



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Analisis Perbandingan Algoritma Random Forest, Support Vector Machine, dan Artificial Neural Network untuk Prediksi Kondisi Jaringan dan Mitigasi Gangguan Jaringan Starlink di PT X

TESIS

Adelia Putri Hanafiah

2409511008

**POLITEKNIK
NEGERI
JAKARTA**

**PROGRAM STUDI MAGISTER TERAPAN
TEKNIK ELEKTRO JURUSAN TEKNIK
ELEKTRO POLITEKNIK NEGERI
JAKARTA
2026**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Analisis Perbandingan Algoritma Random Forest, Support Vector Machine, dan Artificial Neural Network untuk Prediksi Kondisi Jaringan dan Mitigasi Gangguan Jaringan Starlink di PT X

TESIS

**Diajukan sebagai salah satu syarat untuk
memperoleh gelar Magister Terapan**

**POLITEKNIK
NEGERI
JAKARTA**

Adelia Putri Hanafiah

2409511008

**PROGRAM STUDI MAGISTER TERAPAN
TEKNIK ELEKTRO JURUSAN TEKNIK
ELEKTRO POLITEKNIK NEGERI
JAKARTA**

2026



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PENGESAHAN

Tesis ini yang telah diajukan oleh :

Nama : Adelia Putri Hanafiah
NIM : 2409511008
Program Studi : Magister Teknik Elektro
Judul : Analisis Perbandingan Algoritma Random Forest, Support Vector Machine, dan Artificial Neural Networ untuk Prediksi Kondisi Jaringan dan Mitigasi Gangguan Jaringan Starlink di PT X

Telah diuji oleh Tim Penguji dalam sidang Tesis pada hari Senin, tanggal 20 Juli 2026 dan dinyatakan lulus.

Pembimbing I : Dr. Ir. Dewi Yanti Liliana, S.Kom., M.Kom ()

Pembimbing II : Dr. Prihatin Oktivasi, S.Si., M.Si. ()

Penguji I : Dr. Murie Dwiyanti, S. T., M. T. ()

Penguji II : Nana Sutarna, S.T.,M.T. Ph.D ()

Penguji III : Yenniwarti Rafsyam, SST., M.T ()

Depok, 22 Juli 2026

Diketahui Oleh

Ketua Pascasarjana Politeknik Negeri Jakarta



Prof. Dr. Tatun Hayatun Nufus, M.Si.
NIP. 19660416199512200



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tesis ini. Penulisan Tesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Terapan Politeknik.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan Tesis ini, sangatlah sulit bagi penulis untuk menyelesaikan tesis ini. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Dr. Ir. Dewi Yanti Liliana, S.Kom. M. Kom., selaku dosen pembimbing 1 yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan penulis dalam penyusunan tesis ini.
2. Dr. Prihatin Oktivasari, S.Si, M.Si., selaku dosen pembimbing 2 yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan penulis dalam penyusunan tesis ini.
3. Orang Tua dan keluarga penulis yang telah memberikan bantuan dan dukungan material dan moral.
4. Sahabat penulis yang telah banyak membantu dan memberikan semangat dalam menyelesaikan Tesis ini; dan
5. Berlian Satrio Sejati yang telah banyak memberikan semangat dan dukungan dalam menyelesaikan Tesis ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tesis ini membawa manfaat bagi pengembangan ilmu.

Depok, 19 Maret 2026

Penulis



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Analisis Perbandingan Algoritma Random Forest, Support Vector Machine, dan Artificial Neural Network untuk Prediksi Kondisi Jaringan dan Mitigasi Gangguan Jaringan Starlink di PT X

Abstrak

Jaringan Starlink merupakan salah satu teknologi komunikasi satelit Low Earth Orbit (LEO) yang menawarkan cakupan luas dan latensi rendah. Namun, kualitas layanan jaringan masih dipengaruhi oleh kondisi cuaca yang dapat menyebabkan peningkatan delay, jitter, dan packet loss, serta penurunan throughput. Penelitian ini bertujuan untuk membandingkan kinerja algoritma Random Forest (RF), Support Vector Machine (SVM), dan Artificial Neural Network (ANN) dalam memprediksi kondisi jaringan Starlink serta mengimplementasikan model terbaik untuk mendukung mitigasi gangguan jaringan di PT X. Dataset penelitian diperoleh dari integrasi data cuaca Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), yang meliputi curah hujan, kelembapan udara, dan kecepatan angin, dengan data Quality of Service (QoS) jaringan Starlink yang terdiri atas ping, throughput, jitter, dan packet loss. Tahapan penelitian meliputi preprocessing data, pelabelan kondisi jaringan berdasarkan standar TIPHON menjadi kategori Normal, Warning, dan Gangguan, pelatihan model menggunakan algoritma RF, SVM, dan ANN, serta evaluasi menggunakan metrik accuracy, precision, recall, F1-score, dan confusion matrix. Hasil pengujian menunjukkan bahwa algoritma Random Forest memberikan performa terbaik dengan akurasi 99,81%, diikuti oleh Support Vector Machine sebesar 99,03% dan Artificial Neural Network sebesar 98,58%. Model Random Forest kemudian diimplementasikan pada Raspberry Pi yang terintegrasi dengan ESP32, router MikroTik, modem Orbit sebagai jalur komunikasi cadangan, serta Telegram Bot untuk melakukan pemantauan kondisi jaringan secara real-time dan mendukung mekanisme failover otomatis ketika kondisi jaringan diprediksi mengalami critical.

Kata Kunci: Machine Learning, Random Forest, Support Vector Machine, Artificial Neural Network, Starlink, Quality of Service, Failover.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Comparative Analysis of Random Forest, Support Vector Machine, and Artificial Neural Network for Predicting Network Conditions and Mitigating Starlink Network Failures at PT X

Abstract

Starlink is a Low Earth Orbit (LEO) satellite communication technology that provides extensive coverage and low-latency internet connectivity. However, its network performance is still affected by weather conditions, which may increase delay, jitter, and packet loss while reducing throughput. This study aims to compare the performance of Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN) algorithms in predicting Starlink network conditions and to implement the best-performing model to support network disruption mitigation at PT X. The dataset was obtained by integrating weather data from the Indonesian Agency for Meteorology, Climatology, and Geophysics (BMKG), including rainfall, relative humidity, and wind speed, with Starlink Quality of Service (QoS) parameters consisting of ping, throughput, jitter, and packet loss. The research methodology included data preprocessing, network condition labeling based on the TIPHON standard into Normal, Warning, and Disruption categories, model training using RF, SVM, and ANN algorithms, and performance evaluation using accuracy, precision, recall, F1-score, and confusion matrix metrics. The experimental results showed that the Random Forest algorithm achieved the highest performance with an accuracy of 99.81%, followed by Support Vector Machine with 99.03% and Artificial Neural Network with 98.58%. The Random Forest model was then implemented on a Raspberry Pi integrated with an ESP32, a MikroTik router, an Orbit modem as a backup communication link, and a Telegram Bot for real-time network monitoring and automatic failover support when network disruptions were predicted. The implementation results demonstrated that the proposed system was able to accurately classify network conditions and improve the reliability of the Starlink network at PT X.

Keywords: Machine Learning, Random Forest, Support Vector Machine, Artificial Neural Network, Starlink, Quality of Service (QoS), Failover..



Daftar Isi

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PENGESAHAN TESIS	Error! Bookmark not defined.
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.5 Batasan Penelitian	3
1.6 State of the art.....	4
1.7 Novelty.....	4
BAB II Tinjauan Pustaka	6
2.1 Jaringan Starlink	6
2.2 Parameter Kinerja Jaringan (QoS) berdasarkan Tiphon.....	8
2.2.1 Throughput.....	10
2.2.2 Latency	11
2.2.3 Packet Loss	11
2.2.4 Jitter.....	12
2.3 Konsep Machine Learning (ML).....	12
2.3.1 Model Machine Learning.....	13
2.3.2 Proses Machine Learning.....	13
2.4 Prediksi dan Mitigasi Gangguan Jaringan.....	15
2.5 Internet of Things (IoT).....	15
2.5.1 Sensor Hujan (Raindops Sensor).....	16
2.5.2 Raspberry Pi.....	16
2.5.3 Mikrokontroller ESP32.....	16
2.6 Modem GSM (orbit).....	16
2.7 Evaluasi Performa	17
2.7.1 Accuracy (akurasi)	17
2.7.2 Precision (presisi).....	17
2.7.4 F1-Score	18
BAB III METODE PENELITIAN	19
3.1 Lokasi dan Waktu Penelitian.....	19
3.2 Data dan Sumber Data.....	20
3.2.1 Jenis Data.....	21
3.3.2 Sumber Data.....	21
3.3.3 Tahapan Penelitian	23



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.4	Pemodelan Machine Learning.....	24
3.5	Alur Penelitian.....	25
3.5.1	Alur penelitian pre processing data	27
3.5.2	Alur penelitian Process Machine Learning	29
3.6	Diagram Blok	31
3.7	Perancangan Sistem.....	32
3.7.1	Deskripsi Sistem.....	33
3.7.2	Cara Kerja Alat	33
3.7.3	Spesifikasi Alat	34
3.8	Perancangan Hardware	35
BAB IV HASIL DAN PEMBAHASAN		37
4.1	Pengujian Monitoring Jaringan Starlink.....	37
4.1.2	Deskripsi Pengujian	37
4.1.3	Prosedur Pengujian.....	37
4.1.4	Data Hasil Pengujian.....	39
4.2	Process Machine Learning	53
4.2.1	Membagi data training dan testing model Machine Learning.....	53
4.2.2	Model Random Forest.....	54
4.2.3	Model algoritma SVM	58
4.2.4	Model Artificial Neural Network (ANN).....	62
4.3	Perbandingan Performa Model Machine Learning	64
4.4	Pemilihan rekomendasi model machine learning.....	69
4.5	Implementasi Machine Learning pada alat.....	70
4.6	menunjukkan hasil perancangan alat.....	71
4.7	Integrasi Model Random Forest pada alat.....	71
4.8	Hasil implementasi dan pengujian alat.....	72
4.8.1	Pengujian pembacaan sensor.....	72
4.8.2	Pengujian klasifikasi Machine Learning	73
4.9	Hasil pengujian switching jaringan	75
BAB V ANALISA DAN KESIMPULAN.....		79
5.1	Kesimpulan.....	79
5.2	Saran.....	80



Daftar Tabel

Tabel 2. 1 Parameter Intensitas Hujan BMKG	7
Tabel 2. 2 Parameter Perfomansi Throughput	8
Tabel 2. 3 Parameter perfomansi delay	9
Tabel 2. 4 Parameter perfomansi Packet loss	9
Tabel 2. 5 Parameter perfomansi Jitter	10
Tabel 3. 1 Jenis data	21
Tabel 3. 2 Kategori status jaringan normal, warning, gangguan	24
Tabel 3. 3 Tabel Spesifikasi Komponen	34
Tabel 4. 1 Sampel data BMKG	41
Tabel 4. 2 Data ping jaringan starlink	47
Tabel 4. 3 hasil evaluasi model random forest	56
Tabel 4. 4 Hasil Evaluasi SVM	60
Tabel 4. 5 hasil evaluasi model ANN	63
Tabel 4. 6 perbandingan performa model machine learning	65
Tabel 4. 7 hasil pengujian overfitting	68
Tabel 4. 8 Hasil pembacaan sensor	72
Tabel 4. 9 hasil pengujian mitigasi failover jaringan	75

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK
NEGERI
JAKARTA



Daftar Gambar

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 2. 1 perangkat Starlink.....	6
Gambar 3. 1 Grafik ketersediaan data curah hujan	22
Gambar 3. 2 Diagram Alur Penelitian.....	26
Gambar 3. 3 Alur Penelitian Pre Processing Data	28
Gambar 3. 4 Alur penelitian process machine learning	30
Gambar 3. 5 Diagram Blok sistem	31
Gambar 3. 6 Perancangan system	32
Gambar 3. 7 Cara Kerja Alat.....	34
Gambar 3. 8 Perancangan Hardware.....	35
Gambar 4. 1 Data MBKG	40
Gambar 4. 2 import modul dan Inisialisasi dataset BMKG di google collab.	42
Gambar 4. 3 Preprocessing Data BMKG.....	42
Gambar 4. 4 Cleaning Data.....	43
Gambar 4. 5 Tahap statistic dan visual data.....	44
Gambar 4. 6 Aktivasi starlink di PT X.....	45
Gambar 4. 7 hasil test konektivitas bandwidth	46
Gambar 4. 8 Data performa jaringan starlink.....	46
Gambar 4. 9 uji ping konektivitas internet.....	47
Gambar 4. 10 Import dataset jaringan starlink.....	48
Gambar 4. 11 Inisiasi dataset starlink	49
Gambar 4. 12 Pendataan labeling.....	49
Gambar 4. 13 Analisis parameter QoS.....	50
Gambar 4. 14 Diagram klasifikasi label.....	51
Gambar 4. 15 Penggabungan data BMKG dan data starlink	52
Gambar 4. 16 pembagian training dan testing	53
Gambar 4. 17 Hasil training data Random Forest.....	55
Gambar 4. 18 Hasil visual confusion matrix Random Forest	57
Gambar 4. 19 menunjukkan hasil training model SVM	58
Gambar 4. 20 Visual confusion matrix SVM	61
Gambar 4. 21 menunjukkan training model ANN	62
Gambar 4. 22 menunjukkan grafik hasil klasifikasi algoritma machine learning..	66
Gambar 4. 23 perbandingan overfitting model machine learning.....	69
Gambar 4. 24 Hasil perancangan alat	71
Gambar 4. 25 konfigurasi model di alat.....	72
Gambar 4. 26 Hasil klasifikasi model RF pada Telegram	74



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi menuntut penyedia layanan untuk menghadirkan konektivitas internet yang andal, berkecepatan tinggi, dan memiliki ketersediaan (availability) yang tinggi. Salah satu infrastruktur yang kini menjadi solusi utama di wilayah dengan tantangan geografis adalah jaringan satelit *Low Earth Orbit* (LEO) seperti Starlink [1]. Hal ini sebagaimana di PT X yang telah memanfaatkan jaringan internet berbasis starlink untuk menyediakan layanan internet korporat, untuk mendukung sistem operasional publik[2]. Namun, kondisi saat ini menunjukkan bahwa gangguan jaringan masih sering terjadi, kondisi lingkungan ekstreme terutama fenomena *rain fade* (redaman hujan) yang menjadi karakteristik utama gangguan pada frekuensi *Ku band* yang digunakan satelit LEO. Frekuensi waktu yang mengalami gangguan tinggi ini menimbulkan tantangan serius, terlebih karena jumlah tim teknis yang terbatas, sehingga proses perbaikan sering memerlukan waktu lebih lama. Pendekatan yang digunakan di PT X lebih banyak bersifat reaktif, gangguan ditangani setelah terjadi. permasalahan yang muncul adalah belum adanya sistem yang mampu memprediksi potensi gangguan dan melakukan mitigasi lebih awal sebelum gangguan berdampak luas. Akibatnya, perusahaan berpotensi menghadapi kerugian finansial, kehilangan pelanggan, serta menurunnya citra sebagai penyedia layanan yang andal. Untuk menjawab permasalahan, dibutuhkan sebuah solusi berbasis teknologi cerdas. Oleh karena itu, diperlukan sebuah sistem mitigasi yang lebih cerdas dan proaktif. Dengan memanfaatkan sensor IoT untuk memantau kondisi lingkungan secara *real-time* dan algoritma *Machine Learning* untuk memprediksi gangguan sebelum terjadi *diskoneksi*, sistem dapat melakukan pemindahan jalur data (*failover*) ke



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

jaringan GSM secara otomatis [3]. Dengan sistem prediktif berbasis ML di PT X dapat menerapkan strategi preventif, mengurangi ketergantungan pada metode reaktif, dan mengoptimalkan penggunaan sumber daya teknis yang terbatas[4].

Jika penelitian sebelum nya melakukan penelitian mengenai *implementasi kecerdasan buatan untuk deteksi dini gangguan jaringan* yang memanfaatkan data simulasi untuk memprediksi gangguan redaman. Namun, penelitian tersebut belum mengintegrasikan sistem prediksi dengan tindakan mitigasi otomatis[5].

Penelitian ini bertujuan implementasi *Machine Learning* secara spesifik untuk prediksi dan mitigasi gangguan pada jaringan berbasis starlink di PT X. Jika penelitian terdahulu lebih banyak berfokus pada deteksi gangguan di jaringan nirkabel atau monitoring QoS, maka penelitian ini sebagai pendekatan prediksi dan mitigasi menggunakan *Machine Learning* dengan konteks pada jaringan berbasis starlink di PT X . Dengan demikian, hasil penelitian ini membantu monitoring dan laporan proaktif ke pada pelanggan di PT X sehingga dapat meningkatkan keandalan jaringan dan kepuasan pelanggan.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang diatas, dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana penerapan metode *Machine Learning* dapat digunakan untuk memprediksi potensi gangguan pada jaringan berbasis starlink di PT X?
2. Bagaimana perbandingan kinerja algoritma Random Forest, Support Vector Machine (SVM), dan Artificial Neural Network (ANN) dalam mengklasifikasikan kondisi jaringan Starlink di PT X berdasarkan nilai accuracy, precision, recall, dan F1-score?
3. Bagaimana hasil implementasi system prediksi dan mitigasi gangguan berdasarkan algoritma machine learning dengan performa terbaik pada jaringan starlink di PT X?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.3 Tujuan

1. Menerapkan dan membandingkan algoritma Random Forest, Support Vector Machine (SVM), dan Artificial Neural Network (ANN) dalam mengklasifikasikan kondisi jaringan Starlink berdasarkan parameter Quality of Service (QoS).
2. Menganalisis dan membandingkan kinerja algoritma Random Forest, Support Vector Machine (SVM), dan Artificial Neural Network (ANN) dalam mengklasifikasikan kondisi jaringan Starlink di PT X berdasarkan metrik accuracy, precision, recall, dan F1-score untuk menentukan algoritma dengan performa terbaik.
3. Mengimplementasikan machine learning mekanisme prediksi dan mitigasi gangguan jaringan Starlink menggunakan algoritma Machine Learning dengan performa terbaik melalui proses failover ke jaringan cadangan di PT X.

1.4 Manfaat Penelitian

Adapun manfaat dari pembuatan penelitian ini adalah sebagai berikut :

1. Memberikan model prediksi dan mitigasi berbasis *Machine Learning* yang dapat digunakan untuk mendeteksi potensi gangguan di PT X
2. Membantu PT X mengurangi ketergantungan pada metode reaktif, dan mengoptimalkan penggunaan sumber daya teknis yang terbatas.

1.5 Batasan Penelitian

Agar penelitian lebih terarah dan terukur, maka ruang lingkup penelitian ini dibatasi pada hal-hal berikut:

1. Penelitian hanya dilakukan pada jaringan starlink yang digunakan untuk layanan internet.
2. Parameter data yang digunakan pada penelitian ini meliputi parameter *Quality of Service (QoS)*, yaitu *latency (ping)*, *throughput*, *jitter*, dan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

packet loss, serta data cuaca BMKG yang terdiri atas curah hujan, kelembapan udara, dan kecepatan angin. Pada tahap implementasi, sistem menggunakan sensor curah hujan sebagai masukan data cuaca secara *real time*.

3. Perancangan dan pembuatan model machine learning dengan menggunakan pemrograman bahasa *python*.
4. Evaluasi hasil model hanya mencakup aspek akurasi prediksi dan efektivitas mitigasi gangguan berdasarkan data uji dari PT X, tidak mencakup aspek biaya implementasi sistem secara menyeluruh.

1.6 State of the art

Perkembangan teknologi komunikasi satelit orbit rendah (*Low Earth Orbit*) dan metode kecerdasan buatan dalam pemrosesan data jaringan. Yussuff dkk. (2023) dalam penelitiannya yang berjudul "*Adoption of Deep Neural Network Model for the Prediction of Rain Attenuation*" bahwa penggunaan model AI, khususnya *Deep Neural Network* (DNN), memiliki performa yang jauh lebih unggul dibandingkan model tradisional (seperti model ITU-R).

Penelitian tersebut membuktikan bahwa pola non-linear dari redaman hujan paling efektif dipetakan menggunakan arsitektur jaringan saraf tiruan. Namun, penelitian ini terbatas pada akurasi prediksi nilai redaman dan tidak ada mitigasi dari hasil prediksi.

Penelitian mengenai fondasi infrastruktur monitoring telah dilakukan oleh Atarian dkk. (2026) dalam jurnal J-PTIHK, yang memvalidasi keandalan perangkat *edge gateway* berbasis Raspberry Pi untuk melakukan akuisisi data *Quality of Service* (QoS) Starlink secara *real-time*. Di sisi lain, Pramukantoro dkk. (2025) dalam jurnal JTIHK telah mengimplementasikan algoritma *Machine Learning* (khususnya *Random Forest*) untuk klasifikasi kualitas sinyal satelit LEO. Namun, terdapat celah (*gap*) di mana penelitian tersebut masih mengandalkan log internal sistem tanpa melibatkan input kondisi lingkungan fisik



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

secara langsung.

Alifi dan Ichsan (2026) dalam jurnal AVITEC menjelaskan bahwa mekanisme *failover* pada jaringan *Dual-WAN* umumnya masih bersifat reaktif, di mana perpindahan jalur internet baru terjadi setelah koneksi utama terputus sepenuhnya. Hal ini diperkuat oleh studi kasus Distiradkk. (2025) dalam jurnal JTOS yang menunjukkan bahwa degradasi sinyal Starlink akibat hujan terjadi secara bertahap, namun belum ada otomatisasi yang mampu mengambil keputusan sebelum kegagalan koneksi (*downtime*) terjadi.

2.7 Novelty

Kebaruan (*novelty*) pada penelitian ini terletak pada penerapan dan perbandingan tiga algoritma *Machine Learning*, yaitu *Random Forest*, *Support Vector Machine* (SVM), dan *Artificial Neural Network* (ANN) dalam mengklasifikasikan kondisi jaringan Starlink menggunakan dataset yang diintegrasikan dari data historis *Quality of Service* (QoS) jaringan Starlink dan data cuaca BMKG. Selain membandingkan performa algoritma berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score*, penelitian ini juga mengimplementasikan algoritma dengan performa terbaik pada sistem monitoring berbasis Raspberry Pi yang terintegrasi dengan ESP32, router MikroTik, modem Orbit, dan Telegram Bot untuk mendukung mekanisme *failover* otomatis pada jaringan Starlink di PT X.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V

ANALISA DAN KESIMPULAN

5.1 Kesimpulan

Penelitian ini bertujuan membandingkan kinerja algoritma Random Forest, Support Vector Machine (SVM), dan Artificial Neural Network (ANN) dalam mengklasifikasikan kondisi jaringan sebagai dasar penerapan mitigasi gangguan pada jaringan Starlink di PT X. Penelitian menggunakan data historis *Quality of Service* (QoS) jaringan Starlink yang diintegrasikan dengan data cuaca BMKG periode Januari 2025 hingga Juni 2026. Tahap implementasi sistem dilakukan di Depok, Jawa Barat, dengan total periode implementasi selama enam bulan, yaitu pada rentang Februari hingga Juli 2026. Pengujian dilakukan secara berkala dan tidak secara kontinu, dengan menyesuaikan kondisi cuaca serta ketersediaan perangkat operasional yang digunakan dalam penelitian. Berdasarkan hasil penelitian, diperoleh beberapa kesimpulan sebagai berikut:

1. Metode Machine Learning diterapkan menggunakan 12.984 data hasil penggabungan data BMKG dan data *Quality of Service* (QoS) jaringan Starlink yang terdiri atas parameter kelembapan (RH_AVG), curah hujan (RR), delay (Ping), throughput, jitter, dan packet loss. Dataset dibagi menjadi 10.387 data latih (80%) dan 2.597 data uji (20%), kemudian dilakukan pelatihan menggunakan tiga algoritma Machine Learning, yaitu Random Forest, Support Vector Machine (SVM), dan Artificial Neural Network (ANN). Berdasarkan hasil evaluasi, algoritma Random Forest dipilih sebagai model terbaik karena memiliki performa paling tinggi dibandingkan SVM dan ANN dalam mengklasifikasikan kondisi jaringan menjadi normal, *warning*, dan *critical*. Oleh karena itu, model *Random Forest* diimplementasikan pada sistem monitoring sebagai dasar pengambilan keputusan dalam memprediksi potensi gangguan jaringan secara real-time dan mendukung proses mitigasi gangguan.
2. Berdasarkan hasil evaluasi, algoritma Random Forest memperoleh accuracy sebesar 99,81%, sehingga menjadi model dengan performa terbaik dibandingkan algoritma Support Vector Machine (SVM) yang memperoleh



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

99,03% dan Artificial Neural Network (ANN) sebesar 98,58%. Pada evaluasi tiap kelas, Random Forest memperoleh precision sebesar 0,99, recall sebesar 0,98, dan F1-score sebesar 0,99 untuk kelas Gangguan, precision, recall, dan F1-score sebesar 1,00 untuk kelas Normal, serta precision sebesar 0,95, recall sebesar 0,99, dan F1-score sebesar 0,97 untuk kelas Warning. Hasil tersebut menunjukkan bahwa Random Forest memiliki kemampuan klasifikasi yang lebih baik dan lebih stabil dibandingkan algoritma SVM dan ANN.

Berdasarkan hasil evaluasi algoritma model random forest dengan performa terbaik berhasil diimplementasikan pada sistem prediksi dan mitigasi gangguan jaringan Starlink. Model digunakan untuk menganalisis kondisi jaringan, sedangkan sensor hujan berfungsi sebagai early warning. Hasil analisis ditampilkan melalui Telegram sebagai media monitoring, dan sistem mampu menjalankan mekanisme failover ke jaringan Orbit serta failback ke Starlink secara otomatis sesuai kondisi jaringan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut.

1. Menambahkan parameter yang memengaruhi kualitas jaringan, seperti suhu udara, kecepatan angin, intensitas sinyal (RSSI/SNR jika tersedia), atau parameter jaringan lainnya sehingga model Machine Learning dapat mempelajari kondisi jaringan secara lebih komprehensif.
2. Melakukan pengambilan data dalam periode yang lebih panjang dan pada berbagai kondisi cuaca sehingga dataset menjadi lebih beragam dan kemampuan generalisasi model dapat ditingkatkan.
3. Membandingkan algoritma Machine Learning lainnya, seperti XGBoost, LightGBM, atau metode Deep Learning, untuk mengetahui apakah terdapat peningkatan performa dibandingkan algoritma Random Forest.
4. Mengembangkan sistem monitoring berbasis dashboard web atau aplikasi mobile, sehingga proses pemantauan kondisi jaringan, hasil prediksi, riwayat gangguan, dan proses switching dapat dilakukan secara real-time dengan antarmuka yang lebih mudah digunakan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Referensi

- [1] Telkom University Surabaya. (2025, May 8). *Low Earth Orbit (LEO) Satellite: Solusi internet global tanpa blank spot*. <https://bte-sby.telkomuniversity.ac.id/low-earth-orbit-leo-satellite-solusi-internet-global-tanpa-blank-spot/> *Low Earth Orbit (LEO) Satellite: Solusi Internet Global Tanpa Blank Spot*
- [2] PLN Icon Plus <https://plniconplus.co.id/>
- [3] Li, X., Zhang, Y., & Chen, L. (2023). Machine learning-based fault detection in optical networks. *IEEE Access*, 11(2), 17492–17505. <https://doi.org/10.1109/ACCESS.2023.3246578>
- [4] Rahayu, D., Santoso, I., & Kurniawan, A. (2022). *Implementasi kecerdasan buatan untuk deteksi dini gangguan jaringan optik*. *Jurnal Teknologi Telekomunikasi*, 9(1), 45–53. <https://doi.org/10.4018/JITR.2021070102>
- [5] Rahayu, D., Santoso, I., & Kurniawan, A. (2022). Implementasi kecerdasan buatan untuk deteksi dini gangguan jaringan optik. *Jurnal Teknologi Telekomunikasi*, 9(1), 45–53.
- [6] Kasat, N., et al. (2023). *Characterizing Starlink's Performance under Different Weather Conditions*. arXiv preprint.
- [7] Kasat, N., et al. (2023). *Characterizing Starlink's Performance under Different Weather Conditions*. arXiv preprint
- [8] Effendi, Harlan Analisa Pengaruh Redaman Hujan Terhadap Kualitas Sinyal Terima Modem Comtech CDM 600 Pada Jaringan Komunikasi Satelit DOI: <https://doi.org/10.37277/stch.v26i1.74>
- [9] Syafian Putra, ANALISIS ALGORITMA MACHINE LEARNING KNEAREST NEIGHBOR(KNN) PADA QUALITY OF SERVICE JARINGAN 5G
- [10] F. Musumeci, C. Rottondi, and M. Tornatore, “A survey on machine learning for optical network management,” *IEEE Communications Surveys & Tutorials*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- [11] A. Sharma and S. K. Sahu, “Supervised learning approaches for intelligent fault prediction in optical fiber communication,” *IEEE Access*, vol. 9, pp. 176548–176560, 2021, doi: 10.1109/ACCESS.2021.3132536.
- [12] M. A. Khan, M. S. Islam, and M. R. Hossain, “Unsupervised learning approach for anomaly detection in optical communication networks using K-Means clustering,” *IEEE Access*, vol. 11, pp. 105412–105424, 2023, doi: 10.1109/ACCESS.2023.3302519.
- [13] R. Hidayat, A. Santoso, dan D. Rahayu, “Penerapan reinforcement learning untuk optimasi jaringan komunikasi nirkabel berbasis QoS,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 2, pp. 101–110, 2022.
- [14] M. R. Hidayat dan N. Firmansyah, “Penerapan algoritma Random Forest untuk prediksi gangguan jaringan komunikasi berbasis data historis,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 10, no. 3, pp. 211–218, 2024, doi: 10.22146/jnteti.82437.
- [15] R. Hidayat, A. Santoso, dan D. Rahayu, “Klasifikasi gangguan jaringan fiber optik menggunakan algoritma Support Vector Machine,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 2, pp. 101–110, 2022.
- [16] D. Pratama, A. Kurniawan, dan R. Hidayat, “Implementasi Artificial Neural Network untuk prediksi gangguan jaringan fiber optik berbasis data performansi QoS,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 3, pp. 121–130, 2022.
- [17] M. R. Hidayat dan N. Firmansyah, “Analisis kinerja model machine learning menggunakan confusion matrix pada sistem klasifikasi jaringan,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 10, no. 4, pp. 315–322, 2024.
- [18] D. Pratama dan A. Wibowo, “Evaluasi model klasifikasi menggunakan precision, recall, dan F1-score pada sistem deteksi



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

gangguan jaringan,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 3, pp. 123–130, 2022.

- [19] F. Musumeci, C. Rottondi, and M. Tornatore, “Performance evaluation of ML models for fault management in optical networks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 3406–3433, 2020.
- [20] D. Rahayu, I. Santoso, dan A. Kurniawan, “Implementasi kecerdasan buatan untuk deteksi dini dan mitigasi gangguan jaringan optik,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 2, pp. 99–108, 2022.
- [21] S. Irawan, D. Pratama, dan A. R. Wibowo, “Desain dan implementasi sistem manajemen jaringan (NMS) berbasis FCAPS pada jaringan optik,” *Jurnal Teknologi Telekomunikasi*, vol. 9, no. 3, pp. 111–120, 2022.
- [22] X. Li, L. Chen, and M. Yang, “Machine learning-enhanced network management for optical systems,” *IEEE Communications Magazine*, vol. 61, no. 4, pp. 98–104, 2023.
- [23] Marte Ardhianto, M., & Sumarwanto, R. (2022). Analisa Predictive Berbasis Supervised Machine Learning Terhadap Kerusakan Peralatan Pembangkit. *Jurnal Teknologi Elekterika*, 19(2), 143. <https://doi.org/10.31963/elekterika.v6i2.3690>



1. Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.ensemble import
RandomForestClassifier
from sklearn.svm import SVC
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import joblib
```

2. Upload Data Set

```
from google.colab import files
uploaded = files.upload()
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5. Membaca data set

```
data = pd.read_excel("Dataset_Label (2)
final(1).xlsx")

data.head()
```

6. Informasi data set

```
print(data.info())
print(data.describe())
print(data.isnull().sum())
```

5. Hapus missing value

```
data = data.dropna()
```

6. Hapus data duplikat

```
data = data.drop_duplicates()
```

7. Data labelling

```
def label_tiphon(row):
    # Delay
    if row['Ping_ms'] >= 300:
        delay = "Gangguan"
    elif row['Ping_ms'] >= 150:
        delay = "Warning"
    else:
        delay = "Normal"

    # Throughput
    if row['Throughput_%'] <= 25:
        throughput = "Gangguan"
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
elif row['Throughput_%'] <= 50:
    throughput = "Warning"
else:
    throughput = "Normal"

# Jitter
if row['Jitter_ms'] > 125:
    jitter = "Gangguan"
elif row['Jitter_ms'] > 75:
    jitter = "Warning"
else:
    jitter = "Normal"

# Packet Loss
if row['Packet_Loss'] > 25:
    packetloss = "Gangguan"
elif row['Packet_Loss'] >= 15:
    packetloss = "Warning"
else:
    packetloss = "Normal"

# Label akhir berdasarkan kondisi terburuk
kategori = [delay, throughput, jitter,
packetloss]

if "Gangguan" in kategori:
    return "Gangguan"
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
elif "Warning" in kategori:
```

```
    return "Warning"
```

```
else:
```

```
    return "Normal"
```

```
# Membuat label
```

```
data['Label'] = data.apply(label_tiphon, axis=1)
```

```
# Menampilkan hasil
```

```
print(data[['Ping_ms', 'Throughput_%', 'Jitter_ms', 'Packet_Loss', 'Label']].head())
```

8. Visualisasi data distribusi parameter QoS

```
plt.figure(figsize=(14,10))
# Delay (Ping)
plt.subplot(2,2,1)
plt.hist(data["Ping_ms"], bins=30)
plt.title("Distribusi Delay (Ping)")
plt.xlabel("Delay (ms)")
plt.ylabel("Frekuensi")

# Throughput
plt.subplot(2,2,2)
plt.hist(data["Throughput_Mbps"], bins=30)
plt.title("Distribusi Throughput")
plt.xlabel("Throughput (Mbps)")
plt.ylabel("Frekuensi")

# Jitter
plt.subplot(2,2,3)
plt.hist(data["Jitter_ms"], bins=30)
plt.title("Distribusi Jitter")
plt.xlabel("Jitter (ms)")
plt.ylabel("Frekuensi")

# Packet Loss
plt.subplot(2,2,4)
plt.hist(data["Packet_Loss"], bins=30)
plt.title("Distribusi Packet Loss")
```

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
plt.xlabel("Packet Loss (%)")
plt.ylabel("Frekuensi")
plt.tight_layout()
plt.show()
```

Normalisasi Data

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

10. Split Data

```
from sklearn.model_selection import train_test_split

# Fitur
X = data[['RH_AVG',
          'RR',
          'FF_AVG',
          'Ping_ms',
          'Throughput_Mbps',
          'Jitter_ms',
          'Packet_Loss']]

# Label
y = data['Label']

# Split data 80% training dan 20% testing
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.20,
    random_state=42,
    stratify=y
)

# Menampilkan jumlah data
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("Jumlah Data :", len(data))
print("Data Training :", len(X_train))
print("Data Testing :", len(X_test))
```

11. Machine Learning Random Forest

```
rf = RandomForestClassifier(
    n_estimators=100,
    random_state=42

rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
print("Accuracy :", accuracy_score(y_test, rf_pred))
print(classification_report(y_test, rf_pred))

print(confusion_matrix(y_test, rf_pred))
```

12. Machine Learning SVM

```
svm = SVC()

svm.fit(X_train, y_train)

svm_pred = svm.predict(X_test)

print("Accuracy :", accuracy_score(y_test, svm_pred))

print(classification_report(y_test, svm_pred))

print(confusion_matrix(y_test, svm_pred))
```

13. Machine Learning ANN

```
ann = Sequential()

ann.add(Dense(32, input_dim=X_train.shape[1], activation=
'relu'))

ann.add(Dense(16, activation='relu'))
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
ann.add(Dense(3,activation='softmax'))

ann.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])

history = ann.fit(
    X_train,
    y_train,
    epochs=50,
    batch_size=32,
    validation_split=0.2
)

loss,accuracy = ann.evaluate(X_test,y_test)

print("Accuracy :",accuracy)
ann_pred = ann.predict(X_test)

ann_pred = np.argmax(ann_pred,axis=1)

print(classification_report(y_test,ann_pred))

print(confusion_matrix(y_test,ann_pred))
```

14. Perbandingan Model

```
hasil = pd.DataFrame({
    'Model':['Random Forest',
            'SVM',
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
'ANN'],  
Accuracy': [  
    accuracy_score(y_test, rf_pred),  
    accuracy_score(y_test, svm_pred),  
    accuracy_score(y_test, ann_pred)  
)  
print(hasil)
```

5. Grafik Perbandingan Model

```
import matplotlib.pyplot as plt  
import numpy as np  
  
model = ['Random Forest', 'SVM', 'ANN']  
  
train = [1.0000, 0.9919, 0.9920]  
test = [0.9988, 0.9904, 0.9908]  
  
x = np.arange(len(model))  
width = 0.35  
  
plt.figure(figsize=(8,5))  
  
plt.bar(x-width/2, train, width, label='Training Accuracy')  
plt.bar(x+width/2, test, width, label='Testing Accuracy')  
  
plt.xticks(x, model)  
plt.ylabel('Accuracy')  
plt.ylim(0.95,1.01)  
plt.title('Perbandingan Training Accuracy dan Testing  
Accuracy')  
plt.legend()  
  
plt.show()
```

16. Konfigurasi Raspberry Pi

```
import json  
import time
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import csv
import os
import re
import subprocess
import threading
from datetime import datetime
from zoneinfo import ZoneInfo

import joblib
import requests
import paho.mqtt.client as mqtt

from config import (
    MQTT_BROKER,
    MQTT_PORT,
    MQTT_TOPIC,
    MQTT_CLIENT_ID,
    TELEGRAM_BOT_TOKEN,
    TELEGRAM_CHAT_ID,
    MIKROTIK_IP,
    MIKROTIK_USER,
    MIKROTIK_PASS,
    MIKROTIK_SCRIPT_TO_ORBIT,
    MIKROTIK_SCRIPT_TO_STARLINK,
    MODEL_PATH,
    LOG_PATH,
    SWITCH_DELAY_SECONDS,
    DRY_RAIN_THRESHOLD,
    ENABLE_MIKROTIK_SWITCH,
    FEATURE_COLUMNS,
)

# =====
# BASIC SETTING
# =====

TIMEZONE = ZoneInfo("Asia/Jakarta")

NETWORK_CHECK_INTERVAL = 60

PACKET_LOSS_WARNING = 20
PACKET_LOSS_CRITICAL = 50

ACTIVE_CHECK_IP = "8.8.8.8"
STARLINK_CHECK_IP = "1.1.1.1"
STARLINK_OK_LOSS_MAX = 15
FAILBACK_STARLINK_REQUIRED = 2
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
LAST_SENSOR_PATH = "data/last_sensor.json"
QUOTA_SCRIPT_PATH =
"/home/bibaku/rain_ml_failover_v2/quota_orbit.py"

STARLINK_GATEWAY = "192.168.1.1"
ORBIT_GATEWAY = "192.168.8.1"
STARLINK_CONNECTED_NET = "192.168.1.0/24"

# =====
# ICON
# =====

ICON_NORMAL = "\U00002705"
ICON_WARNING = "\U000026A0"
ICON_CRITICAL = "\U0001F6A8"

ICON_HUMIDITY = "\U0001F4A7"
ICON_RAIN = "\U0001F327"
ICON_SENSOR = "\U0001F4DF"
ICON_RSSI = "\U0001F4E1"
ICON_PING = "\U0001F4F6"
ICON_PACKET = "\U0001F4C9"
ICON_TIME = "\U000023F1"

# =====
# GLOBAL STATUS
# =====

current_network_state = "STARLINK"

last_notified_target = ""
last_notified_status = ""
last_notified_weather = ""
last_notified_network_state = ""

starlink_normal_counter = 0

last_sensor_data = {
    "rain_value": None,
    "humidity": None,
    "level_esp": None,
    "rssi": None
}

process_lock = threading.Lock()
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# =====
# HELPER
# =====

def safe_int(value, default=0):
    try:
        if value is None or value == "":
            return default
        return int(float(value))
    except Exception:
        return default

def now_text():
    return datetime.now(TIMEZONE).strftime("%Y-%m-%d %H:%M:%S
WIB")

def now_log():
    return datetime.now(TIMEZONE).strftime("%Y-%m-%d %H:%M:%S")

def mikrotik_ssh(command_text, timeout=15):
    command = [
        "sshpass",
        "-p",
        MIKROTIK_PASS,
        "ssh",
        "-o",
        "StrictHostKeyChecking=no",
        "-o",
        "ConnectTimeout=10",
        f"{MIKROTIK_USER}@{MIKROTIK_IP}",
        command_text
    ]

    result = subprocess.run(
        command,
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        text=True,
        timeout=timeout
    )

    output = result.stdout + "\n" + result.stderr
    return result.returncode, output
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def get_route_flags(line):
    parts = line.strip().split()

    if not parts:
        return ""

    if parts[0].isdigit() and len(parts) > 1:
        return parts[1]

    return parts[0]

def is_active_route_line(line):
    flags = get_route_flags(line)
    return "A" in flags

# =====
# SENSOR LAST DATA
# =====

def load_last_sensor_data():
    global last_sensor_data

    try:
        if not os.path.exists(LAST_SENSOR_PATH):
            print("Last sensor data belum ada")
            return

        with open(LAST_SENSOR_PATH, "r") as file:
            data = json.load(file)

        last_sensor_data = {
            "rain_value": data.get("rain_value", None),
            "humidity": data.get("humidity", None),
            "level_esp": data.get("level_esp", None),
            "rssi": data.get("rssi", None)
        }

        print("Last sensor data loaded:", last_sensor_data)

    except Exception as e:
        print("Load last sensor error:", e)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def save_last_sensor_data():
    try:
        folder = os.path.dirname(LAST_SENSOR_PATH)

        if folder and not os.path.exists(folder):
            os.makedirs(folder)

        with open(LAST_SENSOR_PATH, "w") as file:
            json.dump(last_sensor_data, file)

        print("Last sensor data saved:", last_sensor_data)

    except Exception as e:
        print("Save last sensor error:", e)

# =====
# LOAD MODEL
# =====

try:
    model = joblib.load(MODEL_PATH)
    print("Model loaded:", MODEL_PATH)
    print("Feature columns:", FEATURE_COLUMNS)

except Exception as e:
    print("Gagal load model:", e)
    model = None

# =====
# TELEGRAM
# =====

def send_telegram(message):
    if not TELEGRAM_BOT_TOKEN or not TELEGRAM_CHAT_ID:
        print("Telegram belum dikonfigurasi di .env")
        return

    url =
f"https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendMessage"

    try:
        response = requests.post(
            url,
            data={
                "chat_id": TELEGRAM_CHAT_ID,
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        "text": message
    },
    timeout=10
)

print("Telegram status:", response.status_code)

except Exception as e:
    print("Telegram Error:", e)

# =====
# PING INTERNET AKTIF
# =====

def check_ping(target=ACTIVE_CHECK_IP, count=5):
    try:
        result = subprocess.check_output(
            ["ping", "-c", str(count), target],
            stderr=subprocess.DEVNULL
        ).decode()

        latency = 0.0
        packet_loss = 100.0

        loss_match = re.search(r"(\d+(?:\.\d+)?)% packet loss",
result)

        if loss_match:
            packet_loss = float(loss_match.group(1))

    for line in result.splitlines():
        if "min/avg/max" in line or "rtt min/avg/max" in
line:
            values = line.split("=")[1].strip().split("/")
            if len(values) >= 2:
                latency = float(values[1])

    return latency, packet_loss

except Exception as e:
    print("Ping Error:", e)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        return 0.0, 100.0

# =====
# DETEKSI ROUTE MIKROTIK
# =====

def detect_current_network_from_mikrotik():
    try:
        returncode, output = mikrotik_ssh("/ip route print",
        timeout=15)

        print("Route detect output:")
        print(output)

        active_default_lines = []

        for line in output.splitlines():
            clean = line.strip()

            if "0.0.0.0/0" not in clean:
                continue

            if is_active_route_line(clean):
                active_default_lines.append(clean)

        for line in active_default_lines:
            if STARLINK_GATEWAY in line:
                return "STARLINK"

            if ORBIT_GATEWAY in line:
                return "ORBIT"

        return None

    except Exception as e:
        print("Detect current network error:", e)
        return None

def is_starlink_connected_route_active():
    try:
        returncode, output = mikrotik_ssh("/ip route print",
        timeout=15)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("Starlink connected route output:")
print(output)

for line in output.splitlines():
    clean = line.strip()

    if STARLINK_CONNECTED_NET not in clean:
        continue

    if "ether1" in clean and
is_active_route_line(clean):
        return True

    return False

except Exception as e:
    print("Starlink connected route check error:", e)
    return False

# =====
# CEK KHUSUS STARLINK
# =====

def check_starlink_via_mikrotik(count=5):
    try:
        if not is_starlink_connected_route_active():
            print("Ether1 / Starlink connected route tidak
aktif")

            return "ether1-down", 100.0

        returncode, output = mikrotik_ssh(
            f"/ping {STARLINK_CHECK_IP} count={count}",
            timeout=20
        )

        print("Starlink ping check output:")
        print(output)

        packet_loss = 100.0
        latency_text = "-"

        loss_match = re.search(r"packet-loss=(\d+(?:\.\d+)?)%",
output)

        if loss_match:
            packet_loss = float(loss_match.group(1))
```



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
avg_match = re.search(r"avg-rtt=([^\s]+)", output)
if avg_match:
    latency_text = avg_match.group(1)

return latency_text, packet_loss

except Exception as e:
    print("Starlink Check Error:", e)
    return "-", 100.0

# =====
# RAIN DETECTION
# =====

def detect_is_raining(level_esp, rain_value):
    level_esp = str(level_esp).upper().strip()

    if level_esp == "KERING":
        return 0

    if level_esp in ["GERIMIS", "HUJAN", "HUJAN_DERAS", "HUJAN
DERAS"]:
        return 1

    if rain_value <= DRY_RAIN_THRESHOLD:
        return 1

    return 0

# =====
# ML UNTUK STATUS RISIKO
# =====

def predict_with_ml(features):
    if model is None:
        if features["is_raining"] == 1:
            return "WARNING", 1.0

        if features["humidity"] >= 80:
            return "WARNING", 1.0
```



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        return "NORMAL", 1.0

    try:
        row = []

        for col in FEATURE_COLUMNS:
            row.append(features[col])

        prediction = model.predict([row])[0]
        confidence = 1.0

        if hasattr(model, "predict_proba"):
            probabilities = model.predict_proba([row])[0]
            confidence = float(max(probabilities))

        return str(prediction), confidence

    except Exception as e:
        print("ML Prediction Error:", e)
        return "WARNING", 0.0

def decide_status(features):
    packet_loss = features["packet_loss"]
    is_raining = features["is_raining"]

    if packet_loss >= PACKET_LOSS_CRITICAL:
        return "CRITICAL", 1.0

    if packet_loss >= PACKET_LOSS_WARNING:
        return "WARNING", 1.0

    if packet_loss < PACKET_LOSS_WARNING and is_raining == 0:
        return "NORMAL", 1.0

    status, confidence = predict_with_ml(features)

    if status == "CRITICAL":
        status = "WARNING"

    return status, confidence

# =====
# TARGET INTERNET
# =====

def get_target_network(status, starlink_packet_loss=None):
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
global starlink_normal_counter
global current_network_state

if status == "CRITICAL":
    starlink_normal_counter = 0
    return "ORBIT"

if status == "WARNING":
    print("WARNING hanya laporan, tetap jaringan
sekarang:", current_network_state)
    return current_network_state

if status == "NORMAL":

    if current_network_state == "STARLINK":
        starlink_normal_counter = 0
        return "STARLINK"

    if current_network_state == "ORBIT":

        if starlink_packet_loss is None:
            print("Starlink belum dicek, tetap ORBIT")
            return "ORBIT"

        if starlink_packet_loss <= STARLINK_OK_LOSS_MAX:
            starlink_normal_counter += 1

        print(
            "Starlink normal counter:",
            starlink_normal_counter,
            "/",
            FAILBACK_STARLINK_REQUIRED
        )

        if starlink_normal_counter >=
FAILBACK_STARLINK_REQUIRED:
            starlink_normal_counter = 0
            return "STARLINK"

        return "ORBIT"

    print("Starlink belum stabil / ether1 down, tetap
ORBIT")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
starlink_normal_counter = 0
return "ORBIT"

return current_network_state

def get_status_icon(status):
    if status == "NORMAL":
        return ICON_NORMAL

    if status == "WARNING":
        return ICON_WARNING

    if status == "CRITICAL":
        return ICON_CRITICAL

    return ICON_WARNING

# =====
# MIKROTIK SWITCH CLEAN
# =====

def run_mikrotik_script(script_name):
    if not ENABLE_MIKROTIK_SWITCH:
        print("MikroTik switch OFF / simulasi:", script_name)
        return True

    try:
        returncode, output = mikrotik_ssh(
            f"/system script run {script_name}",
            timeout=20
        )

        print("MikroTik script output:")
        print(output)

        return returncode == 0

    except Exception as e:
        print("MikroTik Error:", e)
        return False

def switch_network(target_network):
    global current_network_state
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
if target_network == current_network_state:
    print("Target sama dengan jaringan aktif, tidak perlu
switch ulang")
    return current_network_state

if target_network == "ORBIT":
    print("Target switch: ORBIT")
    script_success =
run_mikrotik_script(MIKROTIK_SCRIPT_TO_ORBIT)

    time.sleep(3)

    detected = detect_current_network_from_mikrotik()

    if detected == "ORBIT":
        current_network_state = "ORBIT"
        return "ORBIT"

    if script_success:
        current_network_state = "ORBIT"
        return "ORBIT"

    return current_network_state

print("Target switch: STARLINK")
script_success =
run_mikrotik_script(MIKROTIK_SCRIPT_TO_STARLINK)

    time.sleep(3)

    detected = detect_current_network_from_mikrotik()

    if detected == "STARLINK":
        current_network_state = "STARLINK"
        return "STARLINK"

    if script_success:
        current_network_state = "STARLINK"
        return "STARLINK"

    return current_network_state
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# =====
# KUOTA ORBIT
# =====

def send_orbit_quota_report():
    try:
        print("Mengirim laporan estimasi kuota Orbit...")

        result = subprocess.run(
            ["/usr/bin/python3", QUOTA_SCRIPT_PATH],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True,
            timeout=40
        )

        print("Quota stdout:")
        print(result.stdout)

        print("Quota stderr:")
        print(result.stderr)

    except Exception as e:
        print("Gagal kirim laporan kuota Orbit:", e)

# =====
# LOGGING
# =====

def save_log(
    source,
    features,
    rssi,
    status,
    confidence,
    target_network,
    active_network,
    starlink_packet_loss,
    starlink_latency,
    network_changed
):
    try:
        folder = os.path.dirname(LOG_PATH)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
if folder and not os.path.exists(folder):
    os.makedirs(folder)

file_exists = os.path.exists(LOG_PATH)

with open(LOG_PATH, "a", newline="") as file:
    writer = csv.writer(file)

    if not file_exists:
        writer.writerow([
            "timestamp",
            "source",
            "humidity",
            "rain_value",
            "is_raining",
            "rssi",
            "snr",
            "latency",
            "packet_loss",
            "starlink_latency",
            "starlink_packet_loss",
            "status",
            "confidence",
            "target_network",
            "active_network",
            "current_network_state",
            "network_changed"
        ])

    writer.writerow([
        now_log(),
        source,
        features["humidity"],
        features["rain_value"],
        features["is_raining"],
        rssi if rssi is not None else "",
        features["snr"],
        features["latency"],
        features["packet_loss"],
        starlink_latency,
        starlink_packet_loss if starlink_packet_loss is
not None else "",
        status,
        round(confidence, 4),
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        target_network,\n        active_network,\n        current_network_state,\n        network_changed\n    ])\n\nexcept Exception as e:\n    print("Log Error:", e)\n\n# =====\n# NOTIFICATION FILTER\n# =====\n\ndef should_send_notification(status, target_network,\n    active_network, level_esp, network_changed):\n    global last_notified_status\n    global last_notified_target\n    global last_notified_weather\n\n    if network_changed:\n        return True\n\n    if target_network != last_notified_target:\n        return True\n\n    if status == "CRITICAL" and last_notified_status !=\n        "CRITICAL":\n        return True\n\n    if last_notified_status == "CRITICAL" and status ==\n        "NORMAL":\n        return True\n\n    if level_esp and level_esp != last_notified_weather:\n        return True\n\n    return False\n\n# =====\n# MAIN PROCESS\n# =====\n\ndef process_monitoring(source="AUTO_NETWORK_CHECK"):
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
global last_notified_status
global last_notified_target
global last_notified_weather
global last_notified_network_state
global current_network_state

with process_lock:
    try:
        previous_network_state = current_network_state
        detected_network =
detect_current_network_from_mikrotik()

        network_changed = False

        if detected_network in ["STARLINK", "ORBIT"]:
            if detected_network != current_network_state:
                network_changed = True
                print(

                    "Network changed by MikroTik:",
                    current_network_state,
                    "->",
                    detected_network
                )

                current_network_state = detected_network

        raw_rain_value = last_sensor_data.get("rain_value",
None)
        raw_humidity = last_sensor_data.get("humidity",
None)
        raw_level_esp = last_sensor_data.get("level_esp",
None)
        rssi = last_sensor_data.get("rssi", None)

        sensor_available = (
            raw_rain_value is not None
            and raw_humidity is not None
            and raw_level_esp is not None
        )

        if sensor_available:
            rain_value = safe_int(raw_rain_value, 17000)
            humidity = safe_int(raw_humidity, 29)
            level_esp = str(raw_level_esp).upper().strip()
        else:
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
rain_value = 17000
humidity = 29
level_esp = "KERING"

if rssi is not None:
    rssi = safe_int(rssi, None)

is_raining = detect_is_raining(level_esp,
rain_value)

latency, packet_loss = check_ping(ACTIVE_CHECK_IP)

features = {
    "humidity": humidity,
    "rain_value": rain_value,
    "is_raining": is_raining,
    "snr": 0,
    "latency": latency,
    "packet_loss": packet_loss,
}

status, confidence = decide_status(features)

starlink_latency = "-"
starlink_packet_loss = None

if current_network_state == "ORBIT":
    starlink_latency, starlink_packet_loss =
check_starlink_via_mikrotik()

target_network = get_target_network(status,
starlink_packet_loss)

print("=====")
print("Source:", source)
print("Detected Network:", detected_network)
print("Previous Network:", previous_network_state)
print("Current Network:", current_network_state)
print("Network Changed:", network_changed)
print("Sensor Available:", sensor_available)
print("Level ESP:", level_esp)
print("Rain Value:", rain_value)
print("Humidity:", humidity)
print("Is Raining:", is_raining)
print("RSSI:", rssi, "dBm")
print("Active Ping:", latency, "ms")
print("Active Packet Loss:", packet_loss, "%")
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
print("Starlink Latency:", starlink_latency)
print("Starlink Packet Loss:",
starlink_packet_loss)
print("Status:", status)
print("Confidence:", confidence)
print("Target Network:", target_network)
print("=====")

before_switch_state = previous_network_state

active_network = switch_network(target_network)

# Kirim estimasi kuota Orbit saat baru pindah ke
ORBIT,

# baik karena script Raspberry maupun karena
MikroTik otomatis failover.
if before_switch_state != "ORBIT" and
current_network_state == "ORBIT":
    send_orbit_quota_report()

time.sleep(SWITCH_DELAY_SECONDS)

post_latency, post_packet_loss =
check_ping(ACTIVE_CHECK_IP)

save_log(
    source,
    features,
    rssi,
    status,
    confidence,
    target_network,
    active_network,
    starlink_packet_loss,
    starlink_latency,
    network_changed
)

if not should_send_notification(
    status,
    target_network,
    active_network,
    level_esp,
    network_changed
):
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        print("Tidak kirim Telegram, karena tidak ada
        perubahan penting")
        return

    icon = get_status_icon(status)
    rain_text = "YA" if is_raining == 1 else "TIDAK"

    if rssi is None or rssi == 0:
        rssi_text = "Belum tersedia"
    else:
        rssi_text = f"{rssi} dBm"

    sensor_source_text = "Data terakhir ESP32" if
    sensor_available else "Default sistem"

    if current_network_state == "STARLINK" and
    starlink_packet_loss is None:
        starlink_text = "Aktif sebagai jalur utama"
    elif starlink_packet_loss is None:
        starlink_text = "Tidak dicek"
    else:
        starlink_text = f"{starlink_packet_loss}% loss,
        RTT {starlink_latency}"

    message = (

        f"{icon} ML Rain Failover Report\n\n"
        f"Status ML : {status}\n"
        f"Confidence : {confidence:.2%}\n"
        f"Target Internet : {target_network}\n"
        f"Active Internet : {active_network}\n"
        f>Data Sensor : {sensor_source_text}\n\n"
        f"{ICON_HUMIDITY} Kelembapan : {humidity}%\n"
        f"{ICON_RAIN} Hujan : {rain_text}\n"
        f"{ICON_SENSOR} Rain Sensor : {rain_value}\n"
        f"{ICON_SENSOR} Level ESP : {level_esp if
        level_esp else '-'}\n"
        f"{ICON_RSSI} RSSI WiFi ESP32 : {rssi_text}\n"
        f"{ICON_PING} Ping Aktif : {latency} ms\n"
        f"{ICON_PACKET} Packet Loss Aktif :
        {packet_loss}%\n"
        f"{ICON_PING} Cek Khusus Starlink :
        {starlink_text}\n"
        f"{ICON_PING} Ping Setelah Check :
        {post_latency} ms\n"

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
f"{ICON_PACKET} Packet Loss Setelah Check :
{post_packet_loss}%\n"
    f"{ICON_TIME} Time : {now_text()}"
)

send_telegram(message)

last_notified_status = status
last_notified_target = target_network
last_notified_weather = level_esp
last_notified_network_state = current_network_state

except Exception as e:
    print("Process Monitoring Error:", e)

# =====
# AUTO NETWORK LOOP
# =====

def network_monitor_loop():
    time.sleep(10)

    while True:
        process_monitoring("AUTO_NETWORK_CHECK")
        time.sleep(NETWORK_CHECK_INTERVAL)

# =====
# MQTT
# =====

def on_connect(client, userdata, flags, rc):
    print("Connected MQTT:", rc)

    if rc == 0:
        client.subscribe(MQTT_TOPIC)
        print("Subscribed:", MQTT_TOPIC)
    else:
        print("MQTT gagal connect")

def on_message(client, userdata, msg):
    global last_sensor_data

    try:
```



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
raw = msg.payload.decode()
print("MQTT raw:", raw)

data = json.loads(raw)

rain_value = safe_int(data.get("rain_value", None),
None)

humidity = safe_int(data.get("humidity", None), None)
level_esp = str(data.get("level", "")).upper().strip()

if "rssi" in data:
    rssi = safe_int(data.get("rssi"), None)
else:
    rssi = last_sensor_data.get("rssi", None)

if rain_value is None or humidity is None or not
level_esp:
    print("MQTT data sensor tidak lengkap, skip update
sensor")
    return

last_sensor_data = {
    "rain_value": rain_value,
    "humidity": humidity,
    "level_esp": level_esp,
    "rssi": rssi
}

save_last_sensor_data()

process_monitoring("MQTT_SENSOR")

except Exception as e:
    print("MQTT Error:", e)

# =====
# START
# =====

load_last_sensor_data()

detected_network = detect_current_network_from_mikrotik()

if detected_network in ["STARLINK", "ORBIT"]:
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
current_network_state = detected_network

last_notified_target = current_network_state
last_notified_network_state = current_network_state
last_notified_weather = str(
    last_sensor_data.get("level_esp", "")
).upper().strip()

print("MQTT broker:", MQTT_BROKER, MQTT_PORT)
print("MQTT topic:", MQTT_TOPIC)
print("ENABLE_MIKROTIK_SWITCH:", ENABLE_MIKROTIK_SWITCH)
print("Current Network State:", current_network_state)
print("NETWORK_CHECK_INTERVAL:", NETWORK_CHECK_INTERVAL)
print("PACKET_LOSS_WARNING:", PACKET_LOSS_WARNING)
print("PACKET_LOSS_CRITICAL:", PACKET_LOSS_CRITICAL)
print("ACTIVE_CHECK_IP:", ACTIVE_CHECK_IP)
print("STARLINK_CHECK_IP:", STARLINK_CHECK_IP)
print("STARLINK_OK_LOSS_MAX:", STARLINK_OK_LOSS_MAX)
print("FAILBACK_STARLINK_REQUIRED:",
      FAILBACK_STARLINK_REQUIRED)
print("QUOTA_SCRIPT_PATH:", QUOTA_SCRIPT_PATH)

client = mqtt.Client(
    client_id=MQTT_CLIENT_ID,
    clean_session=True
)

client.on_connect = on_connect
client.on_message = on_message

client.connect(
    MQTT_BROKER,
    MQTT_PORT,
    60
)

monitor_thread = threading.Thread(
    target=network_monitor_loop,
    daemon=True
)

monitor_thread.start()

client.loop_forever()
```