithic Systems to Resilient Microservices: Best Practices, Challenges and Future Trends,” 2024. [Online]. Available: www.ijacsa.thesai.org

- [16] S. Xie, H. He, J. Wang, and B. Li, "Root Cause Analysis for Microservice Systems via Cascaded Conditional Learning with Hypergraphs," *ArXiv*, Nov. 2025, [Online]. Available: <http://arxiv.org/abs/2511.17566>
- [17] N. S. Sigala, "Microservices Architecture in Cloud Computing: A Software Engineering Perspective on Design, Deployment, and Management," *IJRIS*, Mar. 2025, doi: 10.47772/IJRIS.
- [18] S. Xie, H. He, J. Wang, and B. Li, "Root Cause Analysis for Microservice Systems via Cascaded Conditional Learning with Hypergraphs," *ArXiv*, Nov. 2025, [Online]. Available: <http://arxiv.org/abs/2511.17566>
- [19] A. Neelan, "The Evolving Role of API Gateways in Scalable Microservices Architecture," *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 6, no. 4, Oct. 2025, doi: 10.63282/3050-9246.IJETCSIT-V6I4P102.
- [20] A. Ganje, "Microservices in Organizations," *Journal of Software Engineering and Applications*, vol. 18, no. 02, pp. 76–86, 2025, doi: 10.4236/jsea.2025.182005.
- [21] V. Punniyamoorthy *et al.*, "Secure and Governed API Gateway Architectures for Multi-Cluster Cloud Environments," *ArXiv*, Dec. 2025, [Online]. Available: <http://arxiv.org/abs/2512.23774>
- [22] M. Vidgof, "Towards Process Mining on Kafka Event Streams," 2024. [Online]. Available: <http://ceur-ws.org>
- [23] X. Cui *et al.*, "International Workshop on Online and Adaptive Recommender Systems (OARS 2024)," in *International Conference on Information and Knowledge Management, Proceedings*, Association for Computing Machinery, Oct. 2024, pp. 5580–5583. doi: 10.1145/3627673.3679083.
- [24] M. Ganesan, "Message Broker-Driven Event-Driven Microservice Application Architecture," Dec. 2025. [Online]. Available: <http://www.ijritcc.org>
- [25] A. Lercher, J. Glock, C. Macho, and M. Pinzger, "Microservice API Evolution in Practice: A Study on Strategies and Challenges," *Journal of Systems and Software*, vol. 215, Sep. 2024, doi: 10.1016/j.jss.2024.112110.
- [26] S. Mylsamy and E. A. Shrivastav, "API Design and Integration in a Microservices Environment," *International Journal for Research Publication and Seminar*, vol. 16, pp. 2278–6848, 2025, doi: 10.36676/jrps.v16.i2.205.
- [27] J. Giao, A. A. Nazarenko, F. Luis-Ferreira, D. Gonçalves, and J. Sarraipa, "A Framework for Service-Oriented Architecture (SOA)-Based IoT Application Development," *Processes*, vol. 10, no. 9, Sep. 2022, doi: 10.3390/pr10091782.
- [28] L. Rushani, F. Halili, and N. Macedonia, "Differences Between Service-Oriented Architecture and Microservices Architecture," 2022. [Online]. Available: https://ijnscfjournal.isrra.org/index.php/Natural_Sciences_Journal/index
- [29] A. N. Tran and S. H. Lim, "International Journal of Modern Computer Science and IT Innovations (IJMCSIT) A Critical Analysis of Apache Kafka's Role in Advancing Microservices Architecture: Performance, Patterns, and Persistence," vol. 10, pp. 99–107, 2025, [Online]. Available: <https://aimjournals.com/index.php/ijmcsit>
- [30] Rahmayani, R., Firmansyah, E., & Hikmah, H. U. (2025). Inovasi Layanan Antar Jemput Paket Surat PTPos Indonesia Berdasarkan Penjualan dan Minat Beli JOVISHE: Journal of Visionary Sharia Economy, 4(1), 33-47.
- [31] Firmansyah, E., Rahman, A. B. A., & Subiyakto, A. A. (2023). Pengukuran Kesiapan Kota Cerdas Berdasarkan SNI ISO 37122:

2019. Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen, 17(2).
- [32] Zulfikar, W. B., Irfan, M., Ghufro, M., Jumadi, J., & Firmansyah, E. (2020). Marketplace affiliates potential analysis using cosine similarity and vision-based page segmentation. *Bulletin of Electrical Engineering and Informatics*, 9(6), 2492-2498.
- [33] Ramadhan, N. D., Fhatturohmah, S., Ramadhani, S., & Firmansyah, E. (2023). Analysis of Digital Wallet Usage on Consumptive Lifestyle. *Journal of Islamic Economics and Business*, 3(2), 118-136.
- [34] Sutara, B., & Firmansyah, E. (2021). Design and Build Student Attendance System Using Fingerprint. *J-Tin's-Jurnal Teknik Informatika*, 5(1).
- [35] Wahana, A., Firmansyah, E., Al Rosyid, H. I., Fuadi, R. S., & Maylawati, D. S. A. (2021). Fuzzy Tahani Method in the Recommendation System for Selecting Mountain Tourism Destinations in West Java.
- [36] Tamrin, M. A., Rizki, B., Nodas, A., Rahman, A., & Firmansyah, E. (2020). Perbandingan Penggunaan Metode Topsis dan Metode AHP dalam Penilaian Kinerja pada Karyawan (PT XYZ). *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, 14(1).
- [37] Firmansyah, E., Herdiana, D., & Yuniarto, D. (2020, October). Examining readiness of e-Learning implementation using information system readiness impact model. In *2020 8th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE.
- [38] Firmansyah, E., Rosmawati, R., Fuadi, R. S., Fauzy, D., & Ramdhani, M. A. (2019, December). Design of expert system to determine the proper diet using harmony search method. In *Journal of Physics: Conference Series* (Vol. 1402, No. 7, p. 077006). IOP Publishing.
- [39] Firmansyah, E., & Helmiawan, M. A. (2025). Pengukuran Kesiapan Transformasi Digital Desa Kaduwulung Menuju Desa Cerdas Berbasis SNI ISO 37122: 2019 Melalui Pemetaan Data Desa. *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, 19(1).
- [40] Firmansyah, E., Herdiana, D., Yuniarto, D., & Junaedi, D. I. (2021, September). The K-Nearest Neighbor Algorithm for the Classification of Internet Users in Rural Campus. In *2021 9th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-6). IEEE.
- [41] Syaripudin, U., Zaenal, R., Duri, M. F. A., Firmansyah, E., & Rahman, A. (2019, December). Comparison between Naïve Bayes and certainty factor to predict big five personality. In *Journal of Physics: Conference Series* (Vol. 1402, No. 7, p. 077030). IOP Publishing.