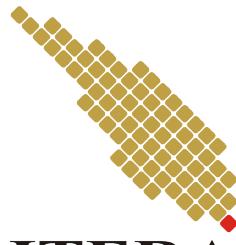


**SISTEM CERDAS PELACAKAN PELANGGARAN HELM BERBASIS YOLOV8
DAN BYTETRACK UNTUK MENDUKUNG ANALITIK DATA LALU LINTAS DI
INSTITUT TEKNOLOGI SUMATERA**

PROPOSAL SKRIPSI

**EGGI SATRIA
122450032**



**PROGRAM STUDI SAINS DATA
FAKULTAS SAINS
INSTITUT TEKNOLOGI SUMATERA
2025**

LEMBAR PENGESAHAN

Tugas Akhir dengan judul “**Sistem Cerdas Pelacakan Pelanggaran Helm Berbasis YOLOv8 dan ByteTrack untuk Mendukung Analitik Data Lalu Lintas di Institut Teknologi Sumatera**” adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

Lampung Selatan, 25 November 2025
Penulis,

Eggi Satria
NIM. 122450032



Pembimbing I

Luluk Muthoharoh, M.Si.
NIP. 199008172020031003

Pembimbing II

Ardika Satria, M.Si.
NIP. 199711102024061001

Disahkan oleh,
Koordinator Program Studi Sains Data
Fakultas Sains
Institut Teknologi Sumatera

Tirta Setiawan, S.Pd., M.Si.
NIP. 199008222022031003

Seminar Proposal:
Penguji I : Ahmad Luky Ramdani, S.Kom., M.Kom
Penguji II :

Daftar Isi

LEMBAR PENGESAHAN	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Landasan Teori	7
2.2.1 Big Data	7
2.2.2 Arsitektur Streaming: <i>Kappa Architecture</i>	7
2.2.2.1 Apache Kafka	8
2.2.2.2 Apache Spark	9
2.2.3 YOLO (<i>You Only Look Once</i>)	10
2.2.4 ByteTrack: Multi-Object Tracking Berbasis Asosiasi Dua Tahap	12
2.2.4.1 Asosiasi Dua Tahap	13
2.2.5 Data Warehouse	13
2.2.6 Data Mart	14
2.2.6.1 Star Schema	14
2.2.7 Grafana	15
2.2.8 Evaluasi Deteksi Objek	15
2.2.8.1 Klasifikasi Deteksi: TP, FP, FN, dan TN	16
2.2.8.2 Average Precision (AP) dan Mean Average Precision (mAP)	17
2.2.9 Evaluasi Sistem Streaming	18
BAB III METODOLOGI PENELITIAN	20
3.1 Deskripsi Data	20
3.1.1 Jenis dan Sumber Data	20
3.1.2 Format dan Karakteristik Data	20

3.2	Arsitektur Sistem	21
3.2.1	Data Source Layer	22
3.2.2	Ingestion & Edge Intelligence Layer	22
3.2.3	Buffering Layer	23
3.2.4	Stream Processing Layer	23
3.2.5	Persistence Layer	23
3.2.6	Presentation Layer	23
3.3	Perancangan Model YOLOv8 dan ByteTrack	23
3.3.1	Preprocessing dan Anotasi Data	23
3.3.2	Pelatihan Model YOLOv8	23
3.4	Logika Deteksi Pelanggaran Helm	24
3.5	Rancangan Basis Data dan Skema Data Mart	25
3.5.1	Pendekatan Data-Driven Data Mart	25
3.5.2	Desain Skema Bintang (Star Schema)	25
3.6	Rancangan Dashboard Analitik	25
3.7	Alur Penelitian	26
3.7.1	Tahap 1: Persiapan dan Pengumpulan Data	26
3.7.2	Tahap 2: Pengembangan dan Pelatihan Model	26
3.7.3	Tahap 3: Pengembangan dan Integrasi Pipeline	27
3.7.4	Tahap 4: Evaluasi dan Analisis	27
	DAFTAR PUSTAKA	29

Daftar Gambar

Gambar 2.1	Alur Data Arsitektur Kappa [17]	8
Gambar 2.2	Arsitektur komponen inti Apache Kafka [19]	9
Gambar 2.3	Ilustrasi Mekanisme Prediksi Bounding Box pada YOLO[22]	10
Gambar 2.4	Arsitektur YOLO sebagai <i>single-stage object detector</i> [22].	11
Gambar 2.5	Pipeline ByteTrack [23]. Arsitektur ini mengintegrasikan tahap deteksi dan asosiasi secara bertingkat untuk mempertahankan identitas objek pada video <i>real-time</i> . .	12
Gambar 2.6	Ilustrasi proses asosiasi dua tahap pada ByteTrack. Tahap pertama mengasosiasikan deteksi skor tinggi, kemudian tahap kedua memulihkan identitas objek melalui asosiasi dengan deteksi skor rendah[23].	13
Gambar 2.7	Ilustrasi kategori hasil deteksi objek pada sistem <i>object detection</i> : True Positive (TP), False Positive (FP), dan False Negative (FN) [29].	16
Gambar 2.8	Hubungan antara Precision, Recall, dan Intersection over Union (IoU) pada sistem deteksi objek. IoU dihitung berdasarkan area tumpang tindih antara kotak prediksi dan anotasi <i>ground truth</i> [30].	17
Gambar 3.1	Arsitektur Sistem Streaming-First Pelacakan Pelanggaran Helm	22
Gambar 3.2	Desain Skema Bintang Data Mart Pelanggaran Helm . .	25
Gambar 3.3	Diagram Alur Penelitian Sistem Pelacakan Pelanggaran Helm	28

Daftar Tabel

Tabel 2.1	Penelitian Terdahulu	5
-----------	--------------------------------	---

BAB I

PENDAHULUAN

1.1 Latar Belakang

Institut Teknologi Sumatera (ITERA) memiliki visi strategis untuk menjadi perguruan tinggi teknologi terkemuka yang mengedepankan lingkungan kampus yang nyaman, aman, dan berwawasan lingkungan [1]. Visi ini diwujudkan melalui berbagai inisiatif, salah satunya adalah pengembangan konsep *forest campus*, yang menekankan pada keberlanjutan, ketertiban, dan kualitas hidup yang tinggi bagi seluruh sivitas akademika. Konsep ini tidak hanya bertujuan memperindah lanskap kampus, melainkan juga mengusung paradigma pendidikan berbasis lingkungan, dengan menekankan keterpaduan antara ruang hijau, infrastruktur teknologi, dan efisiensi sumber daya. *Forest campus* menjadi simbol integrasi antara fungsi akademik, keberlanjutan ekologis, dan kenyamanan hidup kampus. Konsep ini merupakan bagian dari tren global dalam pengembangan institusi pendidikan tinggi modern yang mengintegrasikan prinsip keberlanjutan dengan pemanfaatan teknologi informasi dan komunikasi. konsep *forest campus* dan *smart campus*, sistem digital, Internet of Things (IoT), dan kecerdasan buatan diintegrasikan dengan tata ruang kampus guna meningkatkan efisiensi operasional, pengalaman pengguna, serta keselamatan [2]. Hal ini pula yang menjadikan ITERA memiliki daya tarik tersendiri di kalangan calon mahasiswa. Lingkungan hijau, atmosfer belajar yang tenang, serta visi jangka panjang menuju kampus berkelanjutan menjadikan ITERA sebagai salah satu pilihan utama perguruan tinggi di wilayah Sumatera dan sekitarnya. Kombinasi nilai lingkungan dan teknologi ini memperkuat citra institusi dan memicu pertumbuhan pesat dalam jumlah mahasiswa baru dari tahun ke tahun.

Daya tarik tersebut berimplikasi langsung pada peningkatan jumlah mahasiswa baru yang konsisten setiap tahunnya. Data akademik mencatat penerimaan mahasiswa mencapai 4.434 orang pada tahun 2023, 4.477 orang pada tahun 2024, dan 4.929 mahasiswa pada tahun 2025, sehingga total populasi mahasiswa aktif saat ini telah melampaui 20.000 jiwa [3]. Pertumbuhan demografis ini secara langsung berkorelasi dengan peningkatan penggunaan kendaraan pribadi, khususnya sepeda motor [4]. Berbagai studi akademik telah mengonfirmasi bahwa peningkatan populasi mahasiswa di sebuah perguruan tinggi secara inheren akan meningkatkan volume transportasi dan penggunaan energi, terutama jika

ketersediaan transportasi publik belum memadai [5]. Institut Teknologi Sumatera memulai pengembangan transportasi umum seperti Smart BRT [6]. Kapasitas serta jangkauan transportasi tersebut masih terbatas. Kendaraan pribadi menjadi opsi utama bagi sebagian besar mahasiswa. Peningkatan volume kendaraan di lingkungan kampus secara inheren meningkatkan probabilitas terjadinya pelanggaran lalu lintas, dengan salah satu yang paling krusial dan berisiko tinggi adalah penggunaan motor tanpa memakai helm. Tingkat kepatuhan penggunaan helm di kalangan mahasiswa merupakan isu yang menjadi perhatian nasional. Sebuah studi lintas-seksional yang dilakukan di lingkungan universitas menunjukkan bahwa persentase signifikan, mencapai 40.8% dari mahasiswa pengendara sepeda motor, tidak menggunakan helm secara konsisten [7]. Berbagai penelitian lebih lanjut mengidentifikasi alasan di balik rendahnya kepatuhan ini, yang mencakup faktor-faktor seperti ketidaknyamanan fisik, perasaan gerah dan sesak, serta persepsi bahwa perjalanan yang ditempuh berjarak pendek sehingga tidak memerlukan proteksi [8]. Menyadari tingginya risiko kecelakaan kendaraan bermotor di lingkungan perguruan tinggi, Institut Teknologi Sumatera telah menetapkan kebijakan resmi yang mengatur perilaku berkendara secara aman dan tertib di area kampus. Komitmen tersebut tertuang dalam Peraturan Rektor Institut Teknologi Sumatera Nomor 1 Tahun 2017 tentang Disiplin Mahasiswa, yang menjadi landasan utama dalam penegakan aturan keselamatan berkendara, termasuk kewajiban mematuhi ketentuan penggunaan helm dan kelengkapan keselamatan lainnya [9]. Namun, tantangan utama terletak pada sistem penegakan aturan. Saat ini, ITERA masih mengandalkan pengawasan manual oleh satuan pengamanan (Satpam), sebuah metode yang memiliki keterbatasan inheren: tidak dapat diskalakan, tidak mampu memberikan pengawasan 24/7 di seluruh area kampus, dan rentan terhadap kesalahan manusia. Lebih krusial lagi, metode ini gagal menghasilkan data kuantitatif yang objektif. Tanpa data akurat mengenai lokasi rawan pelanggaran (*hotspot*), waktu puncak, dan frekuensi, unit terkait seperti Unit Pelaksana Teknis Keselamatan, Kesehatan Kerja, dan Lingkungan (UPT K3L) ITERA tidak dapat merumuskan kebijakan intervensi yang efektif dan berbasis bukti.

Penelitian ini mengusulkan pengembangan sistem cerdas berbasis visi komputer untuk mendeteksi dan menganalisis pelanggaran penggunaan helm secara otomatis di lingkungan kampus Institut Teknologi Sumatera dengan mengintegrasikan model deteksi objek YOLOv8 dan algoritma pelacakan multi-objek ByteTrack. YOLOv8 digunakan untuk menghasilkan deteksi objek yang akurat dan berkecepatan real-time, sementara ByteTrack menjaga konsistensi identitas

pengendara lintas frame meskipun terjadi oklusi atau perubahan kondisi visual. Berbeda dari penelitian sebelumnya yang umumnya berhenti pada tahap deteksi atau pelacakan, penelitian ini memposisikan hasil visi komputer sebagai aliran peristiwa (event stream) yang kaya dimensi spasial dan temporal, di mana setiap pelanggaran dicatat sebagai metadata terstruktur dan disimpan dalam data mart tematik. Pendekatan ini dipilih karena karakteristik data pelanggaran yang bersifat historis, non-volatile, dan berorientasi analitik tidak dapat dikelola secara optimal menggunakan basis data relasional konvensional yang berfokus pada transaksi operasional. Dengan demikian, sistem yang diusulkan tidak hanya berfungsi sebagai alat deteksi otomatis, tetapi juga sebagai fondasi analitik jangka panjang untuk mengidentifikasi pola pelanggaran, menentukan lokasi dan waktu rawan, serta mendukung pengambilan keputusan keselamatan berbasis data dalam kerangka implementasi Smart Campus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan utama dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang dan melatih sebuah model deteksi objek berbasis arsitektur YOLOv8 yang mampu secara andal mendeteksi pengendara sepeda motor yang mengenakan dan tidak mengenakan helm pada data video CCTV kampus?
2. Bagaimana mengintegrasikan model deteksi YOLOv8 dengan algoritma pelacakan multi-objek ByteTrack dalam sebuah sistem pemrosesan aliran video real-time, sehingga hasil inferensi dapat direpresentasikan sebagai aliran peristiwa (event stream) yang memiliki konteks spasial dan temporal untuk mendukung analitik lalu lintas?
3. Bagaimana merancang sebuah skema data mart analitik untuk mengelola dan menyimpan peristiwa pelanggaran helm yang dihasilkan oleh sistem real-time secara sistematis, sehingga memungkinkan analisis pola lalu lintas, tren keselamatan, dan pengambilan keputusan berbasis data di lingkungan Institut Teknologi Sumatera?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai melalui penelitian ini adalah sebagai berikut:

1. Mengembangkan dan melatih sebuah model deteksi objek berbasis arsitektur YOLOv8 yang dioptimalkan untuk mendeteksi pengendara sepeda motor

- yang mengenakan dan tidak mengenakan helm pada lingkungan kampus.
2. Mengimplementasikan integrasi model YOLOv8 dan algoritma ByteTrack dalam sebuah sistem pemrosesan video real-time untuk menghasilkan aliran peristiwa pelanggaran helm yang memiliki identitas objek serta konteks spasial dan temporal.
 3. Merancang dan mengimplementasikan sebuah data mart analitik sebagai bagian dari arsitektur data warehouse untuk menyimpan dan mengelola hasil deteksi pelanggaran helm secara historis, sehingga mendukung analisis pola lalu lintas dan pengambilan keputusan keselamatan berbasis data di Institut Teknologi Sumatera.

1.4 Batasan Penelitian

Untuk memastikan penelitian ini tetap fokus dan dapat diselesaikan dalam kurun waktu yang telah ditetapkan, maka ruang lingkup penelitian dibatasi sebagai berikut:

1. Penelitian ini secara khusus difokuskan pada deteksi dan analisis pelanggaran penggunaan helm oleh pengendara sepeda motor. Bentuk pelanggaran lalu lintas lainnya, seperti pelanggaran kecepatan, parkir liar, atau pelanggaran rambu lalu lintas, tidak termasuk dalam cakupan penelitian.
2. Sistem yang dikembangkan memanfaatkan data video kamera CCTV statis yang telah terpasang di beberapa lokasi terpilih di lingkungan kampus Institut Teknologi Sumatera.
3. Penelitian ini berfokus pada analisis berbasis data historis dan agregatif yang dihasilkan dari aliran peristiwa pelanggaran helm. Sistem tidak dirancang untuk melakukan prediksi kejadian di masa depan, pemberian rekomendasi kebijakan secara otomatis, maupun intervensi real-time terhadap pelanggaran yang terdeteksi.
4. Model YOLOv8 digunakan sebagai komponen pendukung dalam sistem untuk mengekstraksi informasi visual dari data video. Penelitian ini tidak membahas secara rinci modifikasi arsitektur internal YOLOv8, perbandingan dengan algoritma deteksi objek lain, maupun optimasi tingkat algoritmik.
5. Pengumpulan dan pengujian data dilakukan dalam rentang waktu tertentu, yaitu pada periode 2 Februari- 6 Februari.
6. Penelitian ini tidak melakukan perbandingan eksperimental antara arsitektur streaming dan arsitektur non-streaming, karena fokus penelitian adalah pada perancangan dan evaluasi sistem analitik end-to-end berbasis event, bukan pada studi komparatif arsitektur sistem.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Berbeda dengan lima penelitian pada Tabel 2.1, penelitian ini berfokus pada pengembangan sistem end-to-end yang mengintegrasikan deteksi, pelacakan, dan evaluasi kinerja simpang secara *real-time* menggunakan model YOLOv8 dan arsitektur big data streaming. Pendekatan ini melampaui studi terdahulu yang hanya berfokus pada deteksi atau pelacakan, dengan menghadirkan sistem otomatis yang langsung menghasilkan analisis lalu lintas adaptif dan terukur.

Tabel 2.1 Penelitian Terdahulu

Peneliti & Judul	Metodologi	Data	Hasil Utama dan Analisis
Figo Agil Alunjati [10] <i>Architecture of Four-Arm Intersection Performance Evaluation System Based on Turn Movement Count and Omnidirectional Camera.(2025)</i>	Mengembangkan sistem evaluasi simpang berbasis video <i>analytics</i> dengan kamera omnidirectional dan model YOLOv5, YOLOv8, serta YOLOv11 untuk menghitung pergerakan kendaraan secara <i>real-time</i> .	CCTV ATCS Sedayu, Yogyakarta (2 hari, siang malam).	Model YOLOv11 batch 32 mencapai mAP 93%, F1 91%, FPS 53. Sistem mampu mendeteksi <i>Turn Movement Count</i> otomatis dan menghasilkan evaluasi kategori E–F. Kamera omnidirectional efektif mengurangi <i>occlusion</i> , meski performa menurun pada kondisi cahaya ekstrem.

Bersambung ke halaman berikutnya

Tabel 2.1 Penelitian Terdahulu

Peneliti & Judul	Metodologi	Data	Hasil Utama dan Analisis
Liu et al.[11] <i>Vehicle Flow Detection dengan YOLOv8n + ByteTrack.(2025)</i>	Menggabungkan MSN-YOLOv8n + SEAM + NWD Loss dengan pelacakan ByteTrack-ReID untuk deteksi kendaraan kecil.	Dataset UA-DETRAC (kendaraan jalan raya).	Akurasi tinggi untuk objek kecil (mAP 62.8%, MOTA 72.16%), stabil untuk pelacakan kendaraan. Terbatas pada dataset tertentu dan belum diuji di streaming real-time.
Aboah et al. [12] <i>Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8.(2023)</i>	Mengembangkan sistem deteksi pelanggaran helm multi-kelas menggunakan YOLOv8 dengan teknik few-shot data sampling untuk mengurangi kebutuhan anotasi.	Dataset AI City Challenge 2023 Track 5, terdiri dari 200 video berdurasi 20 detik dengan 10 fps.	Model YOLOv8+TTA memperoleh mAP 0.5861 pada 95 FPS kecepatan peringkat 7 dari 192 tim, menunjukkan kinerja yang efisien untuk penerapan real-time pada sistem Smart City.
Do et al. [13] <i>RAMOTS: Real-Time Aerial Multi-Object Tracking.(2025)</i>	Pelacakan multiobjek berbasis UAV menggunakan YOLOv8/v10 + BYTETrack dengan arsitektur Kafka-Spark untuk pemrosesan real-time.	Video UAV dengan objek kendaraan dan pejalan kaki.	Sistem mencapai HOTA 48.14 dan 28 FPS , efektif untuk pelacakan kendaraan udara. Namun, performa menurun di lingkungan darat dan tergantung GPU tunggal.

2.2 Landasan Teori

2.2.1 Big Data

Big Data merupakan kumpulan data berukuran sangat besar dan kompleks sehingga tidak dapat ditangani oleh sistem tradisional. Karakteristik utamanya dikenal sebagai "5V" [14]:

- a. *Volume*: Ukuran data sangat besar, seperti aliran video 24/7 dari empat kamera CCTV dan data pelanggaran yang terus tumbuh di data warehouse. Hal ini memerlukan infrastruktur penyimpanan dan pemrosesan yang skalabel.
- b. *Velocity*: Data harus diproses secara real-time dengan target latensi ≤ 2 detik, sehingga sistem membutuhkan arsitektur pemrosesan yang efisien.
- c. *Variety*: Beragam format data:
 - 1). Tidak Terstruktur: Video, gambar, media sosial.
 - 2). Semi-Terstruktur: Log sistem melalui Apache Kafka.
 - 3). Terstruktur: Data pelanggaran dalam skema relasional.
- d. *Veracity*: Menekankan akurasi dan keandalan data; validasi diperlukan untuk meminimalkan false positive/negative.
- e. *Value*: Data harus menghasilkan wawasan bermanfaat, misalnya melalui dashboard analitik untuk pengambilan keputusan.

Tantangan pada aspek *Volume* dan *Velocity* mendorong adopsi *stream processing* dibandingkan *batch*, dengan arsitektur seperti *Kappa Architecture* yang mendukung pemrosesan historis dan real-time secara bersamaan [14].

2.2.2 Arsitektur Streaming: *Kappa Architecture*

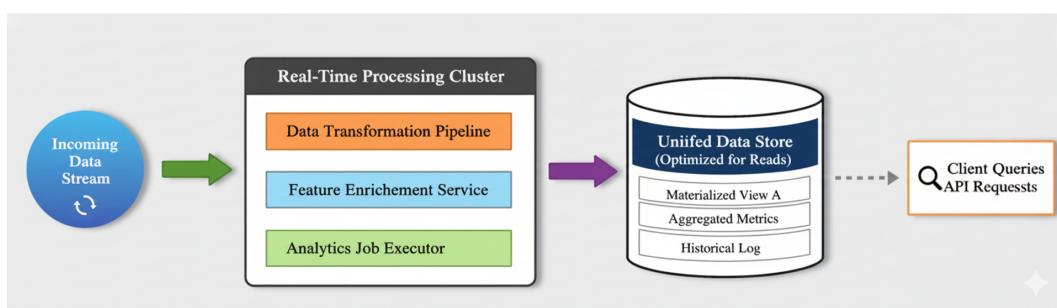
Arsitektur Kappa merupakan pendekatan modern untuk pemrosesan data besar (*Big Data Processing*) yang berorientasi pada pemrosesan aliran data secara *real-time*[15]. Kappa Architecture lahir sebagai penyederhanaan dari Arsitektur Lambda, dengan tujuan mengatasi kompleksitas dan redundansi yang muncul akibat pemisahan antara *batch layer* dan *speed layer*. [15] menekankan bahwa pada dasarnya semua data bersifat temporal dan dapat dipandang sebagai aliran peristiwa (*event stream*). Oleh karena itu, sistem yang dibangun sebaiknya beroperasi pada paradigma *stream-first*, di mana data historis maupun data baru diproses melalui satu jalur pemrosesan tunggal. Arsitektur Kappa menggantikan lapisan *batch* dengan repositori data berbasis log yang berfungsi sebagai sumber kebenaran tunggal (*single source of truth*), misalnya melalui sistem seperti Apache Kafka. Dalam pendekatan ini, setiap peristiwa yang diterima dapat

diproses kembali (*replayable*) kapan pun diperlukan tanpa memerlukan sistem *batch* terpisah. [16] menjelaskan bahwa Arsitektur Kappa memiliki keunggulan dalam hal kesederhanaan, latensi rendah, dan konsistensi kode karena hanya membutuhkan satu alur pemrosesan data. Berbeda dengan Arsitektur Lambda yang menggunakan dua lapisan pemrosesan terpisah, Kappa hanya memiliki satu jalur data berbasis *stream processing engine* (Spark Streaming). Hal ini memungkinkan organisasi untuk mengembangkan, menguji, dan memelihara kode analitik secara lebih efisien, sekaligus meminimalkan risiko inkonsistensi antar-lapisan.

Arsitektur Kappa terdiri atas tiga komponen utama.

- Ingestion Layer*: menangani akuisisi data. Menangani antrian data dari berbagai sumber. Penggunaan *message broker* seperti Apache Kafka dilakukan.
- Stream Processing Layer*: melakukan transformasi data. Melakukan agregasi data. Melakukan analisis data secara terus-menerus. Penggunaan *stream processing framework* dilakukan.
- Serving Layer*: menyimpan hasil olahan. Penyimpanan dilakukan untuk konsumsi pengguna. Penyimpanan dilakukan untuk konsumsi sistem lain. Penggunaan penyimpanan terdistribusi dengan kemampuan *low-latency read* dilakukan.

Model ini lebih cocok untuk aplikasi dengan kebutuhan latensi rendah. Aplikasi tersebut meliputi *IoT analytics*, pemantauan lalu lintas. Arsitektur Kappa memproses data dalam satu pipeline. Kebutuhan sinkronisasi antara lapisan batch tereliminasi. Kebutuhan sinkronisasi antara lapisan streaming tereliminasi. Waktu respon dipercepat. Pemeliharaan sistem disederhanakan.

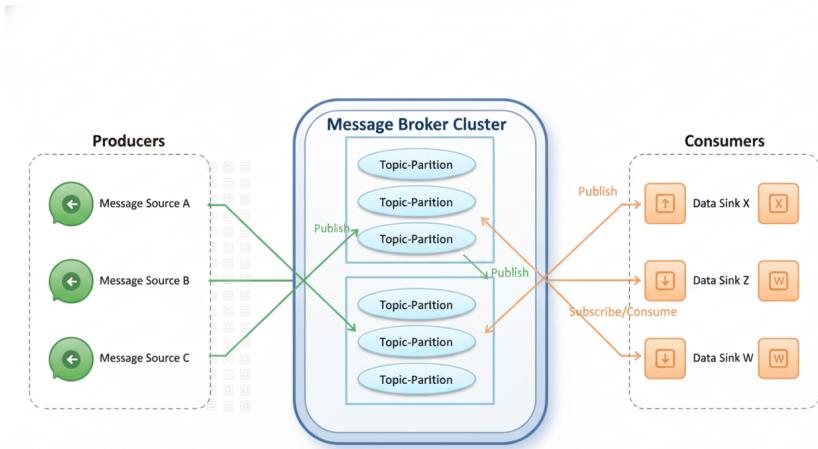


Gambar 2.1 Alur Data Arsitektur Kappa [17]

2.2.2.1 Apache Kafka

Apache Kafka adalah platform *streaming* kejadian terdistribusi yang dirancang untuk menangani aliran data bervolume tinggi dengan latensi rendah dan

keandalan yang tinggi. Kafka berfungsi sebagai *message broker* dan sering digunakan sebagai tulang punggung (*backbone*) dalam arsitektur data *real-time*, termasuk sistem berskala besar dan berbasis *microservices* [18]. Implementasi Kafka untuk sistem pemrosesan video real-time telah dibuktikan dalam penelitian [19], menunjukkan kemampuannya dalam penanganan data streaming berkualitas tinggi secara berkelanjutan.



Gambar 2.2 Arsitektur komponen inti Apache Kafka [19]

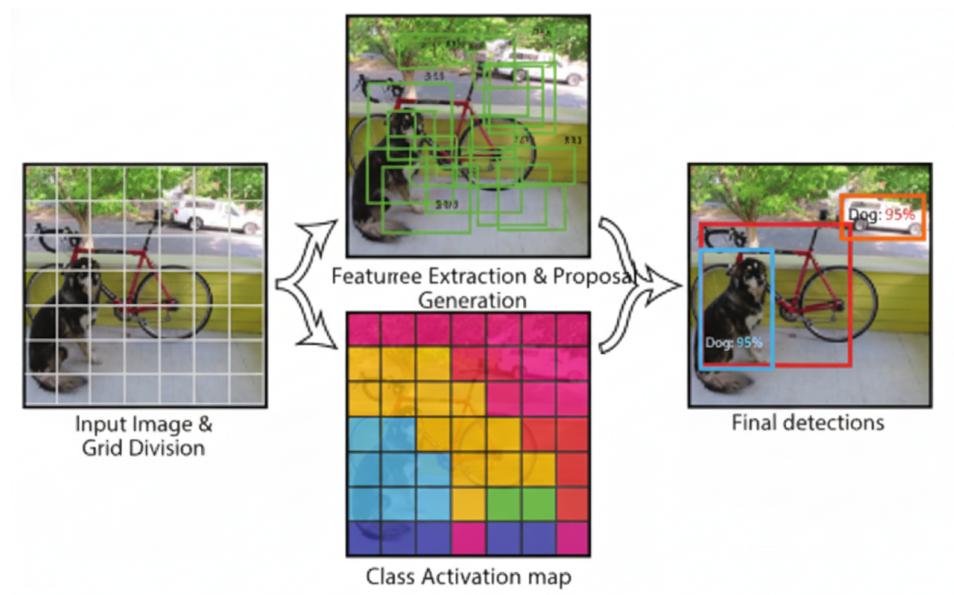
2.2.2.2 Apache Spark

Apache Spark merupakan sebuah *unified analytics engine* yang mendukung pemrosesan data terdistribusi berskala besar melalui model *batch* dan *stream* dalam satu kerangka terpadu [20]. Untuk pemrosesan aliran data *real-time*, Spark menyediakan *Spark Structured Streaming*, yaitu mesin pemrosesan aliran yang memproses data dalam bentuk *incremental micro-batch* dengan semantik deklaratif [21]. Pendekatan ini memungkinkan Spark untuk mempertahankan konsistensi hasil, toleransi kesalahan melalui *checkpointing*, serta *exactly-once processing semantics*.

Spark Structured Streaming memiliki kemampuan fundamental berupa pemrosesan stateful. Kemampuan tersebut mempertahankan informasi historis lintas aliran data. Pemrosesan stateful dimungkinkan melalui fungsi *mapGroupsWithState*. Fungsi tersebut membentuk state untuk setiap grup data. Fungsi *flatMapGroupsWithState* memperbarui state untuk setiap grup data.

2.2.3 YOLO (*You Only Look Once*)

You Only Look Once (YOLO) adalah arsitektur deteksi objek berbasis *Convolutional Neural Network* (CNN) yang melakukan prediksi *bounding box* dan kelas objek secara langsung dalam satu tahap (*single-stage detector*) [22]. Berbeda dengan metode dua tahap seperti R-CNN yang memisahkan proses proposal dan klasifikasi, YOLO memandang deteksi objek sebagai masalah regresi tunggal dari piksel citra ke koordinat dan probabilitas kelas [22].



Gambar 2.3 Illustrasi Mekanisme Prediksi Bounding Box pada YOLO[22]

YOLO membagi citra masukan menjadi grid berukuran $S \times S$. Apabila pusat suatu objek berada dalam suatu *grid cell*, maka cell tersebut bertanggung jawab untuk memprediksi objek tersebut. Setiap cell, model memprediksi B *bounding box*, masing-masing direpresentasikan dengan 5 parameter (x, y, w, h, C) dan C probabilitas kelas. Sehingga output tensor YOLO ditunjukkan pada persamaan 2.1:

$$\text{Tensor Output YOLO} \in \mathbb{R}^{S \times S \times (B \cdot 5 + C)} \quad (2.1)$$

Koordinat pusat kotak relatif terhadap sel dinyatakan sebagai x, y . Lebar dan tinggi relatif terhadap ukuran citra dinyatakan sebagai w, h . Skor kepercayaan dinyatakan sebagai C .

Confidence didefinisikan sebagai probabilitas bahwa kotak tersebut mengandung objek dan seberapa akurat prediksi kotak tersebut. Secara matematis, confidence

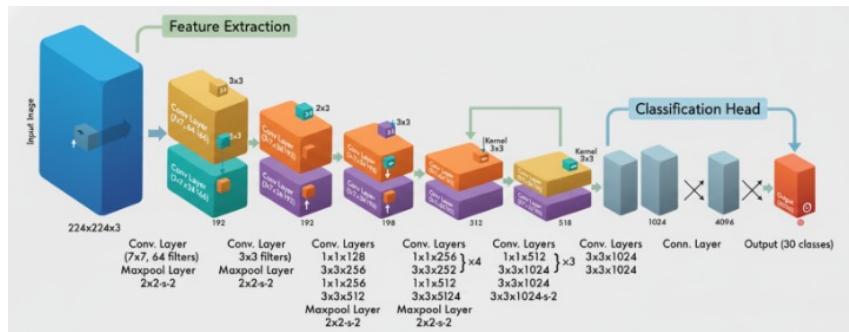
dapat dihitung dengan persamaan 2.2:

$$C = \text{Pr}(\text{Object}) \times \text{IOU}(\text{pred}, \text{truth}) \quad (2.2)$$

Fungsi kerugian YOLO menggunakan *sum-squared error* untuk menyeimbangkan prediksi koordinat, ukuran kotak, *confidence*, dan klasifikasi [22]. Bobot λ_{coord} dan λ_{noobj} digunakan untuk mengontrol dominasi kesalahan koordinat dan penalti kesalahan kotak yang tidak mengandung objek. Rumus fungsi kerugian YOLO ditunjukkan pada persamaan 2.3:

$$\begin{aligned} \text{Loss} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{I}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (2.3)$$

Arsitektur YOLO awal terdiri dari 24 lapisan konvolusional diikuti oleh 2 lapisan *fully connected* [22]. Lapisan-lapisan konvolusional ini menggunakan *filters* berukuran 3×3 dan 1×1 , dengan *max pooling* untuk mengurangi dimensi spasial. Lapisan *fully connected* menghasilkan prediksi akhir untuk kotak pembatas dan kelas objek.

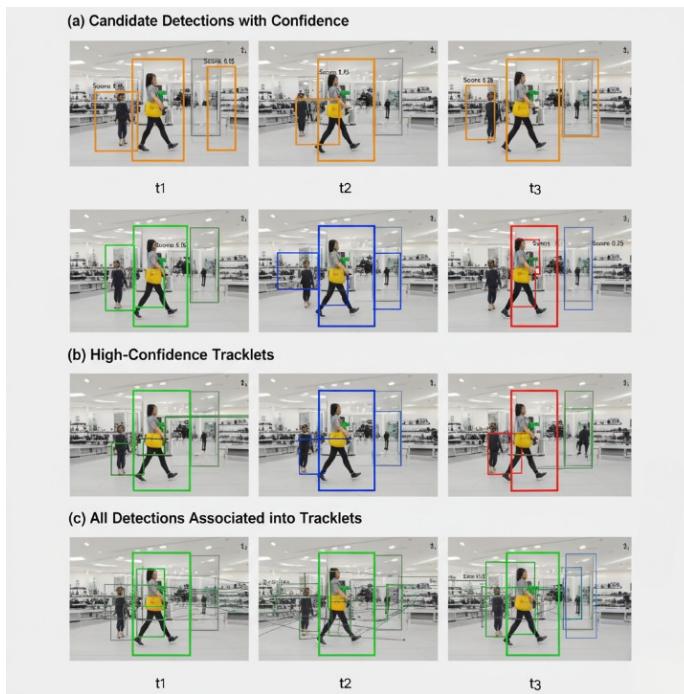


Gambar 2.4 Arsitektur YOLO sebagai *single-stage object detector*[22].

2.2.4 ByteTrack: Multi-Object Tracking Berbasis Asosiasi Dua Tahap

Sistem pemrosesan video *real-time* memerlukan kemampuan untuk mempertahankan identitas objek secara konsisten pada setiap frame agar pelacakan bersifat berkelanjutan. Metode deteksi objek tunggal umumnya hanya mampu menentukan posisi objek tanpa mengelola identitas antar frame. Keterbatasan tersebut mendorong pengembangan pendekatan *Multi-Object Tracking* (MOT) yang mampu menggabungkan deteksi dan pelacakan secara adaptif. Salah satu metode MOT yang banyak diimplementasikan pada sistem visi komputer modern ialah *ByteTrack* [23], yang memanfaatkan mekanisme asosiasi dua tahap untuk menjaga kontinuitas identitas objek selama pergerakan.

Metode ByteTrack mengatasi kelemahan algoritma asosiasi konvensional dengan mempertahankan seluruh hasil deteksi, termasuk yang memiliki skor kepercayaan rendah akibat faktor seperti occlusion, variasi pencahayaan, atau ukuran objek yang kecil. Proses pelacakan dilakukan secara dua tahap, dimulai dengan asosiasi terhadap *deteksi skor tinggi* yang paling dapat diandalkan, diikuti oleh asosiasi terhadap *deteksi skor rendah* guna memulihkan identitas objek yang sempat tidak terdeteksi. Pendekatan ini meningkatkan stabilitas pelacakan serta menurunkan kemungkinan kehilangan identitas pada lingkungan yang dinamis dan padat [23].

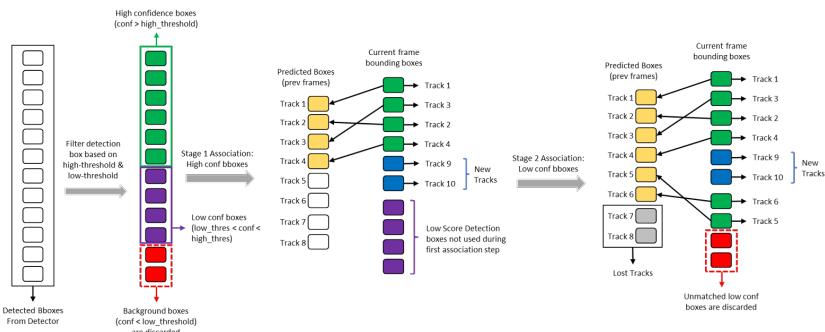


Gambar 2.5 Pipeline ByteTrack [23]. Arsitektur ini mengintegrasikan tahap deteksi dan asosiasi secara bertingkat untuk mempertahankan identitas objek pada video *real-time*.

2.2.4.1 Asosiasi Dua Tahap

ByteTrack menerapkan strategi asosiasi dua tahap untuk mempertahankan identitas objek secara konsisten antar frame. Hasil deteksi dari setiap frame dikelompokkan menjadi dua kategori berdasarkan tingkat kepercayaan, yaitu deteksi dengan skor tinggi dan deteksi dengan skor rendah. Prediksi posisi objek diperoleh menggunakan *Kalman Filter*, sedangkan pencocokan antara hasil deteksi dan prediksi dilakukan dengan menghitung kesamaan spasial melalui metrik *Intersection over Union (IoU)*.

Proses asosiasi dilakukan secara bertingkat. Tahap pertama mencocokkan lintasan aktif dengan deteksi skor tinggi untuk memperoleh pasangan dengan kepercayaan tertinggi. Tahap berikutnya meninjau kembali lintasan yang belum terasosiasi dan mencoba mengaitkannya dengan deteksi skor rendah untuk memulihkan identitas objek yang sempat tidak terdeteksi. Penugasan akhir antara lintasan dan deteksi diselesaikan menggunakan *Hungarian Algorithm* yang meminimalkan biaya asosiasi berbasis IoU. Pendekatan ini terbukti meningkatkan stabilitas pelacakan pada kondisi dengan tingkat occlusion dan variasi pencahayaan yang tinggi.



Gambar 2.6 Ilustrasi proses asosiasi dua tahap pada ByteTrack. Tahap pertama mengasosiasikan deteksi skor tinggi, kemudian tahap kedua memulihkan identitas objek melalui asosiasi dengan deteksi skor rendah[23].

2.2.5 Data Warehouse

Data warehouse merupakan sistem penyimpanan terpusat yang dirancang untuk mendukung proses analitik dan pengambilan keputusan berbasis data dalam jangka panjang. Berbeda dengan basis data operasional yang berfokus pada transaksi harian dan efisiensi pembaruan data, *data warehouse* dioptimalkan untuk kebutuhan *query* kompleks, agregasi data, serta analisis historis [24]. Struktur ini memungkinkan organisasi untuk mengintegrasikan data dari berbagai sistem sumber guna menghasilkan informasi yang komprehensif bagi kebutuhan analisis strategis.

Data yang tersimpan dalam *data warehouse* bersifat terintegrasi, berorientasi subjek, historis, dan relatif tidak mudah berubah. Informasi diorganisasikan untuk mendukung eksplorasi multidimensi, analisis tren jangka panjang, serta perbandingan lintas periode waktu. Dalam perkembangan terkini, konsep *data warehouse* juga diadopsi pada lingkungan komputasi berbasis *cloud* dan *big data* untuk meningkatkan skalabilitas serta efisiensi proses analitik [25].

2.2.6 Data Mart

Data mart merupakan subset terfokus dari *data warehouse* yang dirancang untuk mendukung kebutuhan analitik pada satu domain atau fungsi bisnis tertentu. Berbeda dengan *data warehouse* yang mencakup berbagai domain analisis secara luas, *data mart* memiliki ruang lingkup yang lebih sempit dan spesifik, sehingga memungkinkan akses data yang lebih cepat dan struktur yang lebih sederhana.

Dibandingkan dengan basis data operasional konvensional, *data mart* tidak dirancang untuk menangani transaksi harian, melainkan untuk mendukung analisis multidimensi, agregasi historis, dan pelaporan analitik. Pendekatan ini menjadikan *data mart* lebih sesuai untuk kebutuhan analisis pola, tren, dan evaluasi kinerja berbasis data dalam jangka menengah dan panjang.

Dalam konteks penelitian ini, *data mart* dipilih karena sistem yang dikembangkan berfokus pada satu domain analitik spesifik, yaitu pelanggaran penggunaan helm. Pendekatan tematik ini lebih efisien dibandingkan penggunaan *data warehouse* skala penuh maupun *data lake*, yang umumnya digunakan untuk menyimpan data lintas domain dengan tingkat heterogenitas tinggi. Dengan menggunakan *data mart*, hasil deteksi berbasis visi komputer dapat disimpan sebagai fakta analitik terstruktur yang siap dianalisis secara spasial dan temporal untuk mendukung pengambilan keputusan berbasis data.

2.2.6.1 Star Schema

Star schema merupakan paradigma pemodelan data multidimensi yang banyak diterapkan pada sistem *data warehouse*. Struktur ini terdiri atas satu tabel fakta yang menyimpan metrik numerik utama, serta sejumlah tabel dimensi yang memberikan konteks deskriptif terhadap fakta tersebut. Desain ini mendukung efisiensi proses analitik melalui struktur yang terdenormalisasi secara terkontrol, sehingga mempercepat eksekusi *query* aggregatif dan laporan analitik. Kombinasi struktur fakta dan dimensi tersebut memberikan fleksibilitas tinggi untuk analisis multi-sumbu serta mendukung penyusunan laporan berbasis waktu.

Pendekatan *star schema* memiliki beberapa keunggulan metodologis:

- a. Memungkinkan eksekusi query analitik berkecepatan tinggi melalui struktur denormalisasi,
- b. Menyediakan skema data yang mudah diinterpretasikan untuk pelaporan dan visualisasi,
- c. Mendukung analisis *time-series* serta perbandingan historis secara efisien.

2.2.7 Grafana

Grafana merupakan platform sumber terbuka yang berfungsi untuk visualisasi data dan pengembangan *dashboard* interaktif pada lingkungan *Big Data Warehouse*. Platform ini dirancang untuk mendukung integrasi dengan beragam sistem penyimpanan dan pemrosesan data seperti *Apache Kafka*, *Prometheus*, *InfluxDB*, serta *Elasticsearch*, sehingga efektif digunakan pada skenario pemantauan data bersifat *real-time* [26]. *Grafana* memberikan kerangka kerja visualisasi berbasis metrik yang memungkinkan pengguna untuk memantau performa sistem, mendeteksi anomali, serta melakukan analisis operasional secara dinamis melalui antarmuka berbasis web.

Grafana juga berperan sebagai lapisan presentasi utama yang menghubungkan hasil pemrosesan data dari sistem analitik seperti *Apache Spark* ke dalam bentuk visualisasi interaktif yang diperbarui secara otomatis. Aliran data kontinu yang dihasilkan oleh *streaming pipeline* diproses menggunakan mekanisme *Extract–Transform–Load (ETL)* sebelum ditampilkan pada dasbor analitik. Pendekatan tersebut mendukung paradigma *real-time analytics*, yang memungkinkan pengguna melakukan eksplorasi dan pengambilan keputusan berdasarkan data terkini tanpa menunggu siklus pemrosesan *batch* [27].

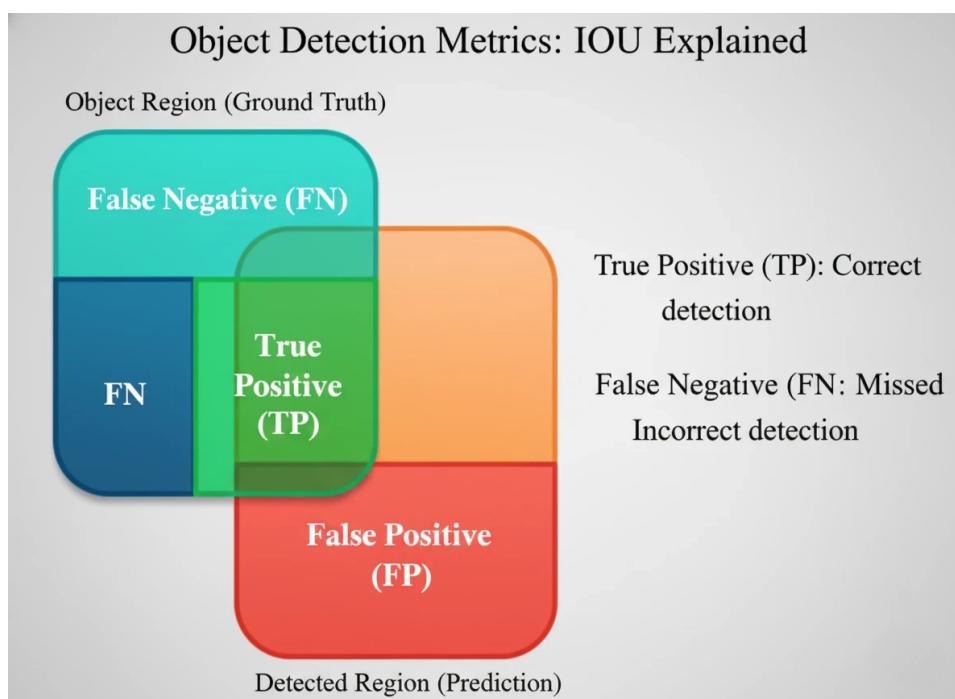
2.2.8 Evaluasi Deteksi Objek

Evaluasi kinerja model deteksi dilakukan untuk mengukur sejauh mana model mampu mengenali serta melokalisasi objek secara akurat dibandingkan dengan anotasi *ground truth*. Proses evaluasi ini mengikuti protokol yang ditetapkan oleh *MS COCO* [28], dengan metrik utama mencakup *Precision*, *Recall*, *Average Precision (AP)*, dan *Mean Average Precision (mAP)*. Pendekatan berbasis metrik ini memungkinkan penilaian menyeluruh terhadap keseimbangan antara kemampuan model mendekripsi objek dan kecenderungannya menghasilkan kesalahan prediksi.

2.2.8.1 Klasifikasi Deteksi: TP, FP, FN, dan TN

Hasil deteksi model diklasifikasikan menjadi empat kategori: *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)* (Gambar 2.7). Klasifikasi ini menjadi dasar perhitungan metrik evaluasi seperti *Precision* dan *Recall*.

True Positive (TP): prediksi yang benar terhadap keberadaan objek dengan kelas dan posisi yang sesuai dengan anotasi *ground truth*. *False Positive (FP)*: prediksi yang salah terhadap objek yang tidak ada atau memiliki kelas yang tidak sesuai. *False Negative (FN)*: kondisi ketika model gagal mendeteksi objek yang sebenarnya terdapat pada citra. *True Negative (TN)*: area pada citra yang tidak mengandung objek dan juga tidak terdeteksi oleh model.



Gambar 2.7 Ilustrasi kategori hasil deteksi objek pada sistem *object detection*: True Positive (TP), False Positive (FP), dan False Negative (FN) [29].

Klasifikasi ini memberikan fondasi kuantitatif untuk mengevaluasi keseimbangan antara kemampuan model dalam mengenali objek yang relevan dan kecenderungannya menghasilkan kesalahan deteksi. Pemahaman terhadap hubungan antar kategori ini menjadi penting dalam interpretasi metrik lanjutan seperti *Precision Recall Curve* dan *mean Average Precision (mAP)*. Hubungan antara *Precision*, *Recall*, dan *Intersection over Union (IoU)* ditunjukkan pada Gambar 2.8. Ketiga metrik tersebut berperan dalam mengukur keseimbangan antara tingkat deteksi yang benar dan kesalahan prediksi, sekaligus menilai

kesesuaian spasial antara kotak prediksi dan anotasi.

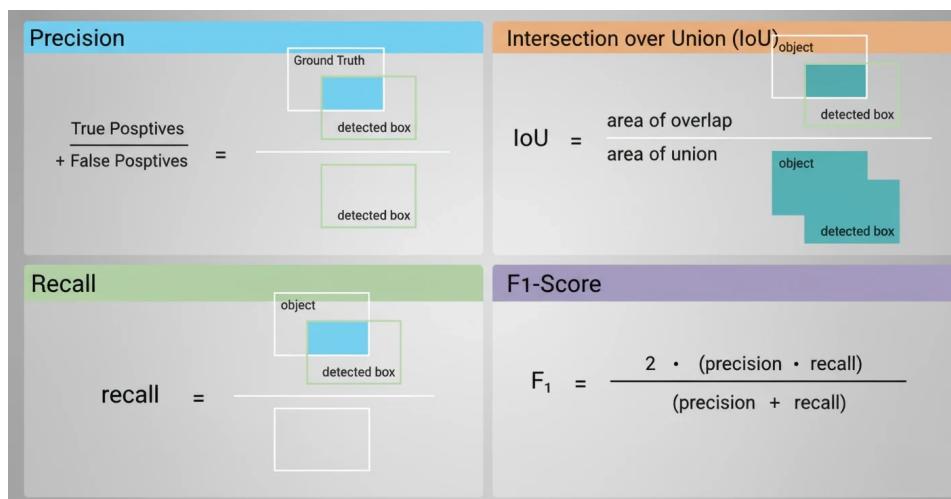
Persamaan yang digunakan untuk menghitung *Precision* dan *Recall* didefinisikan sebagai:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (2.4)$$

Intersection over Union (IoU) digunakan untuk mengukur tingkat kesesuaian geometrik antara kotak prediksi dan anotasi *ground truth*, sebagaimana didefinisikan pada persamaan berikut:

$$\text{IoU}(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2.5)$$

Nilai IoU yang tinggi menunjukkan bahwa area prediksi dan anotasi memiliki tumpang tindih yang besar, yang menandakan lokalisasi yang lebih akurat. Evaluasi berbasis Precision Recall dan IoU digunakan secara bersama untuk menghitung metrik agregat seperti *Average Precision (AP)* dan *mean Average Precision (mAP)*.



Gambar 2.8 Hubungan antara Precision, Recall, dan Intersection over Union (IoU) pada sistem deteksi objek. IoU dihitung berdasarkan area tumpang tindih antara kotak prediksi dan anotasi *ground truth*[30].

2.2.8.2 Average Precision (AP) dan Mean Average Precision (mAP)

Evaluasi performa deteksi objek umumnya menggunakan metrik *Average Precision (AP)* dan *Mean Average Precision (mAP)*, yang menjadi standar penilaian utama pada dataset *MS COCO* [28]. Metrik ini mengukur keseimbangan antara kemampuan model dalam menghasilkan prediksi yang benar dan kestabilan lokalisasi spasial terhadap anotasi *ground truth*.

Average Precision (AP) didefinisikan sebagai luas area di bawah kurva *Precision Recall (PR)* untuk setiap kelas objek. Nilai AP menggambarkan rata-rata presisi yang dicapai model pada berbagai tingkat *recall*, dan secara matematis dapat dituliskan sebagai:

$$AP = \int_0^1 \text{Precision}(\text{Recall}) d(\text{Recall}) \quad (2.6)$$

Nilai AP tinggi menunjukkan bahwa model mampu mempertahankan presisi tinggi pada rentang *recall* yang luas, sehingga deteksi yang dihasilkan konsisten dan stabil. Untuk menilai performa keseluruhan lintas kelas dan tingkat ketelitian lokalisasi, digunakan metrik *mean Average Precision (mAP)* yang merupakan rata-rata dari seluruh nilai AP pada berbagai ambang batas *Intersection over Union (IoU)*. Formulasi mAP yang digunakan pada COCO benchmark adalah:

$$\text{mAP}_{@[0.50:0.95]} = \frac{1}{10} \sum_{k=0}^9 \text{AP}_{\text{IoU}=0.50+0.05k} \quad (2.7)$$

Pendekatan ini mencakup sepuluh tingkat ambang IoU, mulai dari 0.50 hingga 0.95 dengan interval 0.05, untuk memastikan evaluasi model tidak hanya akurat dalam mendekripsi keberadaan objek, tetapi juga presisi dalam menentukan batas spasialnya.

2.2.9 Evaluasi Sistem Streaming

Evaluasi sistem pemrosesan data *real-time* berbasis arsitektur Kappa menekankan tidak hanya pada akurasi keluaran, tetapi juga pada efisiensi aliran data yang diproses secara kontinu dengan jeda minimum [18]. Dua metrik utama yang digunakan adalah *throughput* dan *latency*, yang bersama-sama merepresentasikan kinerja *pipeline* pemrosesan end-to-end [31]. *Throughput* menunjukkan jumlah unit data seperti *frame* video atau pesan Kafka yang dapat diproses sistem per satuan waktu dan biasanya diukur dalam *frame per second (FPS)* [14]. Sebaliknya, *latency* menggambarkan jeda antara waktu kedatangan data dan waktu keluaran hasil [31]. Pada sistem berbasis arsitektur Kappa, seluruh pemrosesan dilakukan melalui satu jalur *stream pipeline*, sehingga latency bergantung pada desain pemrosesan stateful dan efisiensi komunikasi antar komponen. Kinerja sistem streaming yang optimal ditandai dengan throughput tinggi dan latency rendah, umumnya pada rentang *sub-second*, terutama pada aplikasi visi komputer yang memerlukan respons cepat seperti deteksi pelanggaran lalu lintas atau pemantauan

keselamatan [18].

BAB III

METODOLOGI PENELITIAN

3.1 Deskripsi Data

Bagian ini menjelaskan karakteristik data yang digunakan dalam penelitian, mencakup sumber data, ruang lingkup pengambilan data, format dan kualitas data, serta pertimbangan etika dalam pemanfaatan rekaman CCTV.

3.1.1 Jenis dan Sumber Data

Data yang digunakan dalam penelitian ini dikategorikan menjadi dua jenis utama berdasarkan asal dan perannya dalam sistem, yaitu data primer dan data sekunder.

- a. Data Primer: Data primer berupa rekaman video CCTV yang diperoleh dari Unit K3L Institut Teknologi Sumatera (ITERA) berdasarkan izin resmi tertulis. Data yang digunakan merupakan rekaman offline (non-real-time) yang diambil dari satu kamera CCTV yang terpasang di gerbang utama kampus. Rekaman video mencakup periode pengamatan selama lima hari, yaitu dari 2 Februari hingga 6 Februari 2025, dengan rentang waktu perekaman harian antara pukul 07.00 hingga 17.00 WIB. Rekaman ini merepresentasikan arus lalu lintas utama kendaraan bermotor yang masuk dan keluar area kampus, sehingga relevan untuk analisis pelanggaran penggunaan helm pada lingkungan kampus.
- b. Data Sekunder: Untuk mendukung pelatihan dan evaluasi model deteksi objek berbasis YOLOv8, penelitian ini menggunakan dataset sekunder berupa ITERA Helmet Dataset, yaitu dataset kustom yang dibangun dari data primer melalui proses ekstraksi *frame* secara selektif. Pengambilan *frame* dilakukan dengan metode sampling berdasarkan kondisi visual ideal untuk deteksi objek, sehingga diperoleh sekitar 500–700 frame yang representatif. Dataset ini kemudian dianotasi secara manual oleh peneliti dengan melabeli kelas *Helmet* dan *NoHelmet*.

3.1.2 Format dan Karakteristik Data

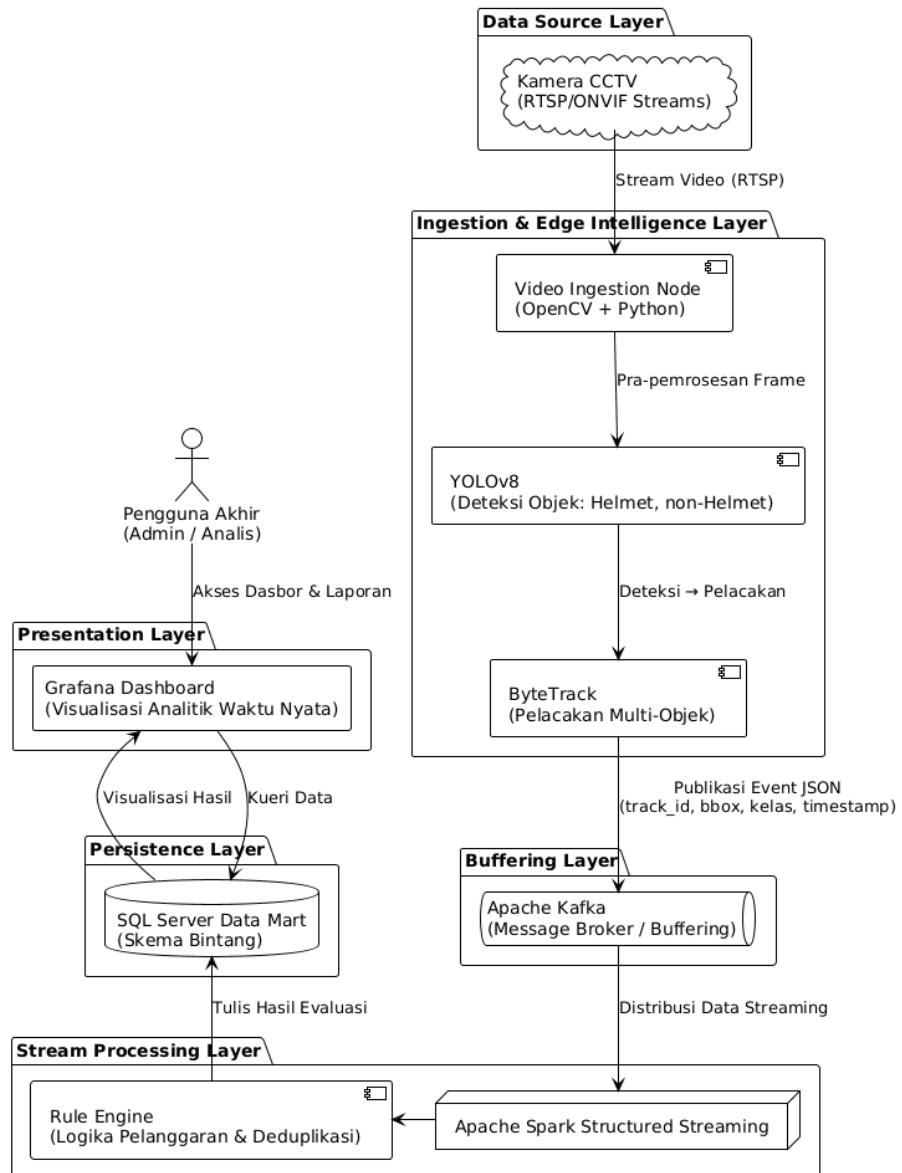
Rekaman video CCTV disimpan dalam format MP4 dengan resolusi 720p dan laju *frame* sebesar 15 FPS. Kondisi visual pada data mencakup variasi pencahayaan pada siang dan sore hari. Variasi kondisi cuaca ekstrem seperti hujan dan cahaya rendah tidak menjadi fokus utama pengambilan data.

Secara keseluruhan, penelitian ini memanfaatkan sekitar 50 jam rekaman video CCTV, yang kemudian diturunkan menjadi sekitar 500–600 frame teranotasi untuk pelatihan model deteksi objek.

3.2 Arsitektur Sistem

Sistem dibangun menggunakan Kappa Architecture [18]. Arsitektur ini memprioritaskan pemrosesan aliran data sebagai satu-satunya jalur untuk menangani data *real-time* dan historis, sehingga menyederhanakan kompleksitas operasional. Arsitektur sistem terdiri atas empat lapisan utama: *Ingestion & Edge Intelligence Layer*, *Buffering Layer*, *Stream Processing Layer*, dan *Persistence & Presentation Layer*. Arsitektur sistem secara keseluruhan ditunjukkan pada Gambar 3.1.

Arsitektur Streaming-First untuk Pelacakan Pelanggaran Helm



Gambar 3.1 Arsitektur Sistem Streaming-First Pelacakan Pelanggaran Helm

3.2.1 Data Source Layer

Lapisan ini merupakan sumber utama data video yang berasal dari kamera CCTV di area pemantauan. Setiap kamera memiliki parameter unik seperti *frame rate*, resolusi, serta posisi geospasial yang direferensikan dalam basis data *dim_camera*. Data ini menjadi input utama untuk sistem *Ingestion Service* di lapisan berikutnya.

3.2.2 Ingestion & Edge Intelligence Layer

Lapisan ini bertanggung jawab untuk akuisisi dan pra-pemrosesan *video frames* dari sumber CCTV. *Ingestion Service* dikembangkan menggunakan *Python + OpenCV*

dan berjalan sebagai *daemon* per kamera.

3.2.3 Buffering Layer

Lapisan ini menggunakan *Apache Kafka* sebagai *message broker* utama yang menjamin *reliability* dan *scalability* aliran data. Kafka memisahkan *producer* (edge nodes) dan *consumer* (Spark Streaming) agar sistem tetap responsif.

3.2.4 Stream Processing Layer

Lapisan ini menangani analisis data secara *real-time* menggunakan *Apache Spark Structured Streaming*. Pipeline ini bertanggung jawab untuk pemrosesan event yang diterima dari Kafka dan evaluasi pelanggaran helm menggunakan *stateful rule engine*.

3.2.5 Persistence Layer

Hasil akhir pelanggaran ditulis ke *Data Mart* dengan desain skema bintang (*star schema*) untuk mendukung analisis multidimensi.

3.2.6 Presentation Layer

Lapisan ini menggunakan *Grafana Dashboard* untuk penyajian visual hasil analitik. Visualisasi ini membantu tim manajemen dalam pemantauan dan pengambilan keputusan berbasis data secara waktu nyata.

3.3 Perancangan Model YOLOv8 dan ByteTrack

3.3.1 Preprocessing dan Anotasi Data

Setiap gambar dalam dataset pelatihan melalui tahapan *preprocessing* dan *augmentation* untuk meningkatkan robustisitas dan kemampuan generalisasi model. Augmentasi dilakukan dengan teknik seperti rotasi acak, *horizontal flipping*, serta penyesuaian pencahayaan, kontras, dan saturasi. Teknik ini membantu model untuk mengenali objek dalam berbagai kondisi di dunia nyata.

3.3.2 Pelatihan Model YOLOv8

Model YOLOv8s, yang telah dilatih pada dataset COCO, di-*fine-tune* menggunakan dataset kustom ITERA dengan strategi *transfer learning*. Pelatihan dilakukan menggunakan *framework* Ultralytics. Hasil pelatihan berupa file model *best.pt* yang memiliki bobot optimal dan siap digunakan untuk inferensi pada lapisan *Ingestion & Edge Intelligence*.

3.4 Logika Deteksi Pelanggaran Helm

Melacak pengendara sepeda motor yang mengenakan dan tidak mengenakan helm, sistem ini mengintegrasikan model deteksi objek berbasis YOLOv8 dengan algoritma pelacakan multiobjek ByteTrack. Integrasi ini memungkinkan pelacakan yang akurat meskipun terjadi oklusi sebagian, pergerakan cepat, atau gangguan visual lainnya. ByteTrack menggunakan pendekatan asosiasi dua tahap (skor tinggi dan skor rendah) untuk menjaga konsistensi identitas objek lintas frame. [23].

Rangkaian proses integrasi sistem adalah sebagai berikut:

- a. *Inferensi Deteksi Objek (YOLOv8)*: setiap *frame* video diproses menggunakan model YOLOv8 untuk menghasilkan *bounding box*, label kelas (*Helmet* atau *NoHelmet* dan *Motorbike*), serta skor kepercayaan. Deteksi dikonfigurasi dengan parameter *confidence threshold* 0,3 dan *IoU threshold* 0,45 untuk menjaga presisi dan *recall* yang seimbang.
- b. *Asosiasi Objek (Rider–Motorbike)*: setelah semua deteksi tersedia, dilakukan proses asosiasi spasial antara objek manusia (hasil deteksi *Helmet* atau *NoHelmet*) dengan objek *Motorbike* yang terdekat. Asosiasi ini menggunakan perhitungan *Intersection-over-Union (IoU)* serta jarak centroid relatif untuk menentukan apakah manusia tersebut sedang menaiki motor. Hanya objek manusia yang memiliki hubungan spasial yang valid dengan *Motorbike* yang diperlakukan sebagai *rider*.
- c. *Pelacakan Multi-Objek (ByteTrack)*: hasil deteksi yang telah diasosiasikan kemudian dikirim ke algoritma ByteTrack, yang melacak setiap *rider* secara individual.
- d. *Logika Evaluasi Pelanggaran Helm*: berdasarkan hasil pelacakan, sistem mendefinisikan kriteria pelanggaran sebagai berikut:
 - 1). Label deteksi adalah *NoHelmet*.
 - 2). Objek berhasil diasosiasikan secara spasial dengan *Motorbike*.
 - 3). Status *NoHelmet* muncul secara berurutan selama setidaknya N frame (misalnya $N = 3$) untuk menghindari *false positive*.
 - 4). Objek memiliki *track ID* yang konsisten (tidak berpindah identitas selama pelacakan).

3.5 Rancangan Basis Data dan Skema Data Mart

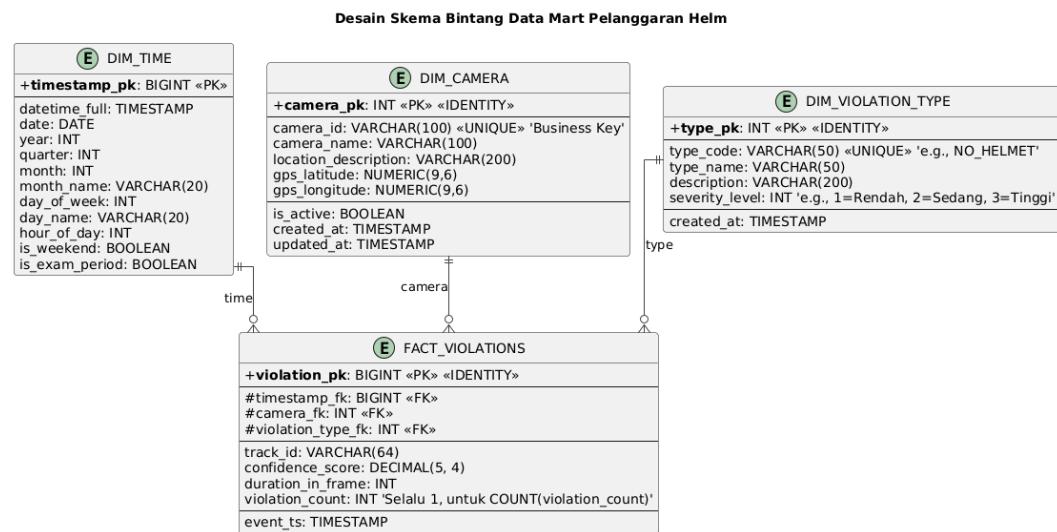
3.5.1 Pendekatan Data-Driven Data Mart

Pendekatan *data-driven* berarti bahwa struktur *data mart* tidak dirancang berdasarkan hipotesis atau kebutuhan bisnis yang abstrak, melainkan dirancang untuk secara langsung menangkap dan menyimpan *event* (kejadian) atau fakta yang dihasilkan oleh sistem sumber data (sistem deteksi video).

Skema bintang yang diusulkan sangat ideal untuk pendekatan ini. *Event* pelanggaran (misalnya, NO_HELMET terdeteksi) adalah unit data inti yang mendorong pengisian data. Tabel fakta (*f_violations*) secara langsung merekam setiap kejadian diskrit ini. Sementara itu, tabel dimensi menyediakan konteks deskriptif (kapan, di mana, dan jenis pelanggaran apa) yang diekstrak dari data yang sama pada saat kejadian. Desain ini memastikan bahwa analisis dan pelaporan (*analytics and reporting*) secara ketat didasarkan pada data aktual yang diamati, memungkinkan pengambilan keputusan yang gesit dan berbasis bukti.

3.5.2 Desain Skema Bintang (Star Schema)

Desain skema bintang secara visual digambarkan pada Gambar 3.2. Desain ini menunjukkan hubungan antara satu tabel fakta pusat (*f_violations*) dengan tabel-tabel dimensi yang mengelilinginya.



Gambar 3.2 Desain Skema Bintang Data Mart Pelanggaran Helm

3.6 Rancangan Dashboard Analitik

Visualisasi hasil deteksi dan pelacakan dilakukan menggunakan *Grafana* yang terhubung langsung ke *data mart* SQL. Dashboard dirancang untuk memberikan

wawasan operasional dan strategis secara *real-time* melalui beberapa panel utama:

- a. *Grafik Deret Waktu*: Menampilkan jumlah pelanggaran helm per menit/jam.
- b. *Peta Geospasial*: Menunjukkan lokasi CCTV dengan jumlah pelanggaran tertinggi (*hotspot*).
- c. *Heatmap Temporal*: Menganalisis pola pelanggaran berdasarkan jam dalam sehari dan hari dalam seminggu.
- d. *Tabel Agregat*: Menyajikan ringkasan statistik pelanggaran harian, mingguan, dan bulanan.

3.7 Alur Penelitian

Alur penelitian dibagi menjadi empat tahapan utama sebagaimana divisualisasikan pada Gambar 3.3.

3.7.1 Tahap 1: Persiapan dan Pengumpulan Data

Tahap ini membangun fondasi dataset dan lingkungan eksperimen dengan kegiatan utama:

- a. *Survei lokasi*: menentukan titik pemasangan kamera dan area pantau.
- b. *Pengumpulan rekaman video*: mengakuisisi video CCTV ITERA pada berbagai kondisi pencahayaan dan cuaca.
- c. *Pembuatan ITERA Helmet Dataset*: mengekstraksi frame, melakukan anotasi manual kelas *Helmet* dan *NoHelmet* menggunakan *LabelImg*, serta membagi data menjadi pelatihan (70%), validasi (20%), dan pengujian (10%).

3.7.2 Tahap 2: Pengembangan dan Pelatihan Model

Tahap ini berfokus pada pengembangan model deteksi objek berbasis YOLOv8.

- a. *Setup lingkungan pelatihan*: menyiapkan dependensi *Python*, *PyTorch*, dan *Ultralytics*, serta berkas konfigurasi *data.yaml*.
- b. *Augmentasi data*: menerapkan rotasi acak ($\pm 15^\circ$), *horizontal flipping*, penyesuaian pencahayaan, dan *Gaussian blur* untuk meningkatkan generalisasi model.
- c. *Fine-tuning YOLOv8*: melatih ulang model *pretrained* YOLOv8 menggunakan dataset kustom hingga diperoleh model terbaik *best.pt*.
- d. *Evaluasi model*: mengukur *precision*, *recall*, dan *mAP@50:95* pada data uji, kemudian memilih model dengan kinerja terbaik untuk diintegrasikan ke pipeline.

3.7.3 Tahap 3: Pengembangan dan Integrasi Pipeline

Tahap ini merealisasikan arsitektur streaming untuk deteksi pelanggaran helm secara *real-time*.

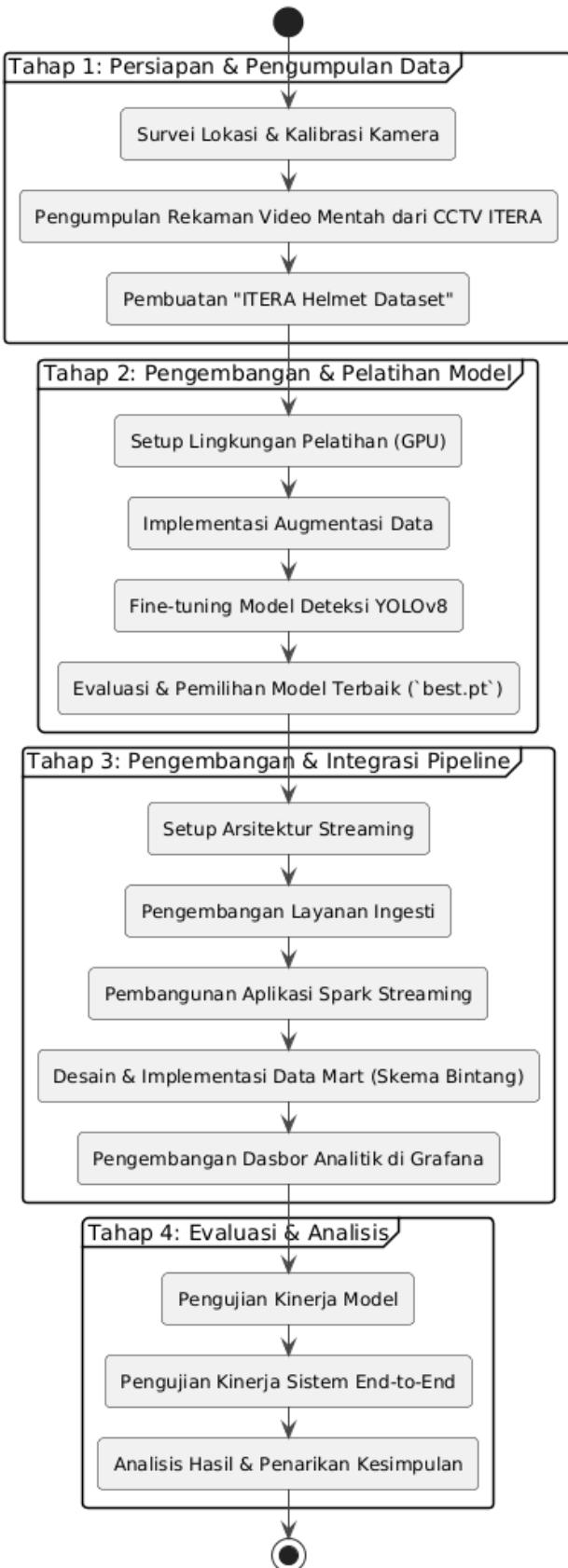
- a. *Setup arsitektur streaming*: mengonfigurasi Apache Kafka sebagai *message broker*, Apache Spark Structured Streaming sebagai *stateful rule engine*, serta SQL sebagai *data mart*.
- b. *Layanan ingest*: membangun layanan Python + OpenCV di sisi *edge* untuk membaca stream CCTV, mengekstrak frame (misalnya 10 FPS), menjalankan inferensi YOLOv8 dan pelacakan ByteTrack, lalu mengirim hasil deteksi (bounding box, kelas, skor, *track_id*) ke topik Kafka `video.detections`.
- c. *Aplikasi Spark Streaming*: mengonsumsi event dari Kafka, melacak status helm per *track_id* menggunakan *stateful rule*, mencatat pelanggaran jika *NoHelmet* muncul berturut-turut selama N frame, dan menyimpan hasil ke skema bintang ($f_violations, dim_time, dim_camera, dim_violation_type$).
- d. *Dasbor analitik*: menghubungkan Grafana ke SQL untuk menampilkan jumlah pelanggaran, tren temporal, *heatmap* waktu, dan indikator kinerja sistem (FPS, latensi).

3.7.4 Tahap 4: Evaluasi dan Analisis

Tahap terakhir mengevaluasi kinerja model dan sistem secara menyeluruh.

- a. *Evaluasi model*: menilai kualitas deteksi YOLOv8 menggunakan *precision*, *recall*, dan *mAP*.
- b. *Evaluasi sistem end-to-end*: mengukur *latency* (waktu dari frame ditangkap hingga hasil tersedia di dasbor) dan *throughput* (FPS) untuk menilai kemampuan pemrosesan *real-time*.
- c. *Analisis hasil*: menginterpretasikan temuan, membandingkan dengan tujuan penelitian, serta menarik kesimpulan dan rekomendasi.

Alur Penelitian Sistem Pelacakan Pelanggaran Helm



Gambar 3.3 Diagram Alur Penelitian Sistem Pelacakan Pelanggaran Helm

Bibliografi

- [1] Institut Teknologi Sumatera, “Draf rencana strategis (renstra) institut teknologi sumatera tahun 2019-2024”, Institut Teknologi Sumatera, Tech. Rep., Feb. 2019. [Online]. Available: <https://www.itera.ac.id/wp-content/uploads/2020/09/Draf-Renstra-Institut-Teknologi-Sumatera-2019-2024.pdf>
- [2] K. Polin, T. Yigitcanlar, M. Limb, dan T. Washington, “The making of smart campus: A review and conceptual framework”, *Buildings*, vol. 13, no. 4, hlmn. 891, 2023.
- [3] I. T. Sumatera, *Dashboard mahasiswa - data akademik dan statistik mahasiswa*, <https://data.itera.ac.id/insights/f5865ac6-3be3-44ce-a319-7641ae1f9ae9>, Accessed: Jan. 29, 2025, 2025.
- [4] O. Purwanti, A. Maulana, dan U. N. Kurniautami, “Analisis pemilihan moda transportasi di kampus itenas”, *Jurnal Rekayasa Sipil*, vol. 12, no. 1, hlmn. 319–326, 2018. [Online]. Available: <http://eprints.itenas.ac.id/246/1/A-13%20Analisis%20Pemilihan%20Moda.pdf>
- [5] R. H. Rachmadian dkk., “Membentuk kesadaran dan keterlibatan mahasiswa sebagai aktor penggunaan transportasi dan energi berkelanjutan di perguruan tinggi”, *Journal of Education Action Research*, vol. 8, no. 1, hlmn. 109–117, 2024. [Online]. Available: <https://ejournal.undiksha.ac.id/index.php/JEAR/article/view/76919>
- [6] ANTARA News Lampung, *Itera kembangkan layanan transportasi umum smart BRT untuk warga*, Okt. 2025. [Online]. Available: <https://lampung.antaranews.com/berita/800457/itera-kembangkan-layanan-transportasi-umum-smart-brt-untuk-warga>
- [7] N. A. Safitri, “Gambaran tingkat kepatuhan penggunaan helm saat berangkat ke kampus dan faktor-faktor yang mempengaruhinya pada mahasiswa diploma tiga jurusan kesehatan lingkungan poltekkes kemenkes yogyakarta”, Diploma Thesis, Poltekkes Kemenkes Yogyakarta, 2020. [Online]. Available: <http://eprints.poltekkesjogja.ac.id/4987/>
- [8] R. Nobrihas, “Kepatuhan penggunaan helm saat berkendara dalam mencegah cedera kepala pada pengendara remaja”, *LaHIMPA Journal*, vol. 2, no. 2, hlmn. 147–153, Nov. 2023. [Online]. Available: <https://journal.unikadelasalle.ac.id/index.php/lhj/article/view/101>
- [9] I. T. Sumatera, *Peraturan rektor institut teknologi sumatera nomor 1 tahun 2017 tentang disiplin mahasiswa*, Peraturan Rektor, Bandar Lampung: ITERA, 2017. Accessed: Nov. 17, 2025. [Online]. Available: <https://fi.itera.ac.id/>

- itera.ac.id/wp-content/uploads/2020/06/PERATURAN_DISIPLIN_MAHASISWA1.pdf
- [10] F. A. Alunjati, “Architecture of four-arm intersection performance evaluation system based on turn movement count and omnidirectional camera”, M.S. thesis, Institut Teknologi Bandung, Bandung, Indonesia, 2025. [Online]. Available: <https://digilib.itb.ac.id/assets/files/2025/RHJhZnQgVGVzaXMgRmlnbyBBZ2lsIEFsdW5qYXRpIC0gMjMyMjMwNjQtM18yNTAxM.pdf>
 - [11] J. Liu, Y. Xie, Y. Zhang, dan H. Li, “Vehicle flow detection and tracking based on an improved yolov8n and bytetrack framework”, *World Electric Vehicle Journal*, vol. 16, no. 1, hlmn. 13, 2025. [Online]. Available: <https://doi.org/10.3390/wevj16010013>
 - [12] A. Aboah, B. Wang, U. Bagci, dan Y. Adu-Gyamfi, “Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8”, di dalam *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE / Computer Vision Foundation, 2023, hlmn. 5350–5358. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023W/AI4CC/html/Aboah_Real-Time_Multi-Class_Helmet_Violation_Detection_Using_Few-Shot_Data_Sampling_Technique_CVPRW_2023_paper.html
 - [13] T. Do, Q. Huynh, K. Kim, dan V. Nguyen, “Ramots: A real-time system for aerial multi-object tracking based on deep learning and big data technology”, *arXiv preprint*, 2025. arXiv: 2502.03760 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2502.03760>
 - [14] N. Marz dan J. Warren, *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems*. Manning Publications, 2015. [Online]. Available: <https://www.manning.com/books/big-data>
 - [15] J. Kreps, *I Heart Logs: Event Data, Stream Processing, and Data Integration*. Sebastopol, CA: O'Reilly Media, Inc., 2014, Accessed on November 12, 2025. [Online]. Available: <https://books.google.com/books?id=gdiYBAAAQBAJ>
 - [16] E. Feick, *Fundamentals of Stream Processing*. Springer, 2018.
 - [17] L. Yang, X. Ding, R. Hou, Y. Tao, dan N. Yang, “Streaming information integration unit for mass customization”, *IOP Conference Series: Materials Science and Engineering*, vol. 768, hlmn. 052120, Mar. 2020.

- [18] J. Kreps, *Questioning the lambda architecture*, O'Reilly Radar, 2014. [Online]. Available: <https://www.oreilly.com/radar/questioning-the-lambda-architecture/>
- [19] A. Ayae, T. Hirotsu, dan K. Oguchi, “A video analysis framework using apache kafka and apache spark streaming”, di dalam *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, hlmn. 4668–4673. [Online]. Available: https://www.is.ocha.ac.jp/~oguchi_lab/Publications/paper2017/bigdata2017_ayae.pdf
- [20] M. Zaharia dkk., “Apache spark: A unified engine for big data processing”, *Commun. ACM*, vol. 59, no. 11, hlmn. 56–65, Okt. 2016. [Online]. Available: <https://doi.org/10.1145/2934664>
- [21] M. Zaharia dkk., “Structured streaming: A declarative api for real-time applications in apache spark”, di dalam *Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18)*, New York, NY, USA: Association for Computing Machinery, 2018, hlmn. 601–614. [Online]. Available: https://people.eecs.berkeley.edu/~matei/papers/2018/sigmod_structured_streaming.pdf
- [22] J. Redmon, S. Divvala, R. Girshick, dan A. Farhadi, “You only look once: Unified, real-time object detection”, di dalam *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, hlmn. 779–788. [Online]. Available: <https://arxiv.org/pdf/1506.02640>
- [23] Y. Zhang dkk., “Bytetrack: Multi-object tracking by associating every detection box”, di dalam *Computer Vision – ECCV 2022*, Ser. Lecture Notes in Computer Science, vol. 13682, Springer, 2022, hlmn. 1–21. [Online]. Available: https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136820001.pdf
- [24] P. Vyas, “Demystifying dimensional modeling for modern data warehousing”, *Journal of Computer Science and Technology Studies*, vol. 7, no. 2, hlmn. 16–25, 2025. [Online]. Available: <https://al-kindipublishers.org/index.php/jcsts/article/view/9184>
- [25] M. Golfarelli dan S. Rizzi, “From star schemas to big data: 20 years of data warehouse research”, di dalam *Data Warehousing and Knowledge Discovery*, Springer, 2024, hlmn. 97–122. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-61893-7_6
- [26] P. Bhatt dan R. Sharma, “Leveraging grafana for real-time data visualization in cloud-native environments”, *International Journal of Data Engineering*,

- vol. 12, no. 3, hlmn. 45–58, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s41019-024-00123-x>
- [27] S. Patel dan D. Mehta, “Integrating stream processing with data warehouses for real-time analytics”, *Journal of Big Data Systems*, vol. 8, no. 1, hlmn. 22–34, 2025. [Online]. Available: <https://journalofbigdatasystems.com/article/view/streaming-warehouse-analytics>
 - [28] T.-Y. Lin dkk., *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1405.0312>
 - [29] M. Lopez-de-la-Calleja, T. Nagai, M. Attamimi, M. Nakano-Miyatake, dan H. Perez-Meana, “Object detection using surf and superpixels”, *Journal of Software Engineering and Applications*, vol. 06, hlmn. 511–518, Jan. 2013.
 - [30] L. Y. D. Team. “Object detection metrics: Precision, recall, map, and more”. Accessed: 12 November 2025, Label Your Data, Accessed: Nov. 12, 2025. [Online]. Available: <https://labelyourdata.com/articles/object-detection-metrics>
 - [31] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017. [Online]. Available: [https://unidel.edu.ng/focelibrary/books/Designing%20Data-Intensive%20Applications%20The%20Big%20Ideas%20Behind%20Reliable,%20Scalable,%20and%20Maintainable%20Systems%20by%20Martin%20Kleppmann%20\(z-lib.org\).pdf](https://unidel.edu.ng/focelibrary/books/Designing%20Data-Intensive%20Applications%20The%20Big%20Ideas%20Behind%20Reliable,%20Scalable,%20and%20Maintainable%20Systems%20by%20Martin%20Kleppmann%20(z-lib.org).pdf)

Tolong ubah paragraf menjadi sebuah point-point saja Tolong ubah paragraf menjadi sebuah point-point saja