

ARCHITECTURE DESCRIPTION*Berdasarkan Standar ISO/IEC/IEEE 42010:2022*

Nama Sistem	Sistem Presensi Mahasiswa Berbasis Face Recognition dengan Pendekatan Event-Driven Architecture (EDA)
Versi Dokumen	1.0
Tanggal	13/12/2025
Status	
Kelompok	Kelompok 5
Daftar Tim	Hendrawan (13020230309)

Kontrol Dokumen**Riwayat Revisi**

Versi	Tanggal	Penulis	Deskripsi Perubahan
1.0	13/12/2025	Hendrawan	Initial Release — Penyusunan Dokumen Deskripsi Arsitektur Sistem Presensi Mahasiswa Berbasis Face Recognition dengan EDA

Persetujuan Dokumen

Peran	Nama	Stambuk	Tanggal
Project Manager	Hendrawan	Hendrawan	13/12/2025
Technical Lead			
Quality Assurance	Mardiyah Hasnawi, S.Kom., M.T., MTA.		

Daftar Isi

Kontrol Dokumen.....	2
Riwayat Revisi	2
Persetujuan Dokumen	2
Daftar Isi.....	3
1. Pendahuluan	5
1.1 Tujuan Dokumen.....	5
1.2 Ruang Lingkup.....	5
1.3 Definisi dan Akronim.....	5
2. System of Interest (SoI).....	7
2.1 Identifikasi Sistem	7
2.2 Konteks Sistem.....	7
2.3 Misi dan Tujuan Sistem	7
3. Stakeholder dan Concern.....	8
3.1 Identifikasi Stakeholder	8
3.2 Concern (Kepentingan Arsitektural)	8
4. Architecture Viewpoint	10
4.1 Viewpoint Catalog	10
4.2 Viewpoint: Context (VP-01)	10
4.3 Viewpoint: Functional (VP-02)	10
4.4 Viewpoint: Information (VP-03).....	10
4.5 Viewpoint: Deployment (VP-04)	10
5. Architecture View	12
5.1 Context View (V-01)	12
5.1.1 Deskripsi View.....	12
5.1.2 Model: Context Diagram.....	12
5.2 Functional View (V-02).....	12
5.2.1 Deskripsi View.....	12
5.2.2 Model: Component Diagram	12
5.3 Information View (V-03)	13
5.3.1 Deskripsi View	13
5.3.2 Model: Entity Relationship Diagram	13
5.4 Deployment View (V-04).....	13
5.4.1 Deskripsi View	13
5.4.2 Model: Deployment Diagram.....	14
6. Correspondence dan Consistency.....	15
6.1 Correspondence Rules	15
6.2 Consistency Analysis	15

7. Architecture Decision.....	17
7.1 Decision Log	17
7.2 ADR-01: [Judul Keputusan]	17
Context.....	17
Decision.....	17
Rationale	17
Consequences	17
Alternatives Considered.....	18
8. Architecture Rationale	19
8.1 Traceability Matrix	19
9. Known Issues dan Architecture Risks.....	21
9.1 Known Issues.....	21
9.2 Architecture Risks	21
Lampiran A: Referensi dan Standar	22
A.1 Dokumen Referensi	22
A.2 Standar yang Digunakan.....	22
Lampiran B: Glosarium.....	22
Lampiran C: Pembagian Tugas	22

1. Pendahuluan

1.1 Tujuan Dokumen

Dokumen ini mendeskripsikan arsitektur dari **Sistem Presensi Mahasiswa Berbasis Face Recognition dengan Pendekatan Event-Driven Architecture (EDA)** yang dikembangkan di lingkungan **Universitas Muslim Indonesia (UMI)**.

Tujuan dari dokumen **Architecture Description (AD)** ini adalah untuk:

- Menyediakan **gambaran menyeluruh** tentang struktur, perilaku, dan komponen sistem.
- Mengkomunikasikan **keputusan arsitektur utama** kepada seluruh stakeholder.
- Menjadi acuan bagi tim pengembang dan pemelihara sistem dalam memastikan konsistensi implementasi terhadap rancangan arsitektur.
- Mendukung kegiatan evaluasi kualitas sistem berdasarkan **atribut arsitektur** seperti keamanan, ketersediaan, skalabilitas, dan performa.

1.2 Ruang Lingkup

Ruang lingkup dokumen ini mencakup seluruh aspek arsitektur dari sistem presensi mahasiswa yang memanfaatkan teknologi **Face Recognition** dan diimplementasikan dengan pendekatan **Event-Driven Architecture (EDA)**.

Sistem yang dimaksud mencakup komponen-komponen utama berikut:

- **Face Recognition Module**: menangkap dan memverifikasi wajah mahasiswa secara otomatis.
- **Attendance Service**: mencatat kehadiran dan menghasilkan event kehadiran.
- **Event Broker (Kafka/RabbitMQ)**: menyalurkan event antar layanan secara asinkron.
- **Dashboard Service**: menampilkan data kehadiran secara real-time untuk dosen dan admin fakultas.
- **Integration Layer (API Gateway)**: menghubungkan sistem dengan SIAKAD universitas.

Batasan dokumen ini :

- Tidak mencakup manajemen nilai akademik, jadwal kuliah, maupun sistem e-learning.
- Fokus pada **arsitektur sistem presensi mahasiswa** (bukan staf atau dosen).
- Aspek implementasi teknis detail (kode program, konfigurasi server) tidak dibahas secara eksplisit.

1.3 Definisi dan Akronim

Istilah/Akrоним	Definisi
EDA	<i>Event-Driven Architecture</i> — Pendekatan arsitektur yang berfokus pada pemrosesan event secara asinkron dan terdistribusi.
API	<i>Application Programming Interface</i> — Antarmuka untuk komunikasi antar komponen perangkat lunak.
SIAKAD	<i>Sistem Informasi Akademik</i> — Sistem utama kampus yang menyimpan dan mengelola data akademik mahasiswa.
Face Recognition	

	Teknologi biometrik untuk mendeteksi dan mengenali wajah sebagai bentuk autentikasi identitas pengguna.
Smart Campus	Konsep digitalisasi kampus yang mengintegrasikan berbagai sistem akademik dan teknologi cerdas untuk efisiensi operasional.
UML	<i>Unified Modeling Language</i> — Bahasa pemodelan standar untuk menggambarkan desain sistem perangkat lunak.
API Gateway	Komponen arsitektur yang mengelola lalu lintas permintaan antara klien dan layanan backend secara terpusat.

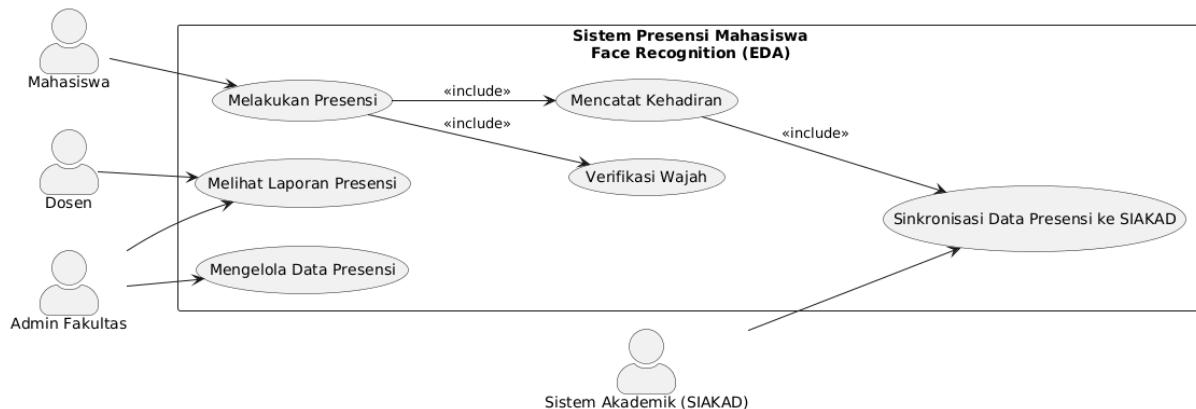
2. System of Interest (SoI)

2.1 Identifikasi Sistem

Nama Sistem	Sistem Presensi Mahasiswa Berbasis Face Recognition dengan Pendekatan Event-Driven Architecture (EDA)
Versi Sistem	1.0
Deskripsi Singkat	Sistem ini berfungsi untuk melakukan pencatatan kehadiran mahasiswa secara otomatis melalui pengenalan wajah (face recognition) dan menyimpan data presensi ke dalam sistem akademik (SIAKAD) secara real-time menggunakan pendekatan arsitektur event-driven.
Domain	Akademik / Smart Campus / Sistem Informasi Pendidikan
Studi Kasus	Universitas Muslim Indonesia (UMI) – Fakultas Ilmu Komputer

2.2 Konteks Sistem

Sistem presensi ini beroperasi di lingkungan kampus Universitas Muslim Indonesia dan berinteraksi dengan beberapa sistem eksternal seperti SIAKAD, sistem autentikasi pengguna (Single Sign-On), dan infrastruktur jaringan kampus.



2.3 Misi dan Tujuan Sistem

Tujuan utama sistem:

- **Meningkatkan efisiensi dan akurasi presensi mahasiswa** dengan teknologi pengenalan wajah otomatis.
- **Mengintegrasikan hasil presensi langsung ke sistem akademik (SIAKAD)** untuk mengurangi kesalahan input manual.
- **Mendukung digitalisasi kampus (Smart Campus)** dengan sistem presensi berbasis data real-time dan event-driven.
- **Mengurangi potensi kecurangan (titip absen)** melalui validasi wajah biometrik yang unik.
- **Menyediakan laporan presensi yang akurat** bagi dosen dan fakultas untuk keperluan akreditasi.

3. Stakeholder dan Concern

3.1 Identifikasi Stakeholder

Stakeholder adalah individu, tim, atau organisasi yang memiliki kepentingan terhadap sistem presensi mahasiswa berbasis Face Recognition dengan pendekatan Event-Driven Architecture (EDA).

ID	Stakeholder	Peran	Deskripsi
SH-01	Mahasiswa	End User	Pengguna utama sistem yang melakukan presensi secara otomatis melalui pengenalan wajah di ruang kelas.
SH-02	Dosen	End User	Menggunakan sistem untuk memantau kehadiran mahasiswa secara real-time dan mengakses laporan presensi
SH-03	Admin Fakultas	System Administrator	Bertanggung jawab terhadap pengelolaan data presensi, validasi hasil presensi, serta monitoring operasional sistem.
SH-04	Tim IT Kampus	System Administrator	Mengelola infrastruktur server, keamanan sistem, integrasi dengan SIAKAD, dan pemeliharaan teknis sistem.
SH-05	Developer	Pengembang	Membangun, mengembangkan, dan memelihara sistem presensi sesuai dengan arsitektur yang telah ditetapkan.
SH-06	Biro Akademik / Wakil Dekan III	Business Owner	Pemilik kepentingan bisnis dan pengambil keputusan terkait kebijakan akademik serta pemanfaatan data presensi untuk evaluasi dan akreditasi.

3.2 Concern (Kepentingan Arsitektural)

Concern adalah kepentingan atau kebutuhan arsitektural yang relevan terhadap satu atau lebih stakeholder dan memengaruhi keputusan desain arsitektur sistem.

ID	Concern	Deskripsi	Stakeholder
C-01	Performance	Sistem harus mampu memproses deteksi wajah dan pencatatan presensi secara real-time dengan waktu respons kurang dari 2 detik.	SH-01, SH-02
C-02	Security	Perlindungan data biometrik wajah mahasiswa melalui enkripsi,	SH-03, SH-04, SH-06

		autentikasi, dan kontrol akses yang ketat.	
C-03	Scalability	Sistem harus mampu menangani peningkatan jumlah mahasiswa, kelas, dan event presensi tanpa penurunan performa.	SH-03, SH-04, SH-05
C-04	Maintainability	Kemudahan modifikasi dan pemeliharaan	SH-03
C-05	Usability	Kemudahan penggunaan sistem	SH-01
C-06	Availability	Sistem harus tersedia selama jam perkuliahan dengan tingkat ketersediaan tinggi (uptime minimal 99.9%).	SH-01, SH-02, SH-03
C-07	Integration	Sistem harus terintegrasi dengan Sistem Informasi Akademik (SIAKAD) untuk sinkronisasi data presensi secara otomatis.	SH-03, SH-04, SH-06

4. Architecture Viewpoint

Architecture Viewpoint mendefinisikan konvensi untuk konstruksi, interpretasi, dan penggunaan *architecture view*.

Setiap viewpoint dirancang untuk menangani concern tertentu dari stakeholder yang relevan, sehingga arsitektur sistem dapat dipahami secara konsisten dari berbagai sudut pandang.

4.1 Viewpoint Catalog

ID	Viewpoint	Concern yang Ditangani	Stakeholder	View
VP-01	Context	Batasan sistem, aktor eksternal, dan antarmuka	Semua Stakeholder	V-01
VP-02	Functional	Fungsionalitas sistem dan alur kerja utama	SH-01, SH-02, SH-05	V-02
VP-03	Information	Struktur data, penyimpanan, dan integritas informasi	SH-03, SH-04, SH-06	V-03
VP-04	Deployment	Infrastruktur, distribusi komponen, dan skalabilitas	SH-03, SH-04, SH-05	V-04

4.2 Viewpoint: Context (VP-01)

Nama	Context Viewpoint
Concern	Menjelaskan batasan sistem, aktor eksternal, serta interaksi sistem dengan lingkungan di luar batas sistem.
Stakeholder	Semua stakeholder (Mahasiswa, Dosen, Admin Fakultas, Tim IT, Business Owner).
Model Kind	Context Diagram, System Boundary Diagram
Notasi	UML Use Case Diagram, Block Diagram

4.3 Viewpoint: Functional (VP-02)

Nama	Functional Viewpoint
Concern	Fungsionalitas utama sistem, kapabilitas layanan, dan alur kerja presensi mahasiswa.
Stakeholder	End User (Mahasiswa, Dosen) dan Developer.
Model Kind	Component Diagram, Activity Diagram
Notasi	UML Component Diagram, BPMN

4.4 Viewpoint: Information (VP-03)

Nama	Information Viewpoint
Concern	Struktur data, penyimpanan informasi presensi, dan integritas data biometrik mahasiswa.
Stakeholder	Developer dan Business Owner.
Model Kind	Entity Relationship Diagram, Class Diagram
Notasi	UML Class Diagram, ERD

4.5 Viewpoint: Deployment (VP-04)

Nama	Deployment Viewpoint
Concern	Infrastruktur sistem, node fisik/logis, distribusi komponen, serta dukungan terhadap skalabilitas dan ketersediaan sistem.
Stakeholder	System Administrator, Developer
Model Kind	Deployment Diagram, Infrastructure Diagram
Notasi	UML Deployment Diagram, Cloud Architecture Diagram

5. Architecture View

Architecture View merupakan representasi arsitektur sistem dari perspektif viewpoint tertentu. Setiap view pada bagian ini berfungsi untuk menjelaskan aspek arsitektur yang berbeda sesuai dengan concern stakeholder yang telah diidentifikasi sebelumnya.

5.1 Context View (V-01)

5.1.1 Deskripsi View

Context View menggambarkan bagaimana sistem presensi mahasiswa berinteraksi dengan lingkungan eksternalnya, termasuk aktor pengguna dan sistem eksternal. View ini bertujuan untuk menunjukkan batasan sistem (*system boundary*), aktor yang terlibat, serta alur interaksi utama antara sistem presensi dan pihak luar seperti mahasiswa, dosen, admin fakultas, serta Sistem Informasi Akademik (SIAKAD).

5.1.2 Model: Context Diagram

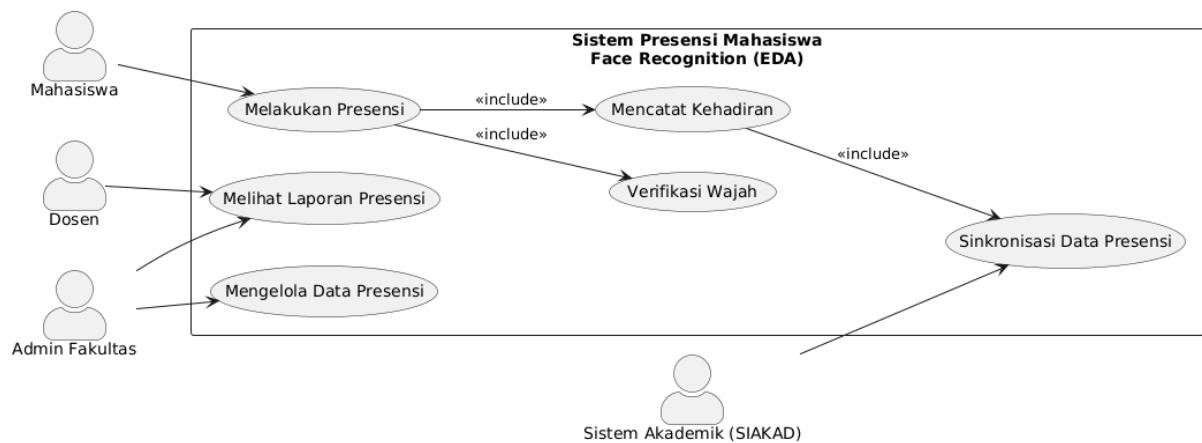


Diagram ini menunjukkan bahwa mahasiswa melakukan presensi melalui sistem, yang secara otomatis memicu proses verifikasi wajah dan pencatatan kehadiran. Data kehadiran selanjutnya disinkronkan ke SIAKAD. Dosen dan admin fakultas memiliki akses untuk melihat laporan presensi, sementara admin juga dapat mengelola data presensi.

5.2 Functional View (V-02)

5.2.1 Deskripsi View

Functional View menjelaskan fungsionalitas utama dan kapabilitas sistem presensi mahasiswa. View ini berfokus pada pembagian komponen fungsional dan bagaimana masing-masing komponen berinteraksi untuk mendukung proses presensi berbasis Face Recognition dengan pendekatan Event-Driven Architecture.

5.2.2 Model: Component Diagram

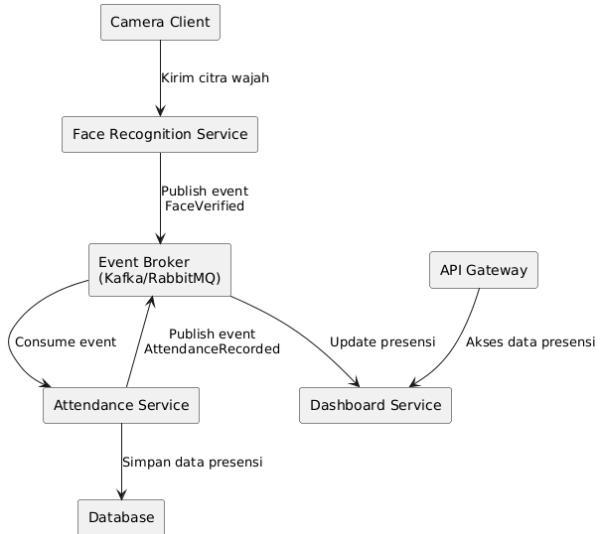


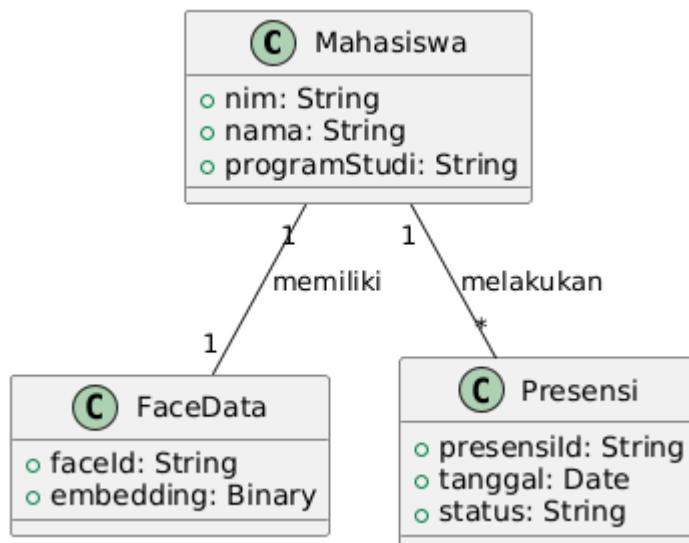
Diagram komponen ini menunjukkan pemisahan fungsi sistem ke dalam komponen mandiri. Event Broker menjadi pusat komunikasi asinkron antar komponen, memastikan loose coupling dan mendukung pemrosesan real-time.

5.3 Information View (V-03)

5.3.1 Deskripsi View

Information View menggambarkan struktur data utama dan aliran informasi dalam sistem presensi. View ini berfokus pada entitas data yang digunakan untuk menyimpan informasi mahasiswa, data wajah, dan catatan presensi, serta hubungan antar entitas tersebut.

5.3.2 Model: Class Diagram



Setiap mahasiswa memiliki satu data wajah (FaceData) dan dapat memiliki banyak catatan presensi. Struktur ini mendukung integritas data biometrik dan riwayat kehadiran mahasiswa.

5.4 Deployment View (V-04)

5.4.1 Deskripsi View

Deployment View menjelaskan bagaimana sistem presensi dideploy pada infrastruktur fisik dan logis.

View ini mencakup distribusi komponen pada edge device (kamera), server aplikasi, event broker, dan database yang berada di lingkungan cloud atau data center kampus.

5.4.2 Model: Deployment Diagram

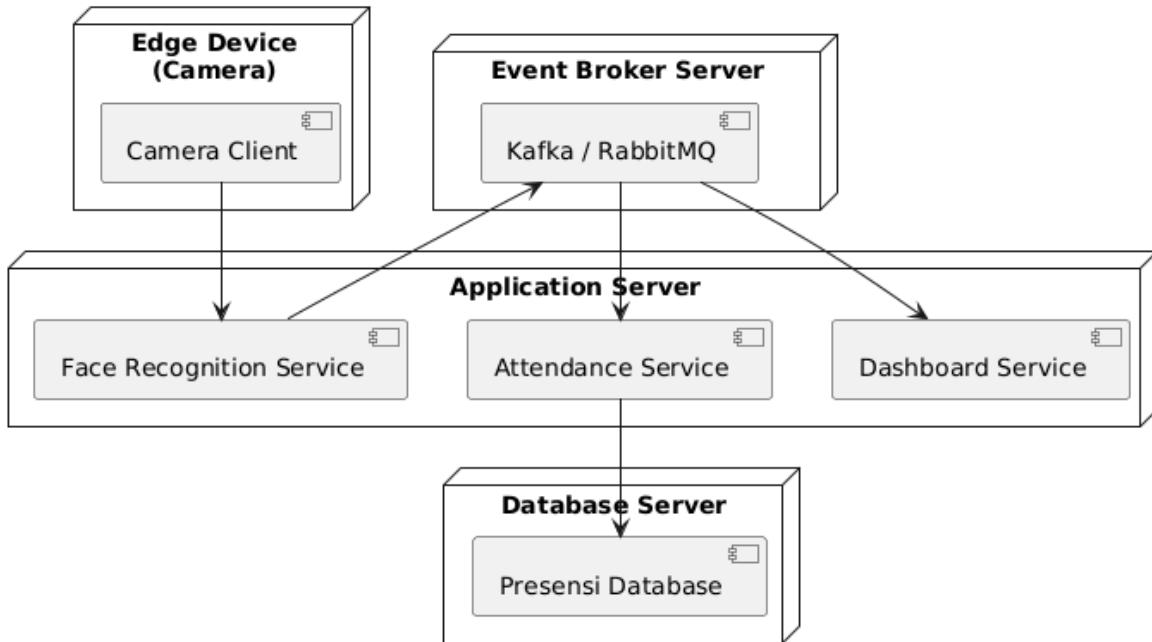


Diagram deployment menunjukkan pemisahan antara edge device, application server, event broker, dan database server.

Pendekatan ini mendukung skalabilitas, ketersediaan tinggi, dan pemrosesan event secara asinkron.

6. Correspondence and Consistency

6.1 Correspondence Rules

Correspondence rules mendefinisikan hubungan dan batasan (*constraint*) antar elemen arsitektur yang terdapat pada view yang berbeda, guna memastikan bahwa seluruh view saling terhubung dan konsisten satu sama lain.

ID	View Source	View Target	Rule Description
CR-01	Functional View	Information View	Setiap komponen fungsional yang memproses atau menghasilkan data harus memiliki entitas data yang didefinisikan pada Information View.
CR-02	Functional View	Deployment View	Setiap komponen pada Functional View harus dipetakan dan di-deploy pada minimal satu node di Deployment View.
CR-03	Context View	Functional View	Setiap aktor eksternal pada Context View harus memiliki interaksi yang direpresentasikan oleh satu atau lebih komponen pada Functional View.
CR-04	Information View	Deployment View	Setiap entitas data utama harus memiliki lokasi penyimpanan yang jelas pada Deployment View (misalnya database server).
CR-05	Context View	Deployment View	Setiap sistem eksternal yang berinteraksi dengan sistem harus memiliki jalur koneksi yang jelas pada Deployment View.

6.2 Consistency Analysis

Bagian ini mendokumentasikan hasil analisis konsistensi antar architecture view dan model yang telah disusun.

Konsistensi Context View – Functional View

Semua aktor eksternal yang didefinisikan pada Context View, yaitu Mahasiswa, Dosen, Admin Fakultas, dan Sistem Akademik (SIAKAD), telah memiliki interaksi yang jelas dengan komponen sistem pada Functional View.

Mahasiswa berinteraksi melalui proses *Melakukan Presensi* yang direpresentasikan oleh komponen *Camera Client* dan *Face Recognition Service*.

Dosen dan Admin Fakultas berinteraksi melalui *Dashboard Service* untuk melihat laporan presensi.

Konsistensi Functional View – Information View

Seluruh komponen pada Functional View yang memproses data telah memiliki entitas data terkait pada Information View.

Komponen *Face Recognition Service* berhubungan dengan entitas *FaceData*, sedangkan *Attendance Service* berhubungan dengan entitas *Presensi* dan *Mahasiswa*.

Tidak ditemukan komponen yang memproses data tanpa dukungan struktur data yang sesuai.

Konsistensi Functional View – Deployment View

Semua komponen yang didefinisikan pada Functional View telah dipetakan ke node yang sesuai

pada Deployment View.

Camera Client dideploy pada *Edge Device*, sementara *Face Recognition Service*, *Attendance Service*, dan *Dashboard Service* dideploy pada *Application Server*.

Event Broker dan *Database* masing-masing memiliki node tersendiri pada Deployment View.

Konsistensi Information View – Deployment View

Entitas data utama seperti *Mahasiswa*, *FaceData*, dan *Presensi* telah dipetakan ke *Database Server* pada Deployment View.

Hal ini memastikan bahwa seluruh data penting memiliki lokasi penyimpanan yang jelas dan terkelola.

Inkonsistensi yang Ditemukan

Pada versi dokumen ini, tidak ditemukan inkonsistensi signifikan antar view dan model. Seluruh elemen arsitektur telah terpetakan secara konsisten sesuai correspondence rules yang ditetapkan.

Status

Konsistensi antar view telah tervalidasi dan dinyatakan **konsisten**. Tidak diperlukan perubahan arsitektur pada tahap ini.

7. Architecture Decision

Architecture Decision Record (ADR) mendokumentasikan keputusan arsitektural yang signifikan, termasuk konteks, alasan pengambilan keputusan, serta konsekuensi yang ditimbulkan. ADR membantu memastikan bahwa setiap keputusan arsitektur dapat ditelusuri dan dipahami oleh seluruh stakeholder.

7.1 Decision Log

ID	Keputusan	Status	Tanggal	ADR
AD-01	Menggunakan Event-Driven Architecture (EDA)	Approved	13/12/2025	ADR-01
AD-02	Menggunakan Face Recognition sebagai metode presensi utama	Approved	13/12/2025	ADR-02
[AD-XX]	Menggunakan Event Broker (Kafka/RabbitMQ) untuk komunikasi asinkron	Approved	13/12/2025	

7.2 ADR-01: [Judul Keputusan]

Context

Sistem presensi mahasiswa berbasis Face Recognition harus mampu memproses banyak aktivitas secara real-time, seperti deteksi wajah, verifikasi identitas, pencatatan kehadiran, dan pembaruan dashboard secara bersamaan.

Pendekatan arsitektur sinkron atau monolitik berpotensi menimbulkan bottleneck, meningkatkan latensi, serta menyulitkan pengembangan dan skalabilitas sistem di masa depan.

Decision

Sistem presensi mahasiswa dirancang menggunakan **Event-Driven Architecture (EDA)**, di mana setiap aktivitas utama direpresentasikan sebagai event dan diproses secara asinkron melalui event broker.

Rationale

Pendekatan Event-Driven Architecture dipilih karena:

- Mendukung pemrosesan **real-time** dan **asinkron**, sesuai dengan kebutuhan sistem presensi berbasis Face Recognition.
- Mengurangi ketergantungan langsung antar komponen (*loose coupling*).
- Mempermudah integrasi dengan sistem eksternal seperti SIAKAD.
- Mendukung skalabilitas sistem seiring pertumbuhan jumlah mahasiswa dan kelas.

Consequences

- **Positif:**
 - Sistem mampu menangani banyak event presensi secara bersamaan dengan latensi rendah.
 - Arsitektur lebih modular dan mudah dikembangkan di masa depan.
 - Kegagalan pada satu komponen tidak langsung menghentikan keseluruhan sistem.

- **Negatif:**
 - Kompleksitas sistem meningkat karena adanya event broker dan mekanisme event handling.
 - Proses debugging menjadi lebih sulit dibandingkan arsitektur sinkron.

Alternatives Considered

- **Monolithic Architecture**

Ditolak karena kurang scalable dan berpotensi menjadi bottleneck pada beban tinggi.
- **Layered Architecture**

Ditolak karena terlalu banyak lapisan yang dapat meningkatkan latensi pada sistem real-time.
- **Microservices Architecture**

Ditunda untuk tahap awal karena kompleksitas operasional dan biaya yang lebih tinggi, meskipun memiliki skalabilitas yang baik.

8. Architecture Rationale

Architecture Rationale mendokumentasikan alasan dan justifikasi di balik keputusan arsitektural yang diambil.

Bagian ini menghubungkan *concern stakeholder* dengan *viewpoint*, *architecture view*, dan *architecture decision* untuk memastikan bahwa setiap keputusan arsitektur dapat ditelusuri dan dipertanggungjawabkan.

8.1 Traceability Matrix

Concern	Viewpoint	View	Decision	Rationale
C-01 Performance	VP-02 Functional, VP-04 Deployment	V-02, V-04	AD-01	Event-Driven Architecture memungkinkan pemrosesan asinkron dan paralel sehingga waktu respon presensi dapat dijaga tetap rendah.
C-02 Security	VP-03 Information, VP-04 Deployment	V-03, V-04	AD-02	Penggunaan Face Recognition sebagai metode presensi meningkatkan keamanan autentikasi dan mengurangi kecurangan presensi.
C-03 Scalability	VP-04 Deployment	V-04	AD-01, AD-03	Event Broker mendukung skalabilitas horizontal dengan memisahkan producer dan consumer event.
C-04 Maintainability	VP-02 Functional	V-02	AD-01	Pendekatan event-driven menghasilkan loose coupling antar komponen sehingga sistem lebih mudah dipelihara dan dikembangkan.
C-05 Usability	VP-01 Context, VP-02 Functional	V-01, V-02	AD-02	Face Recognition memberikan pengalaman presensi yang cepat dan mudah tanpa interaksi manual.
C-06 Availability	VP-04 Deployment	V-04	AD-01, AD-03	Arsitektur berbasis event dan infrastruktur terdistribusi meningkatkan

				toleransi kesalahan dan ketersediaan sistem.
C-07 Integration	VP-01 Context, VP-04 Deployment	V-01, V-04	AD-01, AD-03	Event-Driven Architecture memudahkan integrasi sistem presensi dengan SIAKAD melalui mekanisme event dan API.

9. Known Issues dan Architecture Risks

9.1 Known Issues

Known Issues adalah permasalahan yang telah teridentifikasi pada tahap desain atau implementasi awal dan perlu ditangani agar tidak berdampak pada kualitas sistem.

ID	Issue	Impact	Mitigation
ISS-01	Akurasi Face Recognition menurun pada kondisi pencahayaan rendah	Presensi mahasiswa dapat gagal atau tidak terdeteksi	Menambahkan preprocessing citra (normalisasi cahaya) dan peningkatan kualitas kamera
ISS-02	Latensi jaringan saat pengiriman event presensi	Keterlambatan pencatatan presensi dan pembaruan dashboard	Menggunakan event broker dengan mekanisme retry dan buffering
ISS-03	Sinkronisasi data dengan SIAKAD tidak real-time saat beban tinggi	Data presensi di sistem akademik tidak langsung terbarui	Implementasi mekanisme asynchronous event queue dan scheduled reconciliation
ISS-04	Kesulitan debugging alur event yang kompleks	Sulit melacak kesalahan sistem	Menambahkan centralized logging dan event tracing
ISS-05	Keterbatasan dataset wajah awal	Akurasi sistem belum optimal pada tahap awal	Melakukan pengayaan dataset dan retraining model secara berkala

9.2 Architecture Risks

Architecture Risks adalah risiko potensial yang dapat memengaruhi keberhasilan sistem dalam jangka panjang jika tidak dimitigasi dengan baik.

ID	Risk	Probability	Impact	Mitigation Strategy
RSK-01	Kebocoran data biometrik mahasiswa	Medium	High	Menerapkan enkripsi data (AES-256), kontrol akses berbasis role, dan audit keamanan berkala
RSK-02	Kegagalan event broker (Kafka/RabbitMQ)	Low	High	Menggunakan cluster broker, failover, dan persistent message queue
RSK-03	Lonjakan jumlah event presensi secara bersamaan	Medium	Medium	Menerapkan autoscaling consumer dan load balancing
RSK-04	Ketergantungan pada infrastruktur jaringan kampus	Medium	Medium	Menggunakan fallback offline buffer pada edge device
RSK-05	Kompleksitas arsitektur Event-Driven	Medium	Low	Dokumentasi arsitektur yang lengkap dan standar pengembangan yang jelas

Lampiran A: Referensi dan Standar

A.1 Dokumen Referensi

1. ISO/IEC/IEEE 42010:2022 - Systems and software engineering - Architecture description
2. Gregor Hohpe and Bobby Woolf, *Enterprise Integration Patterns*, Addison-Wesley, 2004.
3. Martin Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002.
4. C. Fernando, *Solution Architecture Patterns for Enterprise*, Apress, 2022.
5. K. Alhanaee et al., "Face Recognition Smart Attendance System using Deep Transfer Learning," *Procedia Computer Science*, 2021.
6. H. Al Akbar et al., "Guidance in Designing a Smart Campus: A Systematic Literature Review," *Procedia Computer Science*, 2023.

A.2 Standar yang Digunakan

- UML 2.5 - Unified Modeling Language
- ISO/IEC/IEEE 42010:2022 – Architecture Description
- IEEE 1471 – Recommended Practice for Architectural Description
- REST Architectural Style (untuk integrasi API)
- Event-Driven Architecture (EDA) Pattern

Lampiran B: Glosarium

Istilah	Definisi
Architecture	Konsep atau properti fundamental suatu sistem dalam lingkungannya yang terwujud dalam elemen, hubungan, dan prinsip desainnya.
Architecture Description	Produk kerja yang digunakan untuk mengekspresikan suatu arsitektur sistem.
Architecture View	Representasi arsitektur sistem dari sudut pandang tertentu.
Architecture Viewpoint	Konvensi untuk membangun dan menafsirkan architecture view.
Concern	Kepentingan stakeholder terhadap sistem yang memengaruhi keputusan arsitektur.
Model	Representasi elemen sistem yang digunakan dalam architecture view.
Stakeholder	Individu, tim, atau organisasi yang memiliki kepentingan terhadap sistem.
Event	Kejadian penting dalam sistem yang memicu proses lanjutan.
Event-Driven Architecture (EDA)	Pendekatan arsitektur yang berfokus pada pemrosesan sistem berdasarkan event secara asinkron.
Face Recognition	Teknologi biometrik untuk mengenali identitas berdasarkan ciri wajah.
Event Broker	Komponen perantara yang mendistribusikan event antar layanan.
Smart Campus	Konsep digitalisasi kampus berbasis integrasi sistem dan teknologi cerdas.

Lampiran C: Pembagian Tugas

Stambuk	Nama	Deskripsi Pekerjaan Detail	Status (Selesai, Gagal, Tunda)

13020230309	Hendrawan	Perancangan arsitektur sistem, penyusunan dokumen Architecture Description (ISO 42010), pembuatan diagram UML, analisis EDA dan Face Recognition	Selesai

Lampiran D: Link Dokumentasi Foto/Video

<https://github.com/hendrawanwawan/smart-campus-face-attendance.git>