

Enhancing Isolation Forest with Threshold-based Filtering and LSTM for Attendance Anomaly Detection

Article History:

Received
17 December 2025
Revised
20 January 2026
Accepted
05 February 2026

ZULLVAN SUGIANTORO, REGINA LIONNIE

Magister Teknik Elektro, Universitas Mercubuana, Indonesia
Email: 55423120006@student.mercubuana.ac.id

ABSTRAK

Penelitian membahas simulasi sistem verifikasi keaslian titik koordinat presensi berbasis GPS. Verifikasi dilakukan menggunakan model Isolation Forest untuk mendeteksi outliers berdasarkan anomali jarak antara presensi masuk dan pulang serta anomali total lintasan yang dikombinasikan dengan Threshold-based Filtering untuk menentukan ambang batas jarak normal, serta LSTM untuk menganalisis pola temporal berdasarkan total lintasan yang tercatat. Hasil pengujian menunjukkan kombinasi Threshold-based Filtering, Isolation Forest, dan Long Short-Term Memory (LSTM) mampu mendeteksi titik koordinat yang tidak valid secara akurat, dari kombinasi tersebut diperoleh hasil akurasi 99,74%, presisi 99,49%, recall 100% dan F1-score 99,74%. Hasil ini membuktikan kinerja kombinasi (hibrida) lebih unggul dibandingkan penggunaan masing-masing komponen model secara terpisah.

Kata kunci: *Presensi Berbasis Lokasi, GPS Spoofing, Threshold-Based Filtering, Isolation Forest, LSTM.*

ABSTRACT

The research discusses the simulation of a GPS-based attendance coordinate point authenticity verification system. Verification is carried out using the Isolation Forest model to detect outliers based on distance anomalies between entry and exit attendances and total path anomalies combined with Threshold-based Filtering to determine the normal distance threshold, and LSTM to analyze temporal patterns based on the total recorded path. The test results show that the combination of Threshold-based Filtering, Isolation Forest, and Long Short-Term Memory (LSTM) is able to detect invalid coordinate points accurately, from this combination obtained accuracy results of 99.74%, precision 99.49%, recall 100% and F1-score 99.74%. These results prove that the performance of the model combination (hybrid) is superior to using each component model separately.

Keywords: *Location-Based Attendance, GPS Spoofing, Threshold-Based Filtering, Isolation Forest, LSTM.*

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



1. PENDAHULUAN

Teknologi *Global Positioning System* (GPS) telah menjadi alat penting dalam berbagai aspek kehidupan, termasuk dalam manajemen kepegawaian. Sistem manajemen kehadiran berbasis GPS merupakan cara yang efektif untuk meningkatkan akurasi, efisiensi, dan transparansi. Sistem ini menggunakan teknologi GPS untuk pencatatan lokasi pegawai, dan memberikan kinerja yang optimal dan tidak membutuhkan banyak waktu **(Bhatt, dkk, 2023)**. Sejak pandemi Covid-19, banyak organisasi baik di sektor swasta maupun lembaga atau kementerian salah satunya Mahkamah Agung melakukan peralihan dari penggunaan presensi *fingerprint* dan manual menjadi sistem berbasis web dengan mengadopsi teknologi GPS **(Mahkamah Agung RI, 2022)**.

Implementasi sistem melalui aplikasi web membuatnya lebih praktis dengan menyediakan antarmuka pengguna yang mendasar dan sederhana bagi pengguna **(Bhatt, dkk, 2023)**. Langkah ini diambil untuk memberikan kemudahan dan keamanan serta memberikan kepastian informasi kehadiran pegawai di lokasi yang tepat pada waktu yang ditentukan. Pegawai dapat melakukan presensi masuk dan pulang melalui perangkat mereka yang dilengkapi dengan GPS, seperti *smartphone* atau laptop yang secara otomatis mencatat lokasi presensi.

Berbagai penelitian telah menyoroti celah manipulasi lokasi pada sistem berbasis GPS. *Fake GPS Defender* mengusulkan solusi *server-side* untuk mendeteksi *mock location* dan *emulator*, namun masih kesulitan membedakan lokasi asli saat pengguna memakai VPN **(Chang, dkk, 2018)**. Pendekatan lain memanfaatkan *isolation forest* pada fitur sinyal seperti *pseudorange*, *Doppler shift*, dan *SNR* untuk mendeteksi *spoofing* dengan akurasi tinggi, tetapi menggunakan data dengan interval tetap dan skenario yang lebih terkendali **(Zuo, dkk, 2021)**. Studi pada aplikasi *ride-hailing* menunjukkan *logistic regression* mampu mengklasifikasikan GPS palsu secara cukup baik, namun pengujian masih terbatas pada data historis, bukan sistem *live* **(Mar'i & Pangestu, 2021)**. Kombinasi *isolation forest* dan *long short-term memory* (LSTM) juga telah dievaluasi untuk deteksi anomali deret waktu, tetapi masih menghasilkan *False Positive* (FP)/*False Negative* (FN) yang signifikan dan belum terhubung dengan aturan domain jarak/lintasan **(Priyanto, dkk, 2021)**. Berbeda dengan studi-studi tersebut, penelitian ini mengintegrasikan *threshold-based filtering*, *isolation forest*, dan LSTM dalam satu kerangka hibrida yang dihubungkan ke sistem presensi Mahkamah Agung dan diuji pada data presensi pegawai riil dengan skenario simulasi *near real time*.

Pada Penelitian ini penulis mengusulkan pengembangan teknik digital forensik untuk verifikasi keaslian titik koordinat presensi dalam sistem informasi kepegawaian Mahkamah Agung (SIKEP) dengan meningkatkan model *isolation forest* menggunakan *threshold-based filtering* dan *long short-term memory* (LSTM). Parameter yang digunakan ialah jarak antara presensi masuk dan pulang, total lintasan dan pola temporal. *Threshold-based filtering* memeriksa ambang batas jarak dan lintasan normal, lalu *Isolation Forest* mendeteksi *outlier* berdasarkan data biner hasil *threshold-based filtering*. Secara paralel LSTM menganalisis pola lintasan dan historis, sehingga GPS palsu dapat langsung ditolak sebelum tercatat dalam *database*.

Dengan demikian, sistem ini dapat meningkatkan integritas data presensi dan mendukung pengelolaan kepegawaian yang lebih transparan dan akurat. Implementasi teknik ini diharapkan dapat menjadi model yang dapat diadopsi oleh institusi lain yang menghadapi tantangan serupa dalam verifikasi data presensi berbasis GPS. Hasil penelitian ini tidak hanya akan memperkuat sistem informasi kepegawaian Mahkamah Agung tetapi juga berkontribusi pada pengembangan metode digital forensik yang lebih canggih dan dapat diandalkan.

2. METODE PENELITIAN

Bagian ini menguraikan desain penelitian, karakteristik data, serta prosedur pra-pemrosesan. Kami merancang metode hibrida menggunakan *Threshold-based filtering*, *Isolation Forest*, dan LSTM. Selanjutnya, skema fusi keputusan dan metrik evaluasi serta rancangan eksperimen untuk verifikasi keaslian titik koordinat presensi berbasis data GPS pada SIKEP. Pendekatan hibrida ini menggabungkan keputusan dari tiga komponen utama. Pertama, *Threshold-based filtering* berfungsi sebagai penjaga berbasis aturan domain. Kedua, *Isolation Forest* mendeteksi *outlier* tanpa pengawasan pada fitur lintasan. Terakhir, LSTM bertugas menangkap anomali pada sekuens pergerakan. Keputusan akhir menggunakan logika fusi yang memprioritaskan deteksi konsisten antar komponen. Mekanisme ini mengurangi *false negative* sekaligus meningkatkan ketahanan terhadap manipulasi. Tujuannya adalah membangun simulasi verifikasi otomatis untuk pemeriksaan secara *real time*.

2.1 Desain Penelitian dan Kerangka Konseptual

Penelitian ini merupakan studi eksperimental komputasional untuk pengembangan teknik digital forensik dalam verifikasi keaslian titik koordinat presensi. Kerangka umum metode menggabungkan tiga lapis deteksi:

1. Penyaringan berbasis ambang (*threshold-based filtering*) untuk menangkap anomali yang eksplisit menurut aturan domain antara lain jarak masuk–pulang = 0 m atau > 300 m, total lintasan = 0 m atau > 300 m.
2. *Isolation Forest* sebagai *unsupervised anomaly detector* yang mempelajari pola dari fitur hasil penyaringan (indikator biner) dan mengisolasi *outlier*.
3. *Long Short-Term Memory* (LSTM) untuk memodelkan dinamika sekuens spasial (trajektori *latitude-longitude*) dan mendeteksi anomali berdasarkan *reconstruction error*.

Tabel 1. Kolom Utama dan Sampel Data yang Digunakan

Tanggal	TimeMasuk	TimePulang	DetectLokasi	LokasiMasuk	LokasiPulang
			-		
			6.1707195,106.8264038,22.7462		
			00561523438,-		
			6.1707195,106.8264038,22.7462		
			00561523438,-		
			6.1707195,106.8264038,22.7462		
			00561523438,-		
			6.1707195,106.8264038,22.7462		
			00561523438,-		
2-1-2024	05:07:34	18:41:32	6.1707195,106.8264038,25.1164	6.1707343,106.826	6.1707195,106.826
			0167236328,-	4487	4038
			6.1707195,106.8264038,25.1164		
			0167236328,-		
			6.1707195,106.8264038,25.1164		
			0167236328,-		
			6.1707195,106.8264038,25.1164		
			0167236328,-		
			6.1707195,106.8264038,25.1164		
			0167236328		
			-		
			6.1701041,106.8263665,16.2870		
			0065612793,-		
			6.1701041,106.8263665,16.2870		
			0065612793,-		
			6.1701041,106.8263665,16.2870		
			0065612793,-		
2-1-2024	06:04:13	16:38:28	6.1701041,106.8263665,100,-	6.1701002,106.826	6.1701041,106.826
			6.1701041,106.8263665,100,-	3879	3665
			6.1701041,106.8263665,100,-		
			6.1701041,106.8263665,100,-		
			6.1701041,106.8263665,100		

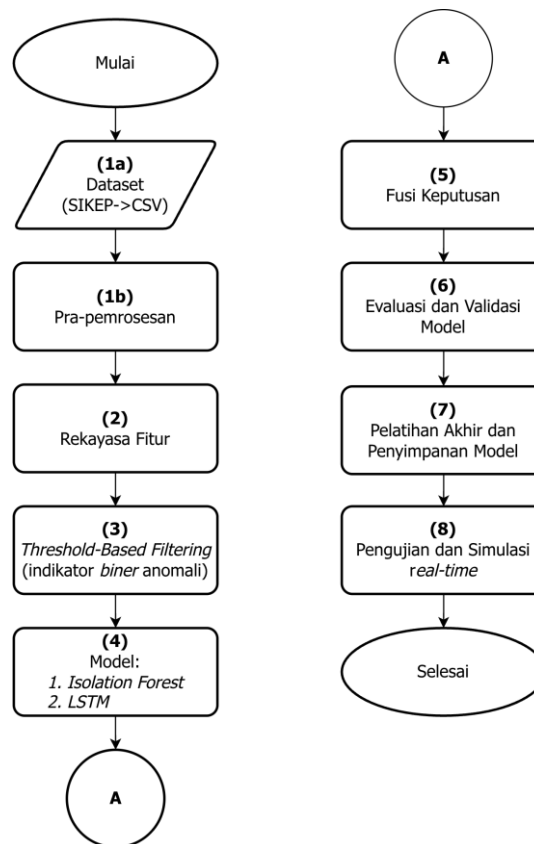
2.2 Data dan Sumber Data

Data penelitian berasal dari aplikasi SIKEP untuk periode Januari–Desember 2024. *Dataset* mencakup seluruh presensi pegawai yang berkantor di Gedung Mahkamah Agung RI di Jakarta Pusat, dengan titik referensi koordinat kantor sebagai pusat area presensi. Data diambil melalui kueri SQL oleh pihak berwenang, diekspor ke format *Comma-Separated Values (CSV)*, dan hanya mencakup atribut yang relevan dengan analisis presensi (tanggal, waktu, dan koordinat lokasi). Seluruh proses pengambilan data mengikuti prosedur resmi dan prinsip perlindungan data pribadi yang berlaku. Data primer yang diambil dari SIKEP memiliki struktur mencakup tanggal, waktu dan data geospasial. Kolom utama yang digunakan dapat dilihat pada Tabel 1.

Kolom *DetectLokasi* berisi rangkaian titik koordinat yang direkam perangkat selama sesi presensi. Pada *dataset*, setiap titik umumnya tersusun sebagai tiga nilai numerik berurutan (*latitude, longitude, accuracy*) yang dipisahkan koma dan berulang mengikuti urutan waktu perekaman. Token non-numerik (mis. "Android", "Mac", "Linux", atau "browser") yang kadang menyertai data diabaikan karena tidak merepresentasikan koordinat. Dalam penelitian ini, perhitungan jarak (*Haversine*) hanya menggunakan *latitude* dan *longitude* pada *DetectLokasi*, *LokasiMasuk*, dan *LokasiPulang*, sedangkan *accuracy* tidak dimasukkan ke perhitungan jarak. Secara operasional, setiap *triplet* dibaca sebagai satu titik, urutan titik membentuk lintasan (*trajectory*) yang digunakan untuk menghitung total lintasan.

2.3 Alur Penelitian

Tahapan penelitian secara rinci diilustrasikan pada Gambar 1.



Gambar 1. Tahapan Penelitian Simulasi Verifikasi Keaslian Titik Koordinat Presensi

Alur penelitian pada Gambar 1 diawali (1a) akuisisi dan (1b) pra-pemrosesan data presensi dari SIKEP. Tahap (2) menghitung fitur jarak masuk–pulang, total lintasan, dan pola temporal. Tahap (3) membentuk indikator anomali berbasis *threshold* menjadi data *biner*. Tahap (4) melatih dan menjalankan *Isolation Forest* pada fitur biner hasil aturan. Secara paralel, LSTM menganalisis sekuens koordinat yang dinyatakan normal. Tahap (5) melakukan fusi keputusan antara aturan dan model pembelajaran mesin. Tahap (6) mengevaluasi dan memvalidasi kinerja menggunakan *confusion matrix*. Tahap (7) melatih ulang model akhir, lalu menyimpannya untuk operasional. Tahap (8) menguji sistem melalui simulasi verifikasi presensi *near real time* via *Representational State Transfer Application Programming Interface* (REST API).

2.4 Pra-pemrosesan

Pra-pemrosesan diawali dengan *parsing* koordinat dari kolom *LokasiMasuk*, *LokasiPulang*, dan *DetectLokasi* menjadi pasangan [*lat*, *lon*] yang mempertahankan urutan waktu. Nilai hilang dan koordinat yang tidak lengkap diberi penanda khusus agar dapat diperlakukan defensif pada tahap fitur dan pemodelan. Untuk keperluan LSTM, sekuens koordinat dinormalisasi menggunakan *Min-Max* ke rentang [0,1], lalu diseragamkan panjangnya melalui *padding/truncating* hingga 50 langkah waktu. Hasil akhir tahap ini adalah tabel terstruktur yang memuat pasangan koordinat masuk–pulang yang siap dihitung jaraknya, sekuens *DetectLokasi* yang telah dinormalisasi, serta metadata *seq_len* dan *seq_valid* untuk menyaring data yang tidak memenuhi kualitas minimum.

Hasil ini menjadi prasyarat bagi rekayasa fitur dan tahap *threshold-based filtering*, *isolation forest*, serta LSTM yang memerlukan sekuens terstandar untuk menghitung rekonstruksi *error* dan menetapkan ambang berbasis kuantil. Setelah pra-pemrosesan diperoleh 11.622 sampel data presensi pegawai yang siap digunakan. Hasil pra-pemrosesan dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pra-pemrosesan Data

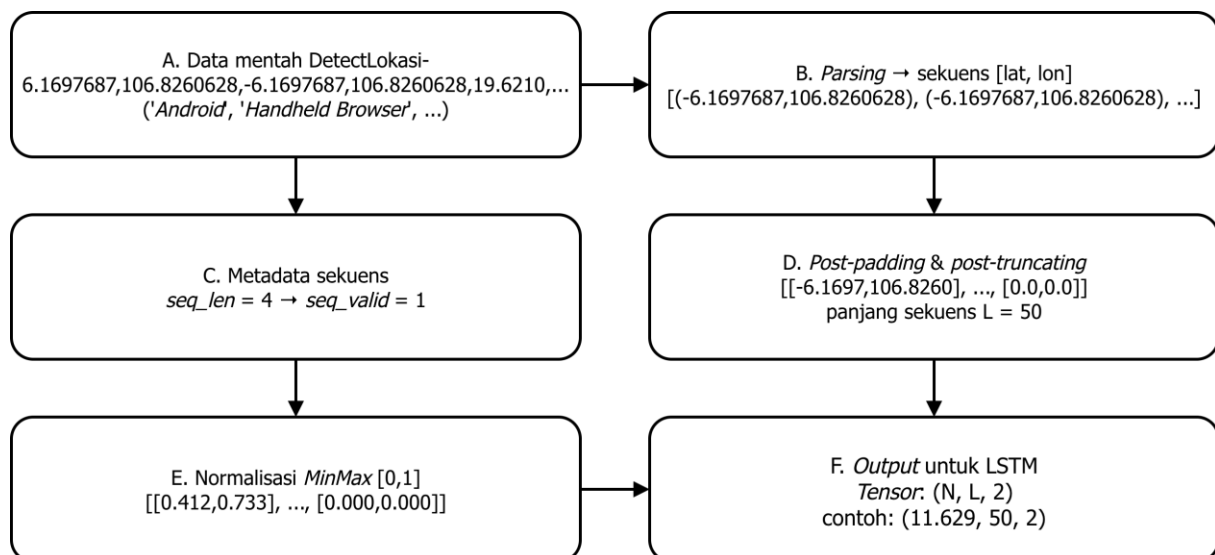
lat_masuk	lon_masuk	lat_pulang	lon_pulang	seq_len	seq_valid	DetectLokasi
-6.1707343	106.8264487	-6.1707195	106.8264038	9	1	-6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038, -
						6.1707195, 106.8264038
-6.1705653	106.8268811	-6.1703275	106.8264782	5	1	-6.1702194, 106.8264715, -
						6.1702194, 106.8264715, -
						6.1703275, 106.8264782, -
						6.1703275, 106.8264782, -
						6.1703275, 106.8264782
-6.1708096	106.8261826	-6.170403	106.8272021	6	1	-6.170403, 106.8272021, -
						6.170403, 106.8272021, -
						6.170403, 106.8272021, -
						6.1704086, 106.8272069, -
						6.1704218, 106.8272049, -
						6.1704218, 106.8272049

2.5 Rekayasa Fitur dan Perhitungan Jarak

Fokus pada rekayasa fitur (*feature engineering*) adalah membuat variabel-variabel turunan yang akan digunakan sebagai input model. Variabel-variabel tersebut meliputi jarak antar titik masuk dan pulang (*jarakmasukpulang*) dan total jarak lintasan (*total_lintasan*) sebagai penjumlahan jarak antar titik yang berturut disajikan dalam format meter/kilometer untuk memperoleh indikator numerik yang relevan dengan logika domain.

Perhitungan jarak geografis antar dua titik koordinat masuk dan pulang serta total jarak lintasan presensi merupakan komponen krusial. Rekeyasa fitur ini dihitung menggunakan rumus *Haversine*, dengan nilai rata-rata radius bumi = 6371000.0 atau 6.371 km (NGA, 2014). Jarak geografis antara titik masuk dan titik pulang, maupun antar titik berturut dalam lintasan *DetectLokasi*, dihitung menggunakan rumus *Haversine* sehingga menghasilkan jarak lingkaran besar (*great-circle distance*) dalam satuan meter (Winarno, dkk, 2017). Implementasi dilakukan secara tervektorisasi untuk efisiensi, dan dari sini dibentuk dua fitur turunan utama yaitu *jarakmasukpulang* dan *total_lintasan*.

Untuk memodelkan dinamika pergerakan harian, data pada kolom *DetectLokasi* diparse menjadi sekuens dua dimensi [*lat,lon*] dengan mengabaikan komponen non-koordinat *accuracy* dan token non-numerik seperti *string* "Android", "Mac", "Linux", atau "browser" yang kadang menyertai data. Hasil disimpan sebagai *string* koordinat berurutan antara lain *lat1, lon1, lat2, lon2* dan seterusnya, agar memudahkan dalam menghitung panjang sekuens-nya (*seq_len*). Baris dinyatakan valid (*seq_valid = 1*) jika memiliki paling sedikit dua titik (*seq_len ≥ 2*) sehingga secara logis membentuk satu segmen gerak.



Gambar 2. Ilustrasi Parsing Kolom *DetectLokasi*

Output tahap ini adalah *tensor* sekuens berukuran seragam (N,L,2), *N* adalah jumlah sekuens, *L* adalah panjang sekuens dan 2 adalah jumlah nilai data yang siap digunakan untuk pelatihan dan inferensi LSTM, beserta metadata *seq_len/seq_valid* untuk penandaan observasi yang tidak memenuhi syarat minimum kualitas data.

2.6 Threshold-Based Filtering

Threshold-based filtering menerjemahkan pengetahuan domain menjadi aturan eksplisit berbasis jarak yang menghasilkan indikator *biner* (0 = normal, 1 = anomali) sebelum diproses model pembelajaran mesin. Penelitian ini menetapkan dua ambang utama (*threshold* jarak): jarak masuk-pulang (d_{MP}) = 300 m dan total lintasan (d_{tot}) = 300 m. Kedua jarak dihitung menggunakan rumus *Haversine* berbasis *latitude* dan *longitude*. Ambang ini diturunkan dari *geofencing* statis SIKEP yang mensyaratkan presensi masuk/pulang hanya valid dalam area kantor. Berdasarkan pengukuran tata letak gedung, radius fisik maksimum tercatat 210,31 m. Oleh karena itu, digunakan radius toleransi 300 m (*buffer* ±90 m) untuk mengakomodasi ketidakpastian akurasi GPS akibat *multipath* dan kondisi atmosfer, serta menekan risiko *false negative*.

Secara operasional, jarak dianggap wajar bila > 0 dan ≤ 300 m; nilai 0 m maupun > 300 m diperlakukan sebagai indikasi anomali, termasuk kemungkinan koordinat identik/invalid atau titik statis akibat manipulasi. Indikator anomali dari (d_{MP}) dan (d_{tot}) kemudian digabung dengan logika OR menjadi *anomali_berdasarkan_aturan*. Ambang 300 m berfungsi sebagai *guardrail* di hulu proses (Ruff, dkk, 2021), sedangkan pada komponen LSTM ambang ditetapkan menggunakan kuantil tinggi dari distribusi *reconstruction error* agar lebih tahan terhadap pencilan (Zhong, dkk, 2023). Aturan domain ini mempercepat proses *review*, menjadi fitur bagi *isolation forest*, serta berperan sebagai penjaringan awal sebelum validasi sekuens oleh LSTM, sekaligus meningkatkan presisi operasional dan *explainability* tanpa menafikan kekuatan pendekatan *data-driven* (Pang, dkk, 2022).

2.6 Isolation Forest

Isolation Forest digunakan sebagai detektor anomali tak berlabel (*unsupervised*) yang bekerja dengan mengisolasi observasi melalui pemotongan acak (*random partitions*) pada hutan pohon keputusan secara intuitif, titik yang aneh akan lebih cepat terisolasi (memiliki rata-rata panjang jalur yang lebih pendek) dibanding titik normal. Dalam lanskap metode deteksi anomali modern, *isolation forest* diklasifikasikan sebagai pendekatan dangkal (*shallow*) yang tetap relevan dan sering digunakan sebagai komponen hibrida bersama teknik berbasis aturan maupun model pembelajaran mendalam karena efisien, tidak memerlukan label, dan cukup tangguh terhadap skala fitur yang beragam. Hal ini menempatkan *isolation forest* sebagai *baseline* penting/kuat dalam praktik, khususnya ketika data anomali langka dan distribusi dapat bergeser (Ruff, dkk, 2021).

Pada penelitian ini, *isolation forest* digunakan di hilir aturan domain. Dua indikator *biner rule-based* menjadi input *isolation forest*:

$$\chi = [I_1, I_2] \in \{0,1\}^{n \times 2} \quad (1)$$

Dengan :

- I_1 : anomali jarak presensi masuk-pulang (>300 m, termasuk kasus $d_{MP} = 0$ m karena koordinat identik/invalid);
- I_2 : total lintasan $d_{tot} > 300$ m atau $d_{tot} = 0$ m

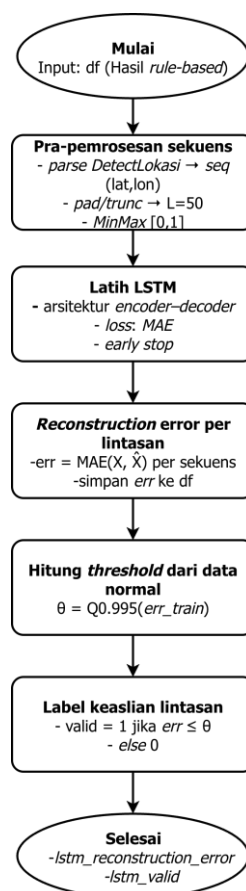
Isolation Forest digunakan setelah *layer* aturan domain, dengan input dua fitur biner [d_{MP} , d_{tot}]. Model dilatih secara *unsupervised* dengan parameter *contamination*=0,15 untuk mengasumsikan sekitar 15% observasi sebagai anomali. Intinya, observasi yang minoritas akan lebih cepat terisolasi di dalam hutan pohon keputusan dan memperoleh skor anomali yang tinggi berdasarkan rata-rata jumlah langkah untuk data yang pada umumnya dalam *dataset* (Priyanto, dkk, 2021). Implementasi *scikit-learn* digunakan untuk menghasilkan label -1 (anomali) dan 1 (normal) yang kemudian dipetakan menjadi kolom *isolation_forest_anomaly* bernilai 1 untuk anomali dan 0 untuk normal.

2.7 Long Short-Term Memory (LSTM)

Dalam arsitektur sistem verifikasi presensi yang dikembangkan, model LSTM difungsikan sebagai filter lapis ketiga untuk menangani data yang bersifat sekuensial (*time series*), yaitu rangkaian koordinat pergerakan pegawai (*DetectLokasi*). Berbeda dengan model *isolation forest* yang memproses fitur secara global, LSTM memiliki kemampuan untuk mengingat dependensi jangka panjang dalam urutan data, sehingga efektif dalam mendeteksi pola manipulasi GPS (*GPS Spoofing*) yang sering kali memiliki karakteristik urutan perpindahan yang tidak wajar atau melompat-lompat secara logis (Aggarwal, 2023).

Arsitektur mengikuti formulasi standar jaringan LSTM, dengan satu lapisan *encoder* berisi 64 unit tersembunyi yang menempatkan sekuens [lat, lon] ter-skala ke dalam representasi laten berdimensi tetap. Vektor ini kemudian direplikasi oleh *RepeatVector* dan diproses oleh satu lapisan *decoder* LSTM 64 unit (Kirichenko, dkk, 2024). Tahap akhir menggunakan *TimeDistributed (Dense)* untuk merekonstruksi kembali sekuens input. Lapisan ini bekerja dengan menerapkan *Dense layer* yang sama pada setiap *timestep* secara independen, atau dengan kata lain, membagikan bobot (*shared weights*) yang sama untuk setiap irisan waktu, sehingga dimensi fitur keluaran kembali sama seperti masukan (Almaghrabi, dkk, 2024). Mekanisme ini memastikan setiap titik koordinat dipetakan kembali ke dimensi aslinya tanpa merusak struktur urutan data (Nigam & Srivastava, 2023). Fungsi aktivasi ReLU (*Rectified Linear Unit*) digunakan dalam implementasi untuk mengurangi risiko *vanishing gradient* dan mempercepat konvergensi saat pelatihan (Jiang, dkk, 2018).

Sebelum dilatih, sekuens koordinat dinormalisasi dengan *Min-Max scaling* ke rentang [0,1] dan diseragamkan panjangnya menjadi 50 *timestep*. LSTM dilatih hanya pada data yang tidak menunjukkan indikasi anomali aturan ($I_1 = 0$ dan $I_2 = 0$), sehingga model belajar pola lintasan presensi yang dianggap wajar. Pelatihan menggunakan *optimizer Adaptive Moment Estimation* (Adam) dengan fungsi kerugian *Mean Absolute Error* (MAE) dan mekanisme *early stopping* untuk mencegah *overfitting* (menghafal) (Anam, dkk, 2024). Setelah pelatihan, setiap sekuens direkonstruksi dan dihitung *reconstruction error*-nya sebagai MAE antara input dan *output*. Ambang keputusan ditetapkan pada kuantil 99,5% (0.995) distribusi *error* data normal sebagai batas anomali pada model *autoencoder* deret waktu (Yahya, dkk, 2025), sekuens dengan *error* di bawah ambang dikategorikan valid (*lstm_valid* = 1), sedangkan sekuens dengan *error* di atas ambang diperlakukan sebagai anomali (*lstm_valid* = 0).



Gambar 3. Alur Proses LSTM

2.8 Skema Fusi Keputusan (*Hybrid Decision Logic*)

Ringkasan definisi variabel utama dan definisi operasionalnya disajikan pada Tabel 3.

Tabel 3. Definisi Variabel Utama

Lapisan/Level	Label	Nilai	Aturan	Sumber Logika
Rule based	<i>anomali_jarak_masuk_pulang</i>	1 = anomali, 0 = normal	Jarak titik masuk-pulang 0 m atau > 300 m	<i>Threshold</i> jarak
Rule based	<i>anomali_total_lintasan</i>	1 = anomali, 0 = normal	Total lintasan pergerakan 0 m atau > 300 m dalam satu sesi presensi	<i>Threshold</i> total lintasan
Ringkasan aturan	<i>anomali_berdasarkan_aturan</i>	1 = anomali, 0 = normal	1 jika salah satu dari dua indikator rule based bernilai 1	OR dari dua aturan
<i>Isolation Forest</i>	<i>isolation_forest_anomaly</i>	1 = anomali, 0 = normal	<i>Record</i> tampak "terisolasi/ganjil" berdasarkan kombinasi dua indikator <i>rule based</i>	Model <i>Isolation Forest</i>
LSTM	<i>lstm_reconstruction_error</i>	Nilai kontinu (≥ 0)	Besar error rekonstruksi lintasan sekuens koordinat	Output rekonstruksi LSTM
LSTM	<i>lstm_valid</i>	1 = valid, 0 = anomali	1 jika $error \leq threshold$ (persentil 0.995), 0 jika $error > threshold$	Perbandingan dengan <i>threshold</i>
Ringkasan Model	<i>anomali_berdasarkan_model_ml</i>	1 = anomali, 0 = normal	1 jika IF mendeteksi anomali atau LSTM menganggap sekuens tidak valid	OR: <i>isolation_forest_anomaly</i> & <i>lstm_valid</i>
Keputusan Akhir (Fusi Hibrida)	<i>ANOMALI_FINAL</i>	1 = anomali, 0 = normal	1 jika aturan atau pembelajaran mesin menandai <i>record</i> sebagai anomali	OR: <i>anomali_berdasarkan_aturan</i> & <i>anomali_berdasarkan_model_ml</i>

Fusi keputusan dilakukan secara bertingkat. Pada level aturan, indikator *anomali_jarak_masuk_pulang* dan *anomali_total_lintasan* digabung menjadi *anomali_berdasarkan_aturan*. Pada level model, label *isolation_forest_anomaly* dan *lstm_valid* digabung menjadi *anomali_berdasarkan_model_ml*, di mana observasi dinyatakan anomali bila minimal salah satu model mendeteksi kegagalan. Label akhir *ANOMALI_FINAL* bernilai 1 bila aturan maupun salah satu model pembelajaran mesin menandai *record* sebagai anomali. Dengan logika OR yang konservatif ini, sistem dirancang *risk-averse*, setiap sinyal risiko dari aturan maupun model cukup untuk memicu *review* forensik.

2.9 Evaluasi dan Validasi Model.

Evaluasi dan validasi model deteksi anomali dilakukan secara kuantitatif menggunakan *confusion matrix* dengan membandingkan hasil dari setiap komponen model terhadap label manual pada kolom *LABEL_ASLI* yang berfungsi sebagai data akurat hasil pengamatan (*ground truth*.) Pada tahap ini, kelas anomali didefinisikan sebagai *positive class* (label = 1), sedangkan kelas normal sebagai *negative class* (label = 0). Evaluasi dilakukan dengan menghitung *confusion matrix* 2x2 (normal vs anomali) untuk setiap aturan dan model, kemudian menurunkan menjadi metrik akurasi, presisi, *recall*, dan *F1-score* dengan menjadikan kelas anomali sebagai *positive class* (**Rachwał, dkk, 2025**). Pendekatan ini mengikuti praktik umum evaluasi *anomaly detection* pada data tidak seimbang, sehingga kinerja model lebih banyak dinilai dari presisi, *recall*, dan *F1-score* kelas minoritas dibanding akurasi *agregat* semata (**Awe, 2020**).

Diterapkan prosedur yang sama pada seluruh komponen arsitektur, aturan berbasis *threshold* melalui kolom *anomali_jarak_masuk_pulang* dan *anomali_total_lintasan*, model *unsupervised isolation forest* melalui kolom *isolation_forest_anomaly*, dan model LSTM melalui kolom *lstm_valid* yang terlebih dahulu dibalik (1 – *lstm_valid*) agar konsisten dengan konversi 1 =

anomali. Selain itu, dihitung pula metrik untuk kombinasi seluruh aturan (*anomali_berdasarkan_aturan*), kombinasi model pembelajaran mesin (*anomali_berdasarkan_model_ml*), dan model hibrida final (*ANOMALI_FINAL*) yang merepresentasikan fusi keputusan antara aturan domain-spesifik dan model statistik/berbasis *deep learning*. Pendekatan multi-lapis dalam penelitian ini sejalan dengan kecenderungan pengembangan *hybrid anomaly detection* pada jaringan dan sistem informasi, di mana beberapa model kombinasi pembelajaran tidak terawasi dan terawasi digabungkan untuk meningkatkan kinerja deteksi (**Chliah, dkk, 2023**).

Secara komputasional, evaluasi dilakukan dengan terlebih dahulu membaca file CSV hasil pelatihan yang memuat *ground truth* (LABEL_ASLI) dan hasil prediksi tiap model/aturan, kemudian menghapus seluruh baris yang belum memiliki label manual (NaN pada LABEL_ASLI) serta menyelaraskan tipe data label dan prediksi menjadi integer. Selanjutnya, untuk setiap model dihitung *confusion matrix* dan diekstrak nilai TN, FP, FN, dan TP, termasuk saat model hanya memprediksi satu kelas, kemudian dari hasil tersebut diturunkan metrik akurasi, presisi, *recall*, *F1-score*, serta *classification report* lengkap sebagai dasar perbandingan kinerja antar komponen.

Evaluasi yang eksplisit terhadap distribusi kelas ekstrem dan bias metrik menjadi kunci agar hasil evaluasi tidak menyesatkan, sebagaimana ditunjukkan oleh (**Zhang, dkk, 2022**) pada *dataset* kredit yang sangat tidak seimbang, *classifier* dengan akurasi tinggi tetap bisa gagal memisahkan transaksi *fraud* dan *non-fraud*, sehingga evaluasi dengan presisi, *recall*, *F1-score*, dan *confusion matrix* menjadi krusial untuk menghindari penilaian yang menyesatkan.

Dengan demikian, keseluruhan prosedur bukan hanya menghasilkan deret angka akurasi, presisi, *recall*, dan *F1-score* untuk setiap aturan dan model, tetapi juga membangun kerangka evaluasi dan validasi yang konsisten dengan menjadikan kelas anomali sebagai *positive class*, menggunakan *confusion matrix* sebagai basis, menekankan metrik spesifik kelas minoritas, mengevaluasi beberapa model sekaligus (termasuk model hibrida), dan secara eksplisit mengantisipasi karakteristik data yang sangat tidak seimbang.

2.10 Pelatihan Akhir dan Penyimpanan Model

Pada tahap pelatihan akhir, dua model utama dikalibrasi yakni *isolation forest* pada fitur *biner* hasil *threshold-based filtering* dan LSTM pada sekuens koordinat yang dianggap normal. Pendekatan tiga lapis menggabungkan penyaringan berbasis ambang, model *unsupervised* untuk pola global dengan model *deep learning* untuk dinamika sekuens spasial, sangat cocok dengan desain model hibrida (**Darban, dkk, 2024**). *Isolation Forest* menerima input vektor [*anomali_jarakmasukpulang*, *anomali_totallintasan*] dan dilatih secara *unsupervised* untuk membedakan pola mayoritas (*inlier*) dan pola jarang (*outlier*) serta menekankan sifat ringan dan dapat melepaskan diri dari ketergantungan pada data belabel (**Geng, dkk, 2025**). Sementara itu, LSTM dilatih hanya pada sekuens yang lolos aturan atau data normal (**Malhotra, dkk, 2016**), dengan sekuens yang telah dinormalisasi dan diseragamkan panjangnya agar proses rekonstruksi menjadi stabil dan tidak bias terhadap rentang nilai tertentu (**Maleki, dkk, 2021**) sebagaimana dijelaskan pada Subbab 2.7. Setelah pelatihan, setiap sekuens direkonstruksi dan dihitung *reconstruction error*-nya, kemudian dibandingkan dengan ambang kuantil 99,5% distribusi *error* data normal.

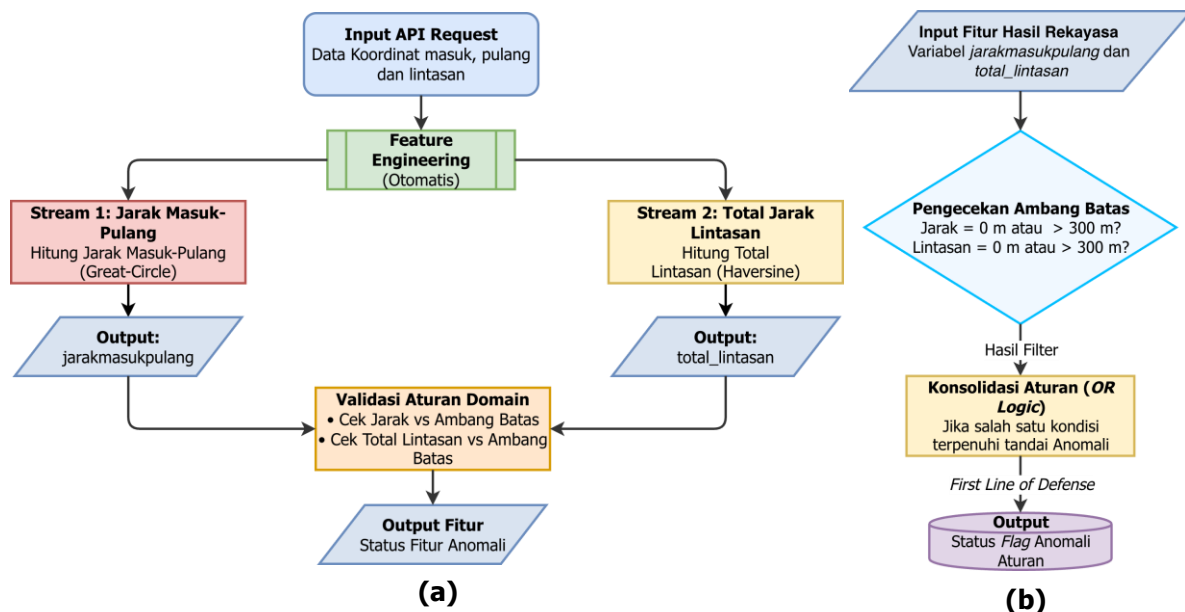
Agar model dapat digunakan kembali pada data baru dan diintegrasikan dalam layanan API, seluruh artefak disimpan secara terpisah, meliputi model *isolation forest*, *scaler* deret waktu LSTM, serta bobot dan arsitektur LSTM. Parameter penting seperti nilai ambang *reconstruction error*, panjang maksimum *sekuens* (*max_seq_length*), dan jumlah fitur juga direkam dalam berkas konfigurasi. Pemisahan antara model, *scaler*, dan konfigurasi ini memudahkan replikasi,

audit, serta pembaruan model tanpa mengubah keseluruhan sistem, sekaligus memastikan bahwa rancangan deteksi anomali dapat dioperasikan kembali pada data presensi baru dalam konteks layanan verifikasi presensi *near real time*, sebagaimana ditekankan dalam kajian-kajian terkini mengenai model *management* yang berjalan di lingkungan produksi (Jain & Noida, 2025).

2.11 Pengujian dan Simulasi *Real time*

Pengujian sistem dilakukan untuk memastikan bahwa metode hibrida *threshold-based filtering*, *isolation forest* dan LSTM yang diimplementasikan menggunakan metode REST API dengan *framework Python Flask*, mampu memberikan keputusan verifikasi presensi secara konsisten, akurat, dan cukup responsif untuk digunakan dalam skenario *real time*. Model yang dimuat meliputi aturan ambang batas jarak masuk-pulang dan total lintasan yang digunakan model *isolation forest* untuk mendeteksi anomali berbasis fitur rule (*anomali jarak masuk-pulang*, *anomali total lintasan*), kemudian model LSTM yang menghitung *error* rekonstruksi lintasan koordinat (*Detect Lokasi*), serta konfigurasi *scaler* dan parameter ambang batas (*threshold*) yang digunakan untuk membedakan valid dan tidak valid.

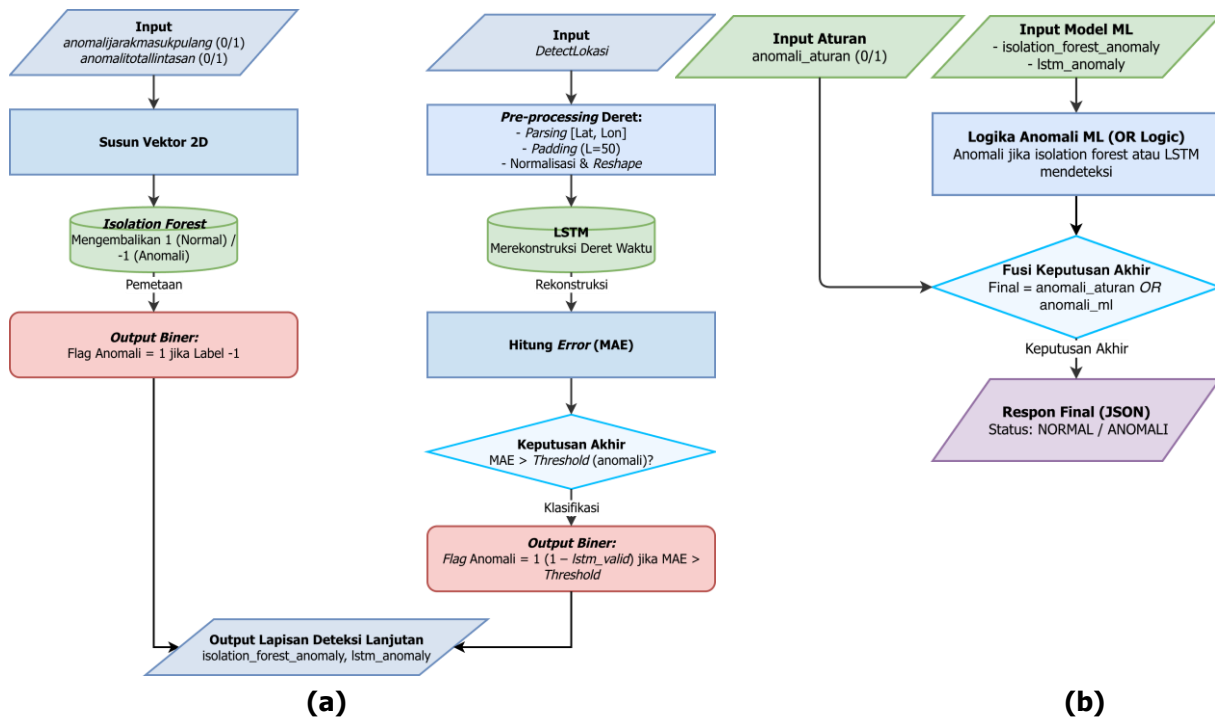
Framework Flask digunakan sebagai server simulasi yang kemudian menjalankan perintah yang memuat seluruh model ke memori pada saat pertama kali dijalankan (*startup*). Strategi ini menurunkan *latency* inferensi karena pemanggilan model dilakukan langsung dari memori tanpa *overhead* pemuatan ulang di setiap permintaan (*request*), sesuai desain umum layanan deteksi anomali *real time* berbasis REST API (Moffitt, 2025). Dengan arsitektur berbasis memori ini, implementasi sistem secara riil dapat dilakukan cukup menggunakan server dengan spesifikasi minimal CPU 4-Core dan RAM 8 GB untuk menangani beban inferensi model hibrida (lstm dan *isolation forest*) tanpa memerlukan GPU khusus. Selain itu, integrasi ke sistem produksi (SIKEP) dirancang bersifat *non-intrusive*, di mana SIKEP cukup mengirimkan HTTP POST Request ke *endpoint* API tanpa memerlukan perombakan struktur *database* inti.



Gambar 4. Alur API Feature Engineering (a) dan Threshold-Based Filtering (b)

Arsitektur pemodelan dan logika fusi keputusan pada pendekatan hibrida ditampilkan pada Gambar 5.

Enhancing Isolation Forest with Threshold-based Filtering and LSTM for Attendance Anomaly Detection



Gambar 5. Alur Proses Model *Isolation Forest*, *LSTM* (a) dan Logika Fusi Keputusan (b)

Setiap permintaan *POST* membawa tiga elemen utama (*LokasiMasuk_clean*, *LokasiPulang_clean*, *DetectLokasi*). Server kemudian secara otomatis melakukan rekayasa fitur, menerapkan *threshold-based filtering* sebagai *first line of defense* yang mengandalkan pengetahuan forensik domain terkait pola pergerakan pegawai yang wajar maupun mencurigakan (Abid, 2025), menjalankan *isolation forest* pada fitur biner, dan menghitung *reconstruction error* LSTM pada sekuens lintasan. LSTM berperan sebagai filter sekuensial yang peka terhadap pola pergerakan yang tidak lazim, bahkan ketika nilai jarak total antar titik tampak masih dalam rentang yang wajar (Hundman, dkk, 2018). Hasil dari ketiga komponen ini digabung dengan logika fusi keputusan untuk menghasilkan respons JSON berisi nilai jarak, skor *error*, bendera anomali per komponen, serta label akhir NORMAL/ANOMALI yang siap diintegrasikan ke SIKEP. Logika fusi keputusan (*Hybrid Decision Logic*) menjadi lapisan terakhir sebelum sistem mengeluarkan status final (Wu, dkk, 2023).

3. HASIL DAN PEMBAHASAN

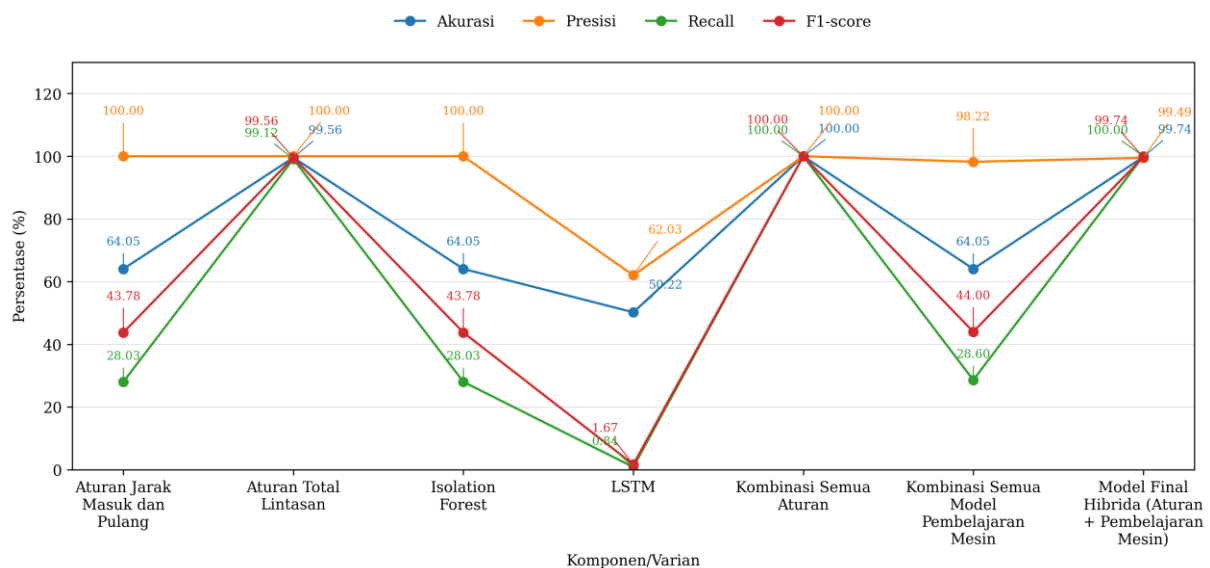
Proses pelatihan dan evaluasi dilakukan dalam rentang waktu Oktober hingga November 2025. Sistem dibangun menggunakan bahasa pemrograman *Python*, implementasi menggunakan pustaka *Scikit-learn* untuk algoritma *Isolation Forest*. Sementara itu, arsitektur LSTM dikembangkan menggunakan *Keras/TensorFlow*. Guna menguji stabilitas performa, eksperimen dilakukan sebanyak 7 (tujuh) kali dengan variasi konfigurasi berbeda. Strategi ini bertujuan untuk menemukan konfigurasi optimal sesuai karakteristik *dataset*. Selain itu, pendekatan ini menjamin akurasi deteksi GPS palsu dan hasil yang representatif. Evaluasi berulang dilakukan untuk menghindari pengaruh fluktuasi hasil dari satu kali percobaan. Hasil pada bagian ini merupakan keluaran konfigurasi final dari percobaan ketujuh.

3.1 Proses Evaluasi dan Validasi Model

Hasil evaluasi dan validasi model terhadap 11.622 *record* presensi yang telah diberikan label manual normal (0) dan anomali (1) dengan kelas anomali diperlakukan sebagai *positive class*, kinerja setiap komponen aturan berbasis *threshold-based filtering*, *isolation forest*, *lstm*, kombinasi antar aturan, kombinasi antar model pembelajaran mesin dan model hibrida final diukur menggunakan *confusion matrix* (*True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)*) dan metrik turunan akurasi, presisi, *recall* dan *F1-score*. Hasil dari evaluasi ditunjukkan pada Tabel 4 di bawah ini.

Tabel 4. Ringkasan Hasil Evaluasi Setiap Komponen

Komponen	TN	FP	FN	TP	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)
Aturan jarak masuk dan pulang 0 m atau > 300 m	5817	0	4178	1627	64.05	100.00	28.03	43.78
Aturan total lintasan 0 m atau > 300 m	5817	0	51	5754	99.56	100.00	99.12	99.56
Isolation Forest	5817	0	4178	1627	64.05	100.00	28.03	43.78
LSTM	5787	30	5756	49	50.22	62.03	0.84	1.67
Kombinasi semua aturan	5817	0	0	5805	100.00	100.00	100.00	100.00
Kombinasi semua Model Pembelajaran Mesin	5787	30	4145	1660	64.05	98.22	28.60	44.00
Model Final Hibrida (aturan + pembelajaran mesin)	5787	30	0	5805	99.74	99.49	100.00	99.74



Gambar 6. Visualisasi Kinerja (Akurasi, Presisi, Recall, F1-Score) Tiap Komponen

Tabel 4 merangkum kinerja setiap komponen dan kombinasi model, kemudian Gambar 6 memperlihatkan tren metriknya secara visual. Aturan jarak masuk–pulang (0 m atau >300 m) mencapai presisi 100% namun *recall* 28,03%. Banyak anomali yang tidak melanggar ambang jarak diklasifikasikan sebagai normal (*false negative*) dan lolos sebagai anomali halus. Anomali halus didefinisikan sebagai *record* berlabel anomali pada verifikasi manual. Namun jarak dan/atau total lintasan masih berada pada rentang normal, sehingga tidak memicu indikator *threshold*. Secara kuantitatif hal ini tercermin dari FN = 4.178 (*recall* 28,03%), pada aturan jarak masuk–pulang. Pola pada grafik juga menunjukkan bahwa komponen tunggal berbasis pembelajaran mesin belum stabil jika berdiri sendiri. *Isolation Forest* mempertahankan presisi tinggi, tetapi *recall* tetap rendah ketika hanya menerima fitur *biner* hasil aturan. LSTM tunggal

menghasilkan metrik terendah pada seluruh varian. Sebaliknya, aturan total lintasan dan kombinasi semua aturan tampak meningkatkan keseimbangan presisi–*recall* secara drastis. Hal ini menunjukkan total lintasan lebih diskriminatif untuk menangkap penyimpangan yang tidak terdeteksi oleh aturan jarak. Tren ini menguat saat fusi keputusan diterapkan pada model hibrida. Fusi aturan dan model pembelajaran mesin mendorong *recall* hingga 100% pada model hibrida final, dengan tetap menjaga presisi tetap tinggi sehingga FP tidak meningkat secara signifikan.

Ketika seluruh aturan digabungkan, sistem mencapai kinerja sempurna pada *dataset* uji. Namun, konfigurasi yang paling relevan secara praktis adalah model hibrida final (aturan + pembelajaran mesin) yang mencapai akurasi 99,74%, presisi 99,49%, *recall* 100%, dan *F1-score* 99,74% dengan *false positive* yang masih sangat rendah. Hal ini menunjukkan bahwa penggabungan *rule-based detection* yang kuat dengan model pembelajaran mesin memberikan keseimbangan terbaik antara sensitivitas terhadap anomali dan stabilitas alarm. Pola serupa terlihat pada berbagai studi *ensemble* dan *hybrid anomaly detection*, di mana model-model murni pembelajaran mesin cenderung membutuhkan pengayaan informasi struktural atau aturan eksplisit untuk mencapai kinerja yang stabil pada kasus berisiko tinggi (Lok, dkk, 2022). Dengan demikian, proses evaluasi dan validasi yang ditunjukkan oleh Tabel 4 memberikan dasar empiris yang kuat bahwa pendekatan hibrida yang diusulkan layak diadopsi sebagai mekanisme utama verifikasi keaslian titik koordinat presensi dalam konteks forensik digital.

3.2 Simulasi Real time

Simulasi *real time* dilakukan dengan skenario *client-server* sederhana. Layanan *Flask* yang memuat model hibrida dijalankan sebagai API yang menerima data presensi dan mengembalikan status NORMAL atau ANOMALI beserta detail jarak dan *error* rekonstruksi. Pengujian dilakukan dengan beberapa set data uji yang merepresentasikan perilaku presensi wajar dan skenario manipulasi lokasi. Pengujian simulasi dilakukan dengan membandingkan data presensi normal (pola pergerakan manusia alami) melawan data simulasi yang meniru perilaku aplikasi manipulator lokasi (*Fake GPS*). Skenario ini mencakup: (1) Teleportasi (perpindahan lokasi instan jarak jauh yang mustahil ditempuh dalam waktu singkat), dan (2) *Mock Location* statis (lokasi diam sempurna tanpa *noise* GPS alami).

```
1 ~ % curl -X POST http://127.0.0.1:5000/verify \  
2 -H "Content-Type: application/json" \  
3 -d '{  
4   "LokasiMasuk_clean": "-6.1702,106.8263",  
5   "LokasiPulang_clean": "-6.1703,106.8264",  
6   "DetectLokasi": "-6.1702,106.8263,0,-6.17025,106.82635,0,  
7   -6.1703,106.8264,0"  
8 }'
```

```
1 {  
2   "anomali_final": 0,  
3   "detail_evaluasi": {  
4     "anomali_aturan": 0,  
5     "anomali_model_ml": 0  
6   },  
7   "penjelasan_aturan": {  
8     "anomali_jarak": 0,  
9     "anomali lintasan": 0,  
10    "jarak_masuk_pulang": "15.68 M",  
11    "total lintasan": "15.68 M"  
12  },  
13  "penjelasan_model": {  
14    "iforest_anomaly": 0,  
15    "lstm_error": 0.01678,  
16    "lstm_threshold_used": 0.035993782877922056,  
17    "lstm_valid": 1  
18  },  
19  "status": "NORMAL"  
20 }
```

Gambar 7. HTTP POST dan Hasil Data Pengujian Normal

Simulasi respons sistem untuk skenario pengujian *spoofing* ditampilkan pada Gambar 8.

```

1 curl -X POST http://127.0.0.1:5000/verify \
2 -H "Content-Type: application/json" \
3 -d '{
4   "LokasiMasuk_clean": "-6.1706884,106.8269907",
5   "LokasiPulang_clean": "-7.2575,112.7521",
6   "DetectLokasi": "-6.1704124,106.8264688,-6.1704124,106.8264688,-6.1704124,106.8264688,
7   -6.1704124,106.8264688,-7.2575,112.7521,-6.1704124,-6.1704124,106.8264688,-7.2575,112.7521,
8   -6.1704124,-6.1704124,106.8264688,18.80620002746582,-6.1704124"
9 }'

```

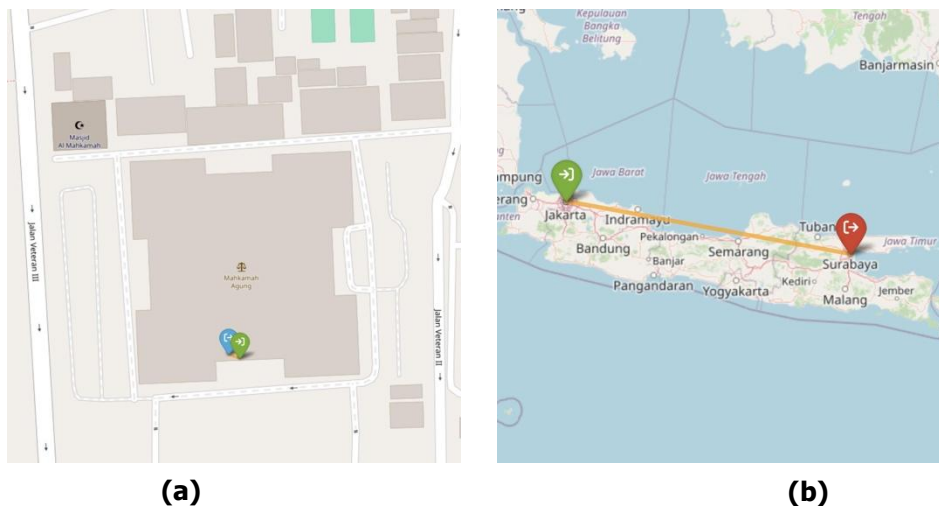
```

1 {
2   "anomali_final": 1,
3   "detail_evaluasi": {
4     "anomali_aturan": 1,
5     "anomali_model_ml": 1
6   },
7   "penjelasan_aturan": {
8     "anomali_jarak": 1,
9     "anomali_lintasan": 1,
10    "jarak_masuk_pulang": "665.38 KM",
11    "total_lintasan": "38842.08 KM"
12  },
13  "penjelasan_model": {
14    "iforest_anomaly": 1,
15    "lstm_error": 0.0505,
16    "lstm_threshold_used": 0.035993782877922056,
17    "lstm_valid": 0
18  },
19  "status": "ANOMALI"
20 }

```

Gambar 8. HTTP POST dan Hasil Data Pengujian *Spoofing*

Pada skenario normal, koordinat masuk–pulang berada di sekitar area kantor dengan pola lintasan yang stabil pada radius kecil. Sistem mengklasifikasikan kasus ini sebagai NORMAL dengan jarak masuk–pulang sekitar belasan meter dan *reconstruction error* LSTM jauh di bawah ambang. Pada skenario *spoofing*, koordinat pulang sengaja digeser jauh, lintasan yang melompat dari Jakarta ke Surabaya dalam satu sesi kerja. Sistem secara konsisten memberikan label ANOMALI, aturan jarak mendeteksi lompatan ratusan kilometer, sementara LSTM menghasilkan *reconstruction error* yang jauh melampaui *threshold*. Untuk memberikan gambaran spasial yang lebih jelas mengenai perbedaan antara pola pergerakan normal dan anomali manipulasi lokasi yang ditunjukkan pada data JSON di atas, visualisasi pemetaan dapat dilihat pada Gambar 9.



Gambar 9. Perbandingan Pola Presensi Normal (a) VS Manipulasi *Fake Gps* (b)

Gambar 9 (a) titik koordinat masuk (hijau) dan pulang (biru) terkluster rapat di area kantor Mahkamah Agung (*zoom* diperbesar), menunjukkan konsistensi lokasi yang valid. Gambar 9 (b) pola anomali (*Spoofing*) simulasi serangan manipulasi lokasi menunjukkan perpindahan drastis (garis oranye) dari Jakarta ke Surabaya dalam satu sesi kerja (*zoom* diperkecil), yang dideteksi sebagai *outlier ekstrem* oleh sistem. Hal ini menunjukkan bahwa arsitektur hibrida tidak hanya unggul pada evaluasi *off-line*, tetapi juga cukup responsif dan dapat dijelaskan dalam konteks layanan verifikasi presensi *near real time*.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa pendekatan hibrida yang menggabungkan *threshold-based filtering*, *isolation forest*, dan LSTM mampu meningkatkan deteksi anomali presensi berbasis GPS. Tabel 4 dan gambar 6 memperlihatkan visualisasi kinerja per komponen, dan menunjukkan bahwa beberapa komponen memiliki akurasi di bawah 90% ketika berdiri sendiri, terutama aturan jarak masuk–pulang dan *isolation forest* yang sensitif terhadap aturan input. Temuan ini menegaskan pentingnya fusi keputusan untuk menutup celah "anomali halus" yang tidak terjaring oleh *threshold* jarak. Model final hibrida mencapai akurasi 99,74%, presisi 99,49%, *recall* 100%, dan *F1-score* 99,74% pada 11.622 *record*. Hasil tersebut relevan untuk simulasi verifikasi *near real time* dan mendukung evaluasi audit serta pembuktian forensik digital.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Biro Kepegawaian Badan Urusan Administrasi Mahkamah Agung RI dan Badan Pengawasan Mahkamah Agung RI dalam proses publikasi penelitian ini.

DAFTAR RUJUKAN

- Abid, B. (2025). *A Scalable Hybrid Approach to Detecting Fraud with Machine Learning*. 1297–1301.
- Aggarwal, S. (2023). *Munich Personal RePEc Archive LSTM based Anomaly Detection in Time Series for United States exports and imports*. (117149).
- Almaghrabi, S., Rana, M., Hamilton, M., & Rahaman, M. S. (2024). Multidimensional dynamic attention for multivariate time series forecasting. *Applied Soft Computing*, 167(PB), 112350. <https://doi.org/10.1016/j.asoc.2024.112350>
- Anam, M. K., Defit, S., Havaluddin, Efrizoni, L., & Firdaus, M. B. (2024). Early Stopping on CNN-LSTM Development to Improve Classification Performance. *Journal of Applied Data Sciences*, 5(3), 1175–1188. <https://doi.org/10.47738/jads.v5i3.312>
- Awe, O. O., (2020). *Computational Strategies for Handling Imbalanced Data in Machine Learning*.
- Bhatt, C., Rawat, A., Chauhan, R., Bhatt, P., Singh, T., & Sharma, S. (2023). GPS Based Automated Attendance System. *Proceedings of IEEE 2023 5th International Conference on Advances in Electronics, Computers and Communications, ICAECC 2023*. <https://doi.org/10.1109/ICAECC59324.2023.10560200>
- Chang, Y.-H., Hu, C.-L., Hwang, Y.-L., Ou, C.-W., & Hsu, F.-H. (n.d.). *Fake GPS Defender: A Server-side Solution to Detect Fake GPS*.

- Chliah, H., Battou, A., hadj, M. A. el, & Laoufi, A. (2023). Hybrid Machine Learning-Based Approach for Anomaly Detection using Apache Spark. *International Journal of Advanced Computer Science and Applications*, 14(4), 870–878. <https://doi.org/10.14569/IJACSA.2023.0140496>
- Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2024). Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Computing Surveys*, 57(1). <https://doi.org/10.1145/3691338>
- Geng, G., Wang, P., Sun, L., & Wen, H. (2025). Enhanced Isolation Forest-Based Algorithm for Unsupervised Anomaly Detection in Lidar SLAM Localization. *World Electric Vehicle Journal*, 16(4). <https://doi.org/10.3390/wevj16040209>
- Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–395. <https://doi.org/10.1145/3219819.3219845>
- Jain, A. K., & Noida, G. (2025). *Machine learning model management in production environments*. 14(1), 149–162.
- Jiang, X., Pang, Y., Li, X., Pan, J., & Xie, Y. (2018). Deep neural networks with Elastic Rectified Linear Units for object recognition. *Neurocomputing*, 275, 1132–1139. <https://doi.org/10.1016/j.neucom.2017.09.056>
- Mahkamah Agung Republik Indonesia. (2022). Keputusan Ketua Mahkamah Agung Republik Indonesia Nomor 368/KMA/SK/XII/2022 tentang pedoman presensi online untuk hakim dan aparatur sipil negara pada Mahkamah Agung dan badan peradilan yang berada di bawahnya melalui aplikasi Sistem Informasi Kepegawaian (Keputusan No. 368/KMA/SK/XII/2022). <https://jdih.mahkamahagung.go.id/legal-product/sk-kma-nomor-368kmaskxii2022/detail>
- Kirichenko, L., Koval, Y., Yakovlev, S., & Chumachenko, D. (2024). Anomaly Detection in Fractal Time Series with LSTM Autoencoders. *Mathematics*, 12(19). <https://doi.org/10.3390/math12193079>
- Lok, L. K., Hameed, V. A., & Rana, M. E. (2022). Hybrid machine learning approach for anomaly detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 27(2), 1016–1024. <https://doi.org/10.11591/ijeecs.v27.i2.pp1016-1024>
- Maleki, Sepehr, Maleki, Sasan, & Jennings, N. R. (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. *Applied Soft Computing*, 108, 107443. <https://doi.org/10.1016/j.asoc.2021.107443>

- Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). *LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection*. <http://arxiv.org/abs/1607.00148>
- Mar'i, F., & Pangestu, G. (2021). Classification of Fake GPS in GOJEK Application using Logistic Regression. *ACM International Conference Proceeding Series*, 94–99. <https://doi.org/10.1145/3479645.3479657>
- NGA. (2014). *National Geospatial-Intelligence Agency (NGA) Standardization Document Department Of Defense Its Definition and Relationships with Local Geodetic Systems*. 207. ftp://ftp.nga.mil/pub2/gandg/website/wgs84/NGA.STND.0036_1.0.0_WGS84.pdf%0Ahttps://www.mendeley.com/catalogue/ed30f6d8-7dde-3ae6-8faa-4aa3f2b50cde/?utm_source=desktop&utm_medium=1.19.8&utm_campaign=open_catalog&userDocumentId=%7B051fcfb8-579d-49c4-beaf-b8
- Nigam, A., & Srivastava, S. (2023). Hybrid deep learning models for traffic stream variables prediction during rainfall. *Multimodal Transportation*, 2(1), 100052. <https://doi.org/10.1016/j.multra.2022.100052>
- Pang, G., Shen, C., Cao, L., & Hengel, A. Van Den. (2022). Deep Learning for Anomaly Detection: A Review. *ACM Computing Surveys*, 54(2), 1–36. <https://doi.org/10.1145/3439950>
- Priyanto, C. Y., Hendry, & Purnomo, H. D. (2021). Combination of Isolation Forest and LSTM Autoencoder for Anomaly Detection. *2021 2nd International Conference on Innovative and Creative Information Technology, ICITech 2021*, (pp. 35–38). <https://doi.org/10.1109/ICITech50181.2021.9590143>
- Rachwał, Alicja, Gałka, Ł., Rachwał, Albert, & Karczmarek, P. (2025). *Aggregating evaluation metrics for anomaly detection: a unified scoring approach*. 19(12), 402–419.
- Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Muller, K. R. (2021). A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109(5), 756–795. <https://doi.org/10.1109/JPROC.2021.3052449>
- Winarno, E., Hadikurniawati, W., & Rosso, R. N. (2017). Location based service for presence system using haversine method. *Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017, 2018-Janua*, 1–4. <https://doi.org/10.1109/INNOCIT.2017.8319153>

- Wu, M., Richard Yu, F., Liu, P. X., & He, Y. (2023). A Hybrid Driving Decision-Making System Integrating Markov Logic Networks and Connectionist AI. *IEEE Transactions on Intelligent Transportation Systems*, 24(3), 3514–3527. <https://doi.org/10.1109/TITS.2022.3227122>
- Yahya, M. A., Moya, A. R., & Ventura, S. (2025). Deep learning for multivariate time series anomaly detection: an evaluation of reconstruction-based methods. *Artificial Intelligence Review*, 58(12). <https://doi.org/10.1007/s10462-025-11401-9>
- Zhang, Y. F., Lu, H. L., Lin, H. F., Qiao, X. C., & Zheng, H. (2022). The Optimized Anomaly Detection Models Based on an Approach of Dealing with Imbalanced Dataset for Credit Card Fraud Detection. *Mobile Information Systems*, 2022. <https://doi.org/10.1155/2022/8027903>
- Zhong, Z., Fan, Q., Zhang, J., Ma, M., Zhang, S., Sun, Y., Lin, Q., Zhang, Y., & Pei, D. (2023). *A Survey of Time Series Anomaly Detection Methods in the AIOps Domain*. (5). <http://arxiv.org/abs/2308.00393>
- Zuo, S., Liu, Y., Zhang, D., Xin, P., & Liu, T. (2021). Detection of GPS Spoofing Attacks Based on Isolation Forest. *2021 IEEE 9th International Conference on Information, Communication and Networks, ICICN 2021*, (pp. 357–361). <https://doi.org/10.1109/ICICN52636.2021.9673863>