DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST*

SKRIPSI

DISUSUN OLEH

TASYA SEPTIA SIREGAR
NPM 2209020140

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026

DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST*

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Tenologi Infromasi. pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

TASYA SEPTIA SIREGAR
NPM 2209020140

PROGRAM STUDI TEKNOLOGI INFROMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026

LEMBAR PENGESAHAN

Judul Skripsi	:	**DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST***
Nama Mahasiswa	:	**TASYA SEPTIA SIREGAR**
NPM	:	**2209020140**
Program Studi	:	**TEKNOLOGI INFORMASI**

Menyetujui

Komisi Pembimbing

(Mulkan Azhari, S.Kom M.Kom)
NIDN. 0108129402

Ketua Program Studi	**Dekan**
(Fatma Sari Hutagalung, S.Kom, M.Kom)	**(Dr. Al-Khowarizmi, S.Kom., M.Kom.)**
NIDN. 0117019301	**NIDN. 0127099201**

PERNYATAAN ORISINALITAS

DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST*

SKRIPSI

Menyatakan dengan sebenarnya bahwa proposal skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri. Apabila di kemudian hari terbukti atau dapat dibuktikan proposal skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Medan, April 2026

Yang membuat pernyataan

TASYA SEPTIA SIREGAR

NPM **2209020140**

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: **TASYA SEPTIA SIREGAR**
NPM	: **2209020140**
Program Studi	: **TEKNOLOGI INFORMASI**
Karya Ilmiah	: **Skripsi**

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, April 2026

Yang membuat pernyataan

TASYA SEPTIA SIREGAR

NPM **2209020140**

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: **TASYA SEPTIA SIREGAR**
Tempat dan Tanggal Lahir	: Medan, 5 September 2004
Alamat Rumah	: Lor Sentosa 4-A LK 17
Telepon/Faks/HP	: 081263465857
E-mail	: tasyaseptiasiregar@gmail.com
Instansi Tempat Kerja	:
Alamat Kantor	:

DATA PENDIDIKAN

SD	: SD Negeri 060966	TAMAT: 2016
SMP	: SMP Negeri 5 Medan	TAMAT: 2019
SMA	: SMA Negeri 19 Medan	TAMAT: 2022

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan proposal skripsi yang berjudul "Deteksi Penyebaran Bot di Media Sosial Menggunakan Algoritma Local Outlier Factor (LOF) dan CatBoost" ini dengan baik. Proposal ini disusun sebagai salah satu persyaratan untuk melaksanakan penelitian skripsi pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.

Penulis menyadari bahwa penyusunan proposal ini tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Agussani, M.Ap., selaku Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU. Tanpa dukungan, doa, bimbingan, serta bantuan yang diberikan, skripsi ini tidak akan dapat terselesaikan dengan baik. Semoga segala kebaikan yang telah diberikan mendapatkan balasan yang berlipat ganda.

3. Ibu Dr. Firahmi Rizky, S.Kom., M.Kom., selaku Wakil Dekan I, atas bimbingan, arahan, dan dukungan yang telah diberikan sehingga skripsi ini dapat terselesaikan dengan baik.
4. Bapak Mhd Basri, S.Si., M.Kom., selaku Wakil Dekan III, atas bimbingan, arahan, dan dukungan yang telah diberikan selama proses penyusunan skripsi ini
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom., selaku Ketua Program Studi Teknologi Informasi FIKTI UMSU. Setiap dukungan dan kebaikan yang diberikan sangat berarti dan menjadi kekuatan bagi saya untuk menyelesaikan skripsi ini.
6. Bapak Okvi Nugroho, S.Kom., M.Kom., selaku Sekretaris Program Studi Teknologi Informasi FIKTI UMSU, atas segala arahan dan dukungan yang telah diberikan selama proses penyusunan skripsi ini.
7. Bapak Mulkan Azhari, S.Kom., M.Kom., selaku dosen pembimbing yang telah dengan sabar memberikan bimbingan, arahan, serta motivasi selama proses penyusunan skripsi ini hingga selesai.
8. Kedua orang tua tercinta, terima kasih atas doa, kasih sayang, dukungan, dan pengorbanan yang tiada henti. Berkat restu dan kesabaran Ayah dan Ibu, penulis dapat menyelesaikan skripsi ini. Pencapaian ini penulis persembahkan sebagai bentuk terima kasih dan bakti sebagai seorang sarjana.

Penulis menyadari bahwa proposal ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang. Akhir kata, semoga proposal ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi, khususnya di bidang keamanan siber dan deteksi anomali.

Medan, April 2026

Penulis,

TASYA SEPTIA SIREGAR

NPM 2209020140

DETEKSI PENYEBARAN *BOT* DI MEDIA SOSIAL MENGGUNAKAN ALGORITMA *LOCAL OUTLIER FACTOR (LOF)* DAN *CATBOOST*

ABSTRAK

Media sosial telah menjadi infrastruktur kritis dalam diskursus publik, namun juga rentan terhadap manipulasi oleh akun bot otomatis. Bot modern semakin canggih, mampu meniru perilaku manusia dan beroperasi secara terkoordinasi, sehingga menyulitkan metode deteksi konvensional. Penelitian ini mengusulkan pendekatan hibrida untuk deteksi akun bot di media sosial dengan menggabungkan algoritma *Local Outlier Factor* (LOF) sebagai metode deteksi anomali berbasis kepadatan lokal dan CatBoost sebagai algoritma klasifikasi *gradient boosting* yang unggul dalam menangani fitur kategorikal. Pendekatan ini diimplementasikan dan diuji menggunakan dataset benchmark TwiBot-22 yang memiliki struktur graf heterogen. Hasil eksperimen menunjukkan bahwa model hibrida dengan parameter optimal k=50 pada LOF mencapai akurasi 74,48%, *recall* 89,81-90,53% untuk kelas bot, dan nilai AUC sebesar 0,826. Analisis *feature importance* mengungkapkan bahwa fitur berbasis struktur relasi sosial seperti followers_count dan listed_count lebih dominan dalam menentukan klasifikasi dibandingkan atribut profil kosmetik. Meskipun kontribusi skor LOF relatif kecil (3,11%), integrasinya terbukti konsisten meningkatkan metrik *recall* dan *Matthews Correlation Coefficient* (MCC). Sistem yang dikembangkan dalam bentuk aplikasi web ini mampu memberikan prediksi cepat (<1 detik) dan menunjukkan potensi sebagai solusi deteksi bot yang efisien dan adaptif.

Kata Kunci: Deteksi Bot, Media Sosial, Local Outlier Factor (LOF), CatBoost, TwiBot-22, Pembelajaran Mesin Hibrida.

SOCIAL MEDIA BOT DETECTION USING LOCAL OUTLIER FACTOR (LOF) AND CATBOOST ALGORITHMS

ABSTRACT

Social media has become a critical infrastructure for public discourse, yet it is also vulnerable to manipulation by automated bot accounts. Modern bots are increasingly sophisticated, capable of mimicking human behavior and operating in a coordinated manner, thus complicating conventional detection methods. This research proposes a hybrid approach for detecting bot accounts on social media by combining the Local Outlier Factor (LOF) algorithm as a density-based anomaly detection method and CatBoost as a gradient boosting classification algorithm that excels in handling categorical features. This approach is implemented and tested using the benchmark TwiBot-22 dataset, which features a heterogeneous graph structure. Experimental results show that the hybrid model with an optimal parameter of k=50 for LOF achieves an accuracy of 74.48%, a recall of 89.81-90.53% for the bot class, and an AUC value of 0.826. Feature importance analysis reveals that features based on social relation structures, such as followers_count and listed_count, are more dominant in determining classification compared to cosmetic profile attributes. Although the contribution of the LOF score is relatively small (3.11%), its integration consistently improves recall and Matthews Correlation Coefficient (MCC) metrics. The system, developed as a web application, is capable of providing fast predictions (<1 second) and demonstrates potential as an efficient and adaptive bot detection solution.

Keywords: *Bot Detection, Social Media, Local Outlier Factor (LOF), CatBoost, TwiBot-22, Hybrid Machine Learning.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Transformasi ekosistem informasi digital dalam satu dekade terakhir telah menempatkan media sosial bukan lagi sekadar sebagai sarana interaksi sosial, melainkan sebagai infrastruktur kritis bagi diskursus publik, partisipasi politik, dan penyebaran berita global. *Platform* seperti X (sebelumnya Twitter), Facebook, dan TikTok telah menjadi arena utama di mana opini publik dibentuk dan digerakkan. Namun, seiring dengan meningkatnya signifikansi strategis media sosial, kerentanan terhadap manipulasi terkoordinasi juga mengalami eskalasi yang mengkhawatirkan. Salah satu instrumen utama dalam manipulasi ekosistem digital ini adalah *social bots* algoritma otonom yang dirancang untuk meniru perilaku manusia guna mengamplifikasi narasi tertentu, menyebarkan disinformasi, memanipulasi pasar keuangan, atau melakukan penipuan. Keberadaan *bot* ini menciptakan distorsi dalam *organic engagement*, yang pada gilirannya mengancam integritas demokrasi dan kepercayaan publik terhadap arus informasi (Veit & Lones, 2025).

Evolusi social *bot*s telah mencapai tingkat kecanggihan yang melampaui kemampuan deteksi metode konvensional. Generasi awal *bot*, yang sering disebut sebagai simple *bot*s, relatif mudah diidentifikasi melalui fitur-fitur statis yang mencolok, seperti ketiadaan foto profil, pola *tweet* yang *repetitif*, atau rasio *following* terhadap *followers* yang ekstrem. Metode deteksi berbasis aturan (*rulebased*) atau analisis fitur metadata sederhana sudah cukup memadai untuk

menangani ancaman ini. Akan tetapi, lanskap ancaman kini didominasi oleh *advanced bots* atau *cyborgs* yang mampu memanipulasi perilaku mereka untuk berbaur dengan pengguna manusia. *Bot* modern ini memanfaatkan kemajuan dalam *Generative AI* dan *Large Language Models* (LLM) untuk menghasilkan konten teks yang bervariasi secara semantik, berinteraksi dalam diskusi yang kompleks, dan mengatur pola aktivitas temporal yang tampak alami. Lebih jauh lagi, ancaman ini diperparah oleh fenomena *Coordinated Inauthentic Behavior* (CIB), di mana ribuan akun *bot* beroperasi dalam sinkronisasi yang ketat namun tersembunyi untuk memviralkan topik tertentu tanpa terdeteksi sebagai anomali individu (Cheng et al., 2025).

Kompleksitas masalah ini menuntut pendekatan deteksi yang tidak hanya mengandalkan fitur permukaan, tetapi juga mampu menangkap anomali struktural dan perilaku yang halus. Salah satu tantangan mendasar dalam penelitian deteksi *bot* selama ini adalah keterbatasan dataset benchmark yang tersedia. Banyak studi sebelumnya mengandalkan dataset seperti Cresci-2015 atau TwiBot-22 yang, meskipun perintis, memiliki keterbatasan signifikan dalam hal skala, keragaman entitas, dan kelengkapan struktur *graf*. Dataset yang lebih tua sering kali tidak mencerminkan topologi jaringan sosial yang heterogen, di mana interaksi tidak hanya terjadi antar-pengguna, tetapi juga melibatkan entitas lain seperti hashtag dan lists. Keterbatasan ini menyebabkan model yang dilatih pada dataset lama mengalami degradasi performa yang parah ketika dihadapkan pada strategi *bot* terbaru, sebuah fenomena yang dikenal sebagai *concept drift* atau kedaluwarsa model.(Veit & Lones, 2025)

Untuk mengatasi stagnasi ini, Feng et al. (2022) memperkenalkan Twi*Bot*22, sebuah benchmark deteksi *bot* berbasis *graf* yang paling komprehensif hingga saat ini. Twi*Bot*-22 menawarkan skala data lima kali lebih besar dari pendahulunya dan struktur *graf* yang *heterogen*, mencakup beragam entitas dan relasi yang merefleksikan kompleksitas ekosistem Twitter yang sesungguhnya. Keberadaan dataset ini membuka peluang baru untuk mengeksplorasi metode deteksi yang lebih canggih. Namun, tantangan komputasi dalam memproses struktur *graf* berskala besar sering kali menjadi hambatan bagi penerapan metode berbasis *Graph Neural Networks* (GNN) secara *real-time*. Oleh karena itu, terdapat kebutuhan mendesak untuk mengembangkan kerangka kerja deteksi yang efisien namun tetap mampu memanfaatkan kekayaan informasi dalam Twi*Bot*-22 tanpa membebani sumber daya komputasi secara berlebihan. (Afansyah & Nawi, 2025)

Penelitian ini mengusulkan pendekatan hibrida yang menggabungkan kekuatan *Local Outlier Factor* (LOF) dan *CatBoost* untuk menjawab tantangan tersebut. Pemilihan algoritma ini didasarkan pada dua hipotesis utama. Pertama, meskipun *bot* canggih berusaha meniru manusia, mereka sering kali meninggalkan jejak anomali pada tingkat densitas local misalnya, sekelompok *bot* mungkin memiliki pola interaksi yang sangat mirip satu sama lain dibandingkan dengan lingkungan pengguna manusia di sekitarnya. *Local Outlier Factor* (LOF), sebagai algoritma deteksi anomali berbasis densitas (*density-based*), memiliki keunggulan unik dalam mendeteksi *local outliers* yang mungkin terlewatkan oleh metode deteksi global. Kedua, data media sosial kaya akan fitur kategorikal (seperti lokasi, bahasa, sumber perangkat, dan entitas hashtag) yang sering kali

sulit ditangani oleh algoritma *machine learning* tradisional tanpa pra-pemrosesan yang rumit seperti *One-Hot Encoding* yang memboroskan memori. *CatBoost* (*Categorical Boosting*) dipilih karena kemampuan superiornya dalam menangani fitur kategorikal secara natif melalui *ordered target statistics*, serta ketahanannya terhadap *overfitting* pada dataset yang heterogen.(Paulauskas & Bagdonas, 2022)

Sinergi antara *Local Outlier Factor* (LOF) sebagai ekstraktor fitur anomali *unsupervised* dan *CatBoost* sebagai pengklasifikasi *supervised* yang kuat diharapkan dapat menghasilkan model deteksi yang tidak hanya akurat dalam membedakan *bot* dan manusia, tetapi juga tangguh terhadap strategi penyamaran *bot*. Pendekatan ini menawarkan alternatif yang lebih efisien secara komputasi dibandingkan arsitektur *deep learning* berbasis *graf* yang berat, sambil tetap memanfaatkan kekayaan fitur yang disediakan oleh dataset Twi*Bot*-22. Melalui penelitian ini, diharapkan dapat dihasilkan kontribusi metodologis yang signifikan dalam upaya pembersihan ekosistem digital dari manipulasi otomatis.

1.2 Rumusan Masalah

Berdasarkan latar belakang, tujuan penelitian, serta hasil eksperimen yang telah dilakukan, maka rumusan masalah dalam penelitian "Deteksi Penyebaran Bot di Media Sosial Menggunakan Algoritma Local Outlier Factor (LOF) dan CatBoost" dapat dirumuskan sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sistem deteksi akun bot pada media sosial menggunakan pendekatan machine learning berbasis algoritma Local Outlier Factor (LOF) dan CatBoost?

2. Bagaimana pengaruh integrasi metode deteksi anomali Local Outlier Factor (LOF) terhadap performa model klasifikasi CatBoost dalam mendeteksi akun bot di media sosial?
3. Bagaimana pengaruh variasi parameter k pada algoritma Local Outlier Factor terhadap kinerja model deteksi bot yang diukur menggunakan metrik evaluasi seperti Accuracy, Precision, Recall, F1-Score, Matthews Correlation Coefficient (MCC), dan AUC?
4. Seberapa baik performa model hybrid LOF–CatBoost dalam membedakan akun bot dan akun manusia berdasarkan fitur metadata dan fitur turunan dari dataset TwiBot?

1.3 Batasan Masalah

Mengingat luasnya cakupan permasalahan dalam deteksi *bot* dan besarnya skala dataset yang digunakan, penelitian ini membatasi ruang lingkup pembahasan pada aspek-aspek berikut untuk menjaga fokus dan kedalaman analisis:

1) Sumber Data: Penelitian ini secara eksklusif menggunakan dataset Twi*Bot*22 yang dipublikasikan oleh Feng et al. (2022). Dataset ini dipilih karena statusnya sebagai benchmark *state of the-art* yang mencakup keragaman entitas dan relasi. Penggunaan dataset lain seperti Cresci-15 atau Twi*Bot*20 hanya dilakukan sebagai referensi komparatif dalam tinjauan pustaka, bukan sebagai objek eksperimen utama.
2) Metode Algoritmik: Fokus utama algoritma dibatasi pada *Local Outlier Factor* (LOF) untuk deteksi anomali secara unsupervised dan *CatBoost* untuk klasifikasi secara *supervised*. Algoritma lain seperti *Isolation Forest*,

XGBoost, LightGBM, atau *Random Forest* akan digunakan semata-mata sebagai baseline untuk memvalidasi kinerja model yang diusulkan.

3) Lingkup Deteksi: Penelitian ini berfokus pada deteksi level pengguna (*userlevel detection*), yaitu mengklasifikasikan apakah suatu akun adalah *bot* atau manusia. Deteksi pada *level tweet* individual atau deteksi klaster *botnet* secara keseluruhan (deteksi komunitas) berada di luar lingkup utama, meskipun fitur-fitur yang merepresentasikan aktivitas komunitas mungkin digunakan.
4) Fitur *Graf*: Meskipun Twi*Bot*-22 adalah dataset berbasis *graf*, penelitian ini tidak akan membangun model *Graph Neural Network* (GNN) end-to-end (seperti R-GCN atau GAT). Sebaliknya, informasi *graf* akan diekstraksi menjadi fitur tabular (misalnya, *degree centrality, clustering coefficient*, jumlah tetangga) untuk diproses oleh *Local Outlier Factor* (LOF) dan *CatBoost*. Pendekatan ini dipilih untuk mengevaluasi efektivitas metode non-GNN yang lebih efisien secara komputasi.
5) Infrastruktur Komputasi: Eksperimen akan dilakukan pada lingkungan komputasi tunggal (non-terdistribusi) menggunakan pustaka *Python* standar (*scikit-learn*, *CatBoost*). Jika ukuran dataset penuh Twi*Bot*-22 melampaui kapasitas memori, akan diterapkan teknik *stratified sampling* untuk mengambil *subset* representatif (misalnya 10-20% dari total data) tanpa mengubah distribusi kelas asli.

1.4 Tujuan Penelitian

Penelitian ini dilaksanakan dengan tujuan-tujuan spesifik sebagai berikut:

1. Membangun dan mengimplementasikan kerangka kerja deteksi *bot* hibrida yang menggabungkan sensitivitas deteksi anomali lokal dari *Local Outlier Factor* (LOF) dengan akurasi klasifikasi dari *CatBoost* pada dataset Twi*Bot*-22
2. Menganalisis parameter optimal untuk algoritma *Local Outlier Factor* (LOF) (seperti jumlah tetangga k) dalam konteks data media sosial untuk memaksimalkan separabilitas antara *inliers* (manusia) dan *outliers* (*bot*)
3. Mengevaluasi secara empiris keunggulan *CatBoost* dalam menangani fitur kategorikal pada data Twitter dibandingkan dengan metode *gradient boosting* konvensional, serta mengukur efisiensi waktu pelatihan dan inferensinya
4. Membuktikan hipotesis bahwa penambahan fitur skor *Local Outlier Factor* (LOF) dapat memberikan informasi kontekstual tambahan yang signifikan bagi *classifier*, sehingga meningkatkan kemampuan model dalam mendeteksi *bot* yang memiliki karakteristik samar atau ambigu.
5. Menyediakan analisis mendalam mengenai karakteristik fitur (profil perilaku) *bot* modern yang terdapat dalam Twi*Bot*-22, guna memberikan wawasan baru bagi komunitas riset keamanan siber mengenai evolusi strategi *bot*

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dan manfaat dalam berbagai aspek:

1. Manfaat Teoritis:
   a) Memperkaya literatur deteksi social *bot* dengan memberikan bukti empiris mengenai efektivitas pendekatan hibrida *unsupervisedsupervised* pada dataset *benchmark* terbaru

b) Memberikan pemahaman baru tentang bagaimana algoritma berbasis densitas *Local Outlier Factor* (LOF) dapat melengkapi algoritma berbasis pohon (*CatBoost*) dalam menangani masalah klasifikasi kelas tidak seimbang dan anomali tersamar

c) Menawarkan perspektif alternatif terhadap tren penggunaan *Deep Learning* yang berat, dengan menunjukkan bahwa metode *ensemble learning* yang dioptimalkan dapat mencapai kinerja kompetitif dengan biaya komputasi yang lebih rendah

2. Manfaat Praktis:

a) Menghasilkan model deteksi yang siap diadaptasi oleh pengelola platform media sosial untuk memitigasi penyebaran disinformasi dan aktivitas inautentik secara lebih efektif.

b) Memberikan panduan bagi praktisi keamanan siber dalam memilih fitur-fitur yang paling relevan untuk memantau aktivitas *bot*, berdasarkan analisis *feature importance* yang dihasilkan.

c) Menyediakan kerangka kerja (*framework*) kode yang dapat digunakan kembali untuk eksperimen lanjutan pada dataset Twi*Bot*22 atau dataset serupa lainnya.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi Terdahulu

Penelitian mengenai deteksi *social bot* telah berkembang pesat seiring dengan evolusi kompleksitas *bot* itu sendiri. Tinjauan literatur ini mengklasifikasikan pendekatan yang ada ke dalam tiga paradigma utama: metode berbasis fitur tradisional, metode berbasis *graf*, dan pendekatan hibrida yang relevan dengan penelitian ini.

2.1.1 Metode Berbasis Fitur dan Pembelajaran Mesin Tradisional

Pada tahap awal perkembangannya, deteksi *bot* sangat bergantung pada *feature engineering manual* yang mengekstraksi atribut dari metadata pengguna dan riwayat konten. Varol et al. (2022) dan Yang et al. (2022) merintis penggunaan algoritma seperti *Random Forest* (RF) dan *Support Vector Machine* (SVM) dengan fitur-fitur statistik sederhana seperti frekuensi *tweet*, rasio re*tweet*, dan usia akun. Pendekatan ini, yang sering disebut sebagai "*Botometer-style*", sangat efektif untuk mendeteksi *bot* sederhana (*simple bots*) yang memiliki perilaku deterministik. Namun, studi oleh Cresci et al. (2022) dan penelitian selanjutnya menunjukkan bahwa metode ini rentan terhadap "evolusi *bot*", di mana *bot* modern mampu memalsukan metadata dan meniru pola temporal manusia, menyebabkan penurunan akurasi yang signifikan(Feng et al., 2022). Khoshkar (2025) baru-baru ini mengevaluasi berbagai model *Machine Learning* (ML) dan hibrida pada dataset publik, menyoroti bahwa meskipun model tradisional cepat, mereka sering gagal menangkap nuansa perilaku *bot* yang

terkoordinasi. Temuan ini menegaskan perlunya model yang lebih canggih yang mampu menangani interaksi fitur yang kompleks. Selain itu, Heidari et al. (2021) telah mengeksplorasi penggunaan *Local Outlier Factor* (LOF) dalam konteks keamanan jaringan umum, namun aplikasinya secara spesifik sebagai komponen fitur untuk deteksi *bot* media sosial pada *dataset* skala besar seperti Twi*Bot*-22 masih jarang dieksplorasi secara mendalam.

Pendekatan berbasis fitur dan pembelajaran mesin tradisional telah banyak digunakan dalam deteksi akun *bot* di media sosial. Metode-metode ini umumnya mengekstraksi fitur perilaku pengguna, fitur jaringan sosial, dan fitur konten untuk membedakan antara akun manusia dan *bot*. Dalam tinjauan literatur terkini, Aljabri et al. (2023) mengulas berbagai teknik pembelajaran mesin yang diterapkan pada deteksi social *bot*, mencakup metode supervised, semi-supervised, maupun unsupervised, serta mengidentifikasi fitur-fitur utama yang menjadi prediktor efektif untuk klasifikasi *bot* di platform seperti Twitter, Facebook, dan Instagram. Selain itu, penelitian oleh Liu et al. (2025) menggambarkan evolusi penelitian deteksi *bot* dari pendekatan profiling individu hingga pendekatan analisis kelompok, dan menunjukkan semakin kompleksnya teknik untuk menghadapi *bot* yang semakin canggih dalam konteks media sosial modern.

2.1.2 Metode Berbasis *Graf*

Menyadari bahwa *bot* sering beroperasi dalam kelompok, fokus penelitian bergeser ke metode berbasis *graf* yang mengeksploitasi struktur jaringan. Feng et al. (2021) dan timnya memelopori penggunaan *Graph Neural Networks* (GNN) seperti *Graph Convolutional Networks* (GCN) dan *Relational Graph*

Transformers (RGT) untuk deteksi *bot*. Dengan diperkenalkannya TwiBot-22 dan Twi*Bot*-22, metode seperti *Bot*RGCN dan RGT telah menjadi standar baru (state-of-the-art), secara konsisten mengungguli metode berbasis fitur dengan memanfaatkan prinsip homofili bahwa *bot* cenderung berinteraksi dengan *bot* lain. (Afansyah & Nawi, 2025)

Meskipun demikian, metode berbasis GNN memiliki kelemahan inheren dalam hal skalabilitas dan kebutuhan sumber daya. Pelatihan GNN pada *graf* dengan jutaan node memerlukan memori GPU yang sangat besar dan waktu komputasi yang lama. Selain itu, banyak metode GNN bersifat transduktif, artinya mereka memerlukan *graf* utuh selama pelatihan dan sulit digeneralisasi ke node baru yang belum terlihat tanpa pelatihan ulang. Penelitian terbaru oleh tim pengembang RoG *Bot* (2025) mencoba mengatasi ketergantungan pada relasi eksplisit ini dengan mengusulkan kerangka kerja multimodal yang tetap menggunakan penalaran berbasis *graf* namun tanpa memerlukan data follower-following yang lengkap, mencapai akurasi tinggi pada dataset Cresci. Tren ini mendukung premis penelitian ini: bahwa pendekatan fitur yang cerdas (termasuk fitur yang diturunkan dari *graf*) dapat menyaingi kompleksitas GNN murni.(Anshul et al., 2025)

2.1.3 Pendekatan Hibrida dan Ensemble

Pendekatan hibrida yang menggabungkan berbagai teknik ML semakin diminati untuk menutupi kelemahan masing-masing model tunggal. Penelitian oleh *Dung Beetle Optimizer* (DBO) yang dioptimalkan untuk parameter *CatBoost* pada deteksi *botnet* IoT menunjukkan bahwa *CatBoost* memiliki kinerja superior dibandingkan algoritma gradient boosting lainnya dalam menangani data trafik

yang heterogen. Keunggulan *CatBoost* dalam menangani fitur kategorikal secara natif tanpa preprocessing yang berat menjadikannya kandidat ideal untuk data media sosial yang kaya akan atribut nominal (Cheng et al., 2025)

Di sisi lain, integrasi metode *unsupervised* seperti *Local Outlier Factor* (LOF) ke dalam kerangka supervised menawarkan potensi untuk meningkatkan deteksi terhadap *bot* yang belum pernah dilihat sebelumnya (*zero-day bots*). Sementara *Local Outlier Factor* (LOF) secara tradisional digunakan sebagai detektor anomali mandiri, penggunaannya sebagai *feature extractor* untuk memperkaya input model klasifikasi supervised merupakan area yang menjanjikan yang belum banyak dieksploitasi secara luas dalam literatur Twi*Bot*-22. Penelitian ini akan mengisi celah tersebut dengan mengintegrasikan *Local Outlier Factor* (LOF) dan *CatBoost* secara sistematis. (Elhadad et al., 2023)

2.2 Landasan Teori

2.2.1 Social *Bot*s: Taksonomi dan Evolusi

Social *bot*s dapat didefinisikan sebagai akun media sosial yang dikendalikan oleh algoritma komputer untuk menghasilkan konten dan melakukan interaksi secara otomatis. Ferrara et al. (2022) dan studi selanjutnya mengklasifikasikan *bot* ke dalam beberapa kategori berdasarkan perilaku dan tujuan strategisnya:

1. *Spam Bots*: Jenis *bot* paling primitif yang bertugas menyebarkan tautan iklan, malware, atau konten promosi secara agresif. Mereka mudah dideteksi karena volume postingan yang tidak wajar dan konten yang repetitif.

2. *Fake Followers*: Akun-akun pasif yang dibuat dalam jumlah ribuan untuk dijual guna meningkatkan jumlah pengikut (*follower count*) akun lain. Mereka jarang berinteraksi dan sering kali tidak memiliki profil yang lengkap.
3. *Political*/*Influence Bots*: *Bot* canggih yang dirancang untuk memanipulasi opini publik. Mereka berpartisipasi dalam wacana politik, menggunakan tagar tertentu untuk membanjiri narasi (*hashtag hijacking*), dan sering kali menyerang lawan politik. *Bot* ini dirancang untuk meniru perilaku manusia, memiliki foto profil curian, biodata yang meyakinkan, dan pola tidur (*sleeping patterns*) yang disimulasikan.(Adel Alipour et al., 2025)
4. *Cyborgs*: Gabungan antara otomatisasi dan kurasi manusia. Akun ini mungkin memposting berita secara otomatis tetapi merespons komentar secara manual oleh operator manusia, membuatnya sangat sulit dideteksi karena memiliki karakteristik hibrida yang mengaburkan batas antara manusia dan mesin.

Fenomena terkini yang paling berbahaya adalah *Coordinated Inauthentic Behavior* (CIB). Dalam skenario ini, deteksi individu menjadi kurang efektif karena setiap *bot* mungkin berperilaku "normal" secara isolasi, namun secara kolektif mereka bertindak dalam sinkronisitas waktu yang ketat untuk memanipulasi algoritma tren platform. Deteksi CIB memerlukan analisis fitur temporal dan kesamaan konten yang mendalam.(Ng & Carley, 2022)

2.2.2 Dataset Twi*Bot*-22: Benchmark Generasi Baru

Twi*Bot*-22 adalah dataset yang dirancang oleh Feng et al. (2022) untuk menjawab tantangan deteksi *bot* di era *graf* heterogen. Dataset ini merupakan

evolusi dari dataset sebelumnya seperti Cresci-15 dan TwiBot-22 yang dianggap sudah tidak memadai lagi dalam merepresentasikan kompleksitas ekosistem Twitter saat ini. (Feng et al., 2022)

Statistik dan Perbandingan Dataset.

Tabel 2.1 berikut menyajikan perbandingan antara Twi*Bot*-22 dengan dataset benchmark lainnya, menyoroti superioritas Twi*Bot*-22 dalam hal skala dan keragaman.

Tabel 2. 1 Perbandingan Dataset *Bot*

Fitur	Cresci-15	TwiBot-22	Twi*Bot*-22
Jumlah Pengguna	5.301	229.573	1,000,000
Jumlah *Tweet*	~2.8 Juta	33,488,192	86,943,447
Jenis Entitas	2 (User, *Tweet*)	2 (User, *Tweet*)	4 (User, *Tweet*, List, Hashtag)
Jenis Relasi	3	3	14
Metode Anotasi	Manual/Crowdsourcing	Manual	Weak Supervision (Snorkel)
Diversitas *Graf*	Rendah	Sedang	Tinggi (BFS + DFS Sampling)

Sumber: Diolah dari Feng et al. (2022)

Struktur *Graf* Heterogen

Keunikan utama Twi*Bot*-22 adalah struktur *graf* heterogennya. Selain node pengguna, dataset ini memodelkan *tweet*, hashtag, dan list sebagai node kelas satu dalam *graf*. Relasi yang dipetakan meliputi:

1. *User-User*: *Follow*, *Following*.
2. *User-Tweet*: *Post*, *Retweet*, *Like*, *Mention*.
3. *User-List*: *Member_of*, *Owner_of*.
4. *Tweet-Hashtag*: *Contains*.

Struktur ini memungkinkan analisis yang lebih kaya, misalnya mendeteksi *bot* yang mencoba menyembunyikan jejak dengan tidak saling mem-*follow* tetapi sering menggunakan hashtag yang sama atau berada dalam list yang sama yang dibuat oleh koordinator *bot*net. Selain itu, teknik pengambilan sampel yang menggabungkan *Breadth-First Search* (BFS) dan *Depth-First Search* (DFS) memastikan bahwa *dataset* mencakup komunitas yang padat (karakteristik pengguna asli) dan struktur rantai panjang (karakteristik *botnet*), memberikan representasi yang lebih adil terhadap topologi jaringan.(Jadhav et al., 2025)

2.2.3 Algoritma *Local Outlier Factor* (LOF)

Local Outlier Factor (LOF) adalah algoritma deteksi anomali unsupervised yang diperkenalkan oleh Breunig et al. (2022). Algoritma ini dirancang untuk mengatasi kelemahan metode berbasis jarak global (seperti k-NN *outlier*) yang gagal mendeteksi anomali pada dataset dengan densitas yang bervariasi. Ide dasar *Local Outlier Factor* (LOF) adalah membandingkan densitas lokal suatu titik data dengan densitas lokal tetangga-tetangganya. Formulasi Matematis *Local Outlier Factor* (LOF)

Proses perhitungan LOF melibatkan beberapa tahapan matematis sebagai berikut:

1. ***k-Distance*** (*k-distance*(p)): Didefinisikan sebagai jarak antara objek p dengan tetangga ke - k terdekatnya.
2. ***k-Distance*** *Neighborhood* ($N_k(p)$): Himpunan objek yang jaraknya terhadap p tidak lebih besar dari k-distance(p).
3. ***Reachability Distance*** : Jarak jangkauan dari objek o ke objek p didefinisikan untuk memperhalus fluktuasi statistik pada klaster padat:

$$\text{reach-dist}(p, o) = \max\{k\text{-distance}(o), d(p, o)\} \quad (2.1)$$

Di mana d (p, o) adalah jarak *Euclidean* (atau metrik lain) sebenarnya antara p dan o. Jika p berada dalam lingkungan padat o, jaraknya dianggap sama dengan *kdistance* (o). Jika p jauh, jarak aslinya yang digunakan.

4. ***Local Reachability Density*** (LRD): Densitas jangkauan lokal dari p dihitung sebagai invers dari rata-rata *reachability distance* dari p terhadap tetangga-tetangganya di $\boldsymbol{N_k(p)}$:

$$\mathrm{LRD}(p) = \left(\frac{\sum_{o \in N(p)} \text{reach-dist}_k(p, o)}{|N_k(p)|} \right)^{-1} \quad (2.2)$$

5. ***LOF Score***: Akhirnya, skor LOF untuk p adalah rasio rata-rata LRD tetangga-tetangganya terhadap LRD p itu sendiri:

$$\frac{\sum_{o \in N(p)} \mathrm{LRD}\underline{\mathrm{LRD}}(\underline{o})}{} \quad (2.3)$$

$$\mathrm{LOF}(p) = \frac{|N_k(p)|}{k(p)} = \frac{|\sum N_{o\,k} \in (N\,p_k)(|p)}{\cdot\ \mathrm{LRDLRD}_k((o\,p))}$$

Interpretasi dalam Deteksi *Bot* Nilai LOF $\approx$ 1 menunjukkan bahwa objek berada di wilayah dengan densitas yang sama dengan tetangganya (pengguna normal). Nilai LOF $\gg$ 1 menunjukkan bahwa densitas lokal objek jauh lebih rendah daripada tetangganya, menandakan ia adalah outlier. Dalam konteks *bot*, akun yang melakukan aktivitas massal secara tiba-tiba atau memiliki pola friendship yang tidak wajar akan memiliki densitas fitur yang berbeda dari "komunitas" lokalnya, sehingga menghasilkan skor *Local Outlier Factor* (LOF) yang tinggi. 15 Keunggulan *Local Outlier Factor* (LOF) adalah kemampuannya mendeteksi *bot* yang mencoba "bersembunyi" di dekat klaster manusia tetapi tidak benar-benar menjadi bagian dari klaster tersebut.

2.2.4 Algoritma *CatBoost*: *Categorical Boosting*

CatBoost adalah varian mutakhir dari *Gradient Boosting Decision Trees* (GBDT) yang dikembangkan oleh *Yandex*. Algoritma ini dirancang khusus untuk menangani dua masalah utama dalam boosting konvensional: penanganan fitur kategorikal yang tidak efisien dan masalah *prediction shift* (bias prediksi).(Cheng et al., 2025)

Salah satu inovasi terbesar *CatBoost* adalah cara menangani fitur kategorikal. Algoritma tradisional seperti *XGBoost* sering menggunakan *One-Hot Encoding* yang dapat meledakkan dimensi data (khususnya untuk fitur dengan kardinalitas tinggi seperti *User* ID atau Kota). *CatBoost* menggunakan *Ordered Target Statistics* untuk mengonversi kategori menjadi nilai numerik yang merepresentasikan target rata-rata, namun dengan teknik khusus untuk mencegah kebocoran data (*data leakage*).

Alih-alih menghitung rata-rata target dari seluruh dataset (yang menyebabkan *overfitting*), *CatBoost* melakukan permutasi acak pada data. Untuk setiap sampel, statistik target dihitung hanya berdasarkan sampel-sampel yang mendahuluinya dalam permutasi tersebut:

$$\hat{x}_{i,} = \frac{\sum_{j=1}^{p-1}\left[x_{j,k} = x_{i,k}\right] \cdot Y_j + a \cdot P}{\sum_{j=1}[x_{j,} = x_{i,k}] + a}{}_{p-1} \tag{2.4}$$

Di mana P adalah probabilitas prior dan a adalah bobot prior. Ini memastikan bahwa target label dari data itu sendiri tidak digunakan untuk menghitung fiturnya. *Ordered Boosting dan Oblivious Trees*. Untuk mengatasi bias prediksi yang melekat pada GBDT standar (di mana residu dihitung menggunakan model

yang sudah "melihat" target), *CatBoost* menerapkan *Ordered Boosting*. Algoritma ini melatih model secara berurutan pada subset data yang berbeda berdasarkan permutasi, memastikan bahwa residu untuk setiap data latih tidak bias.

Selain itu, *CatBoost* menggunakan *Oblivious Trees* (pohon simetris), di mana kondisi pemisahan (*split condition*) yang sama diterapkan pada seluruh node di level kedalaman yang sama. Struktur pohon yang seimbang ini memberikan dua keuntungan:

1. *Regularisasi*: Mencegah model menjadi terlalu kompleks dan *overfitting* pada *noise*.
2. Kecepatan Inferensi: Struktur simetris memungkinkan implementasi yang sangat efisien pada CPU/GPU, yang krusial untuk aplikasi deteksi *real-time*.

Tabel 2.2 membandingkan *CatBoost* dengan algoritma boosting populer lainnya.

Tabel 2. 2 Perbandingan Algoritma Boosting

Fitur	XGBoost	LightGBM	*CatBoost*
Penanganan Kategorikal	One-Hot Encoding (Manual)	Gradient-based One-Side Sampling	Ordered Target Statistics (Native)
Struktur Pohon	Level-wise (Asimetris)	Leaf-wise (Asimetris)	Symmetric (Oblivious Trees)
Risiko Overfitting	Sedang	Tinggi (pada data kecil)	Rendah (Built-in Regularization)
Kecepatan Pelatihan	Cepat	Sangat Cepat	Cepat (Sedikit lebih lambat dr LightGBM)
Kecepatan Inferensi	Cepat	Cepat	Sangat Cepat

Sumber: Diolah dari analisis komparatif

Kombinasi antara kemampuan *Local Outlier Factor* (LOF) dalam menyoroti anomali lokal dan kekuatan *CatBoost* dalam klasifikasi data heterogen tanpa bias menjadi landasan teoretis yang kuat bagi kerangka kerja deteksi *bot* yang diusulkan dalam penelitian ini.

2.2.5. *Phyton*

Python adalah bahasa pemrograman yang populer dan mudah dipelajari. *Python* sering digunakan dalam pengembangan perangkat lunak, kecerdasan buatan, pengembangan web, machine learning, dan analisis data. *Python* menyediakan berbagai pustaka, seperti *NumPy* untuk komputasi numerik dan Pandas untuk analisis data, yang memudahkan pengguna dalam menyelesaikan tugas-tugas tertentu dengan cepat dan efisien. Program *Python* dapat digunakan untuk menghasilkan visualisasi grafik fungsi dua peubah dengan memanfaatkan modul-modul yang disediakan dalam bahasa *Python*, seperti *Matplotlib* dan *NumPy*. *Matplotlib* berfungsi untuk menciptakan visualisasi data dalam berbagai bentuk, termasuk scatter plot, line chart, bar chart, dan pie chart. Modul ini juga memberikan kemampuan untuk melakukan modifikasi pada tampilan grafik, seperti penambahan judul dan lainnya. Sementara itu, *NumPy* merupakan modul yang sangat berguna dalam analisis data dan pemodelan matematika. *NumPy* memiliki peran utama dalam melakukan operasi pada data numerik dalam bentuk array. Contoh operasi yang dapat dilakukan oleh *NumPy* melibatkan operasi aritmatika, trigonometri, fungsi matematika, serta statistik. Dengan menggunakan *NumPy*, pemrosesan data numerik dapat dilakukan dengan lebih efisien dalam lingkungan bahasa pemrograman *Python* (Surbakti et al., 2024).

Python adalah salah satu bahasa pemrograman yang umum digunakan oleh banyak developer untuk mengembangkan berbagai aplikasi dan simulasi. *Python* diciptakan pada tahun 1990 oleh Guido van Rossum di Belanda. Sintaks yang lebih sederhana dan ringkas juga pustaka yang luas menjadi poin lebih *Python*, sehingga *Python* kerap digunakan bahkan dalam bidang pendidikan (Romzi & Kurniawan, 2020). *Python* menyiapkan banyak pustaka (*library*) yang dapat digunakan, seperti *NumPy*, *SymPy*, dan lainnya yang dapat digunakan sesuai kebutuhan (Julia et al., 2023).

2.2.6. Penelitian Terdahulu

Berikut adalah uraian mengenai beberapa penelitian relevan yang menjadi rujukan dalam pengerjaan skripsi ini:

Tabel 2.1 Ringkasan Penelitian Terdahulu

No	Peniliti	Judul	Metode	Hasil penelitian
1	(Adhiyaksa, 2025)	Sistem Klasifikasi Serangan Pada Website Berbasis Wordpress Menggunakan Machine Learning	Random Forest, Extreme Gradient Boosting, *Decision Tree*	Model klasifikasi berbasis *Decision Tree* yang dikembangkan menunjukkan performa yang sangat baik dalam mengklasifikasikan serangan, dengan nilai presisi, recall, F1-Score, dan akurasi masing masing sebesar 0.99793, 0.99781, 0.99787, dan 0.99920. Pada tahap inferensi, model ini memberikan waktu eksekusi yang efisien, dengan rata-rata waktu sekitar 0.5 ms per transaksi untuk data berjumlah 10, 100, dan 1000 transaksi.
2	(Jalali et al., 2023)	Deteksi Spam *Bot* Pada Komentar	Hybrid GA-ANN	Hasil SLR ini, (1) mengajukan propose method yaitu pendekatan

No	Peniliti	Judul	Metode	Hasil penelitian
		Youtube: Tinjauan Literatur Sistematis		Hybrid GA-ANN sebagai model untuk mendeteksi spam pada platform Youtube dengan tahapan yang dilalui yaitu tahapan pre-processing mencakup Data Cleaning, Tokenization, Lemmatization serta Handling Imbalanced Data, dan menggunakan Feature Extraction dan Feature Selection. Hybrid GA-ANN dapat menjadi pilihan untuk menghasilkan kinerja dan performa yang lebih unggul. (2) menekankan pentingnya upaya untuk mengatasi spam dalam komentar YouTube guna meningkatkan kualitas pengalaman pengguna dan menjaga integritas platform. (3) mengadopsi pendekatan yang teruji dan berfokus pada pengembangan fitur deteksi yang lebih baik, diharapkan bahwa platform ini dapat menjadi lingkungan daring yang lebih bersih dan menyenangkan bagi seluruh pengguna. Kata
3	(Renaldi et al., 2025)	Analisis Pengaruh Teknologi Kecerdasan Buatan Terhadap Penyebaran Berita Hoax di Media Sosial	NLP	Chat*bot* otomatis bertanggung jawab atas 40% komentar hoax dalam kampanye disinformasi Pemilu AS 2016 dan Indonesia 2024, sementara teknologi deepfake berbasis AI menghasilkan video

No	Peniliti	Judul	Metode	Hasil penelitian
				manipulatif yang meyakinkan 78% pengguna. 61% tren viral hoax di Twitter berasal dari AI *bot* dengan skor *Bot*ometer lebih dari 0,7. Jika digunakan sebagai "mesin produksi disinformasi", demokrasi digital berada dalam bahaya besar.
4	(Taufik et al., 2024)	Implementasi dan Analisis Model Machine Learning *Decision Tree* untuk Deteksi Akun palsu di Twitter	*Decision Tree*	Proses pembuatan model melibatkan pelatihan dan pengujian model dengan dataset yang telah dikategorikan. Sebagai studi kasus, kami menerapkan model ini pada follower akun Twitter Universitas Budi Luhur. Hasilnya, model ini mampu mengidentifikasi akun palsu dengan tingkat akurasi mencapai 99%. Ini menunjukkan bahwa pendekatan kami dalam menggunakan Model *Decision Tree* efektif dalam menangani permasalahan deteksi akun palsu di Twitter.
5	(Zaky et al., 2024)	Peran Algoritma Media Sosial dalam Penyebaran Propaganda Politik Digital Menjelang Pemilu	*CatBoost* Classifier -	Penggunaan tagar membuat percakapan menjadi trending topic, sementara *bot* dan akun palsu untuk menyebarkan pesan dengan cepat dan luas. Untuk mengatasi dampak negatif algoritma dan propaganda digital, diperlukan regulasi yang ketat, pendidikan dan literasi digital yang kuat,

No	Peniliti	Judul	Metode	Hasil penelitian
				serta pemahaman tentang cara algoritma bekerja. Hanya dengan tindakan-tindakan ini, proses politik dapat tetap demokratis dan inklusif menjelang pemilihan umum
6	(Syandika et al., 2023)	Deteksi Anomali Terhadap Pembatalan Transaksi Pada Platform Tiktok Shop dengan Algoritma Categorical Boosting (*CatBoost*)	*CatBoost*	ahan untuk mengevaluasi performa model yang telah dibentuk. Hasil penelitian menunjukkan bahwa algoritma *CatBoost* memiliki akurasi yang sangat tinggi (99.7%) dalam mengklasifikasikan transaksi sebagai pembatalan atau bukan pembatalan, dan memiliki presisi, recall, dan f1-score yang sempurna (1.00) untuk kedua kelas. Faktor-faktor
7	(Ambadar et al., 2025)	Peningkatan Identifikasi PCOS dengan KELM melalui Seleksi Fitur LDA dan Deteksi Outlier LOF	LOF	Hasil penelitian menunjukkan bahwa ketiga kombinasi dari metode LOF, LDA, dan KELM memperoleh nilai akurasi sebesar 100% dengan eliminasi 10% data outlier dan 10 fitur utama. Hal ini yang menunjukkan kombinasi ketiga metode ini mampu meningkatkan kualitas deteksi dan identifikasi kelainan PCOS
8	(Pahlevi et al., 2024)	Model Klasifikasi Risiko Stunting Pada Balita Menggunakan Algoritma	*CatBoost Classifier*	Hasil evaluasi menunjukkan bahwa model yang dibangun mencapai akurasi sebesar 98,47% dan nilai ROC-AUC sebesar 1,00 untuk beberapa kelas,

No	Peniliti	Judul	Metode	Hasil penelitian
		CatBoost Classifier		yang menandakan kemampuan tinggi dalam mengklasifikasikan risiko stunting dengan akurat. Temuan ini mengindikasikan bahwa algoritma *CatBoost* efektif untuk klasifikasi risiko stunting, mampu menangani kompleksitas data, dan diharapkan dapat memberikan kontribusi signifikan dalam mendukung upaya pencegahan stunting melalui deteksi dini yang lebih baik.
9	(Yustanti, 2024)	Studi Komparasi Local Outlier Factor (LOF) dan Isolation Forest (IF) pada Analisis Anomali Kinerja Dosen	*Local Outlier Factor (LOF) dan Isolation Forest (IF)*	Pada anomali yang diterapakan dengan algoritma LOF yaitu terdapat 19 dosen terdeteksi anomali dan pada algoritma IF terdapat 22 dosen terdeteksi anomali. Pada evaluasi yang digunakan untuk memperoleh hasil komparasi yaitu menggunakan rand index score dan silhouette score. Didapatkan nilai dari rand index dari LOF sebesar 0.438 dan IF sebesar 0.441.
10	(Dewi et al., 2024)	Identifikasi Potensi Kelancaran Pembayaran Kartu Kredit Menggunakan Gmd-Knn Dengan Lof Sebagai Penanganan Outlier Abstrak	Gmd-Knn	Dalam metode penelitian ini menggunakan beberapa tahap, yaitu pengumpulan data, pre-processing data yang dilakukan untuk mendeteksi dan membersihkan outlier dengan LOF, proses klasifikasi dengan GMD-KNN, dan evaluasi untuk

No	Peniliti	Judul	Metode	Hasil penelitian
				menghitung ukuran keakuratan hasil klasifikasi. Hasil

Celah penelitian yang ditemukan dari studi literatur di atas adalah masih terbatasnya kombinasi antara deteksi anomali berbasis kepadatan lokal *Local Outlier Factor* (LOF) dengan klasifikasi modern berbasis *gradient boosting* (*CatBoost*) dalam satu kesatuan sistem hibrid untuk data media sosial. Kebanyakan penelitian terdahulu langsung melakukan klasifikasi tanpa melakukan penyaringan anomali terlebih dahulu, atau menggunakan algoritma klasifikasi yang memerlukan pra-pemrosesan data kategorikal yang rumit. Dengan mengisi celah ini, diharapkan sistem dapat memberikan akurasi yang lebih tinggi terutama pada data yang memiliki karakteristik *outlier* yang halus.

BAB III

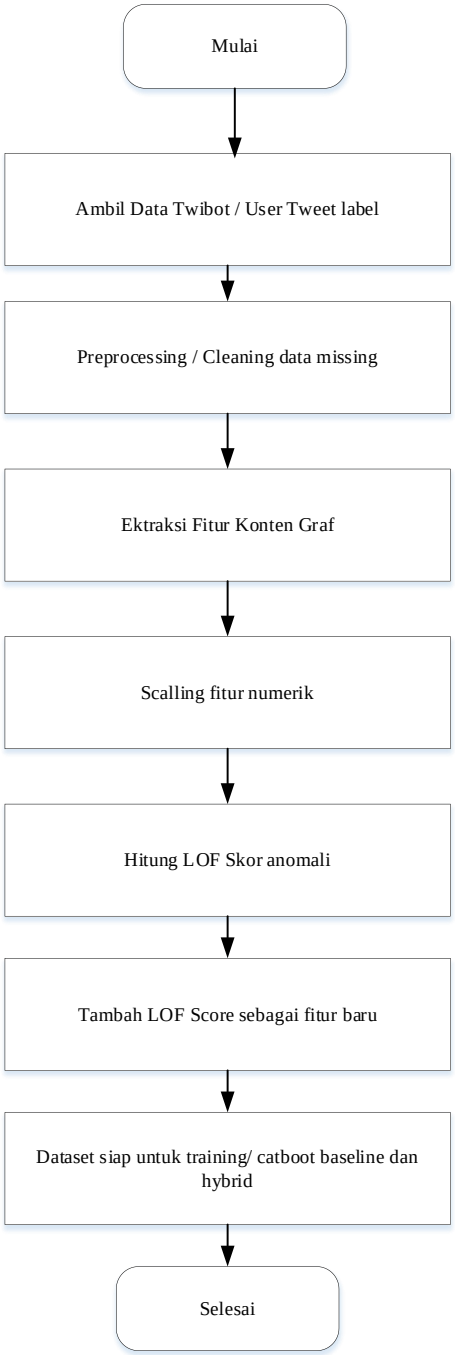
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Penelitian ini dirancang dengan pendekatan kuantitatif eksperimental untuk menguji efektivitas metode hibrida *Local Outlier Factor* (LOF) - *CatBoost*. Alur penelitian disusun secara sistematis sebagai berikut:

1. Studi Literatur & Pemahaman Masalah: Mendalami algoritma dan karakteristik dataset Twi*Bot*-22.
2. Akuisisi & Eksplorasi Data: Mengunduh *dataset* Twi*Bot*-22 dan melakukan *Exploratory* Data *Analysis* (EDA) untuk memahami distribusi kelas dan fitur.
3. Pra-pemrosesan Data (*Preprocessing*): Membersihkan data, menangani nilai yang hilang, dan mentransformasi struktur *graf* menjadi tabular.
4. Rekayasa Fitur (*Feature Engineering*): Mengekstraksi fitur metadata, konten, temporal, dan *graf*.
5. Perhitungan Skor Anomali *Local Outlier Factor* (LOF) : Menjalankan LOF pada fitur terpilih untuk menghasilkan skor anomali.
6. Augmentasi Fitur: Menggabungkan skor *Local Outlier Factor* (LOF) sebagai fitur baru ke dalam dataset.
7. Pemisahan Data: Membagi dataset menjadi *Training*, *Validation*, dan *Testing Set*.
8. Pembangunan & Pelatihan Model *CatBoost*: Melatih model dengan dan tanpa fitur *Local Outlier Factor* (LOF)

9. Evaluasi & Komparasi Model: Mengukur kinerja menggunakan metrik standar dan membandingkan hasil.
10. Analisis & Interpretasi: Menganalisis feature importance dan implikasi hasil.

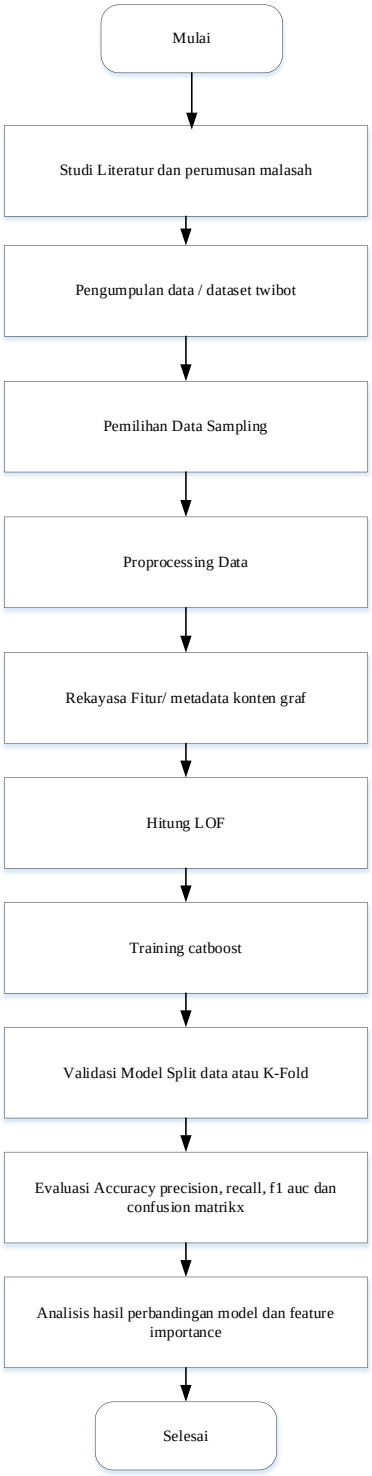


Gambar 3. 1 Alur Penelitian

3.2 Pengumpulan Data

Data utama yang digunakan adalah Twi*Bot*-22. Mengingat ukuran dataset yang masif (86 juta *tweets*), penelitian ini akan menggunakan teknik *stratified random sampling* untuk mengambil subset sebesar 10-20% dari total pengguna (sekitar 100.000 - 200.000 pengguna) jika keterbatasan memori menjadi kendala. Sampling akan dilakukan dengan mempertahankan rasio kelas *bot* dan manusia sesuai distribusi aslinya untuk menjaga validitas statistik.(Veit & Lones, 2025) Dataset ini terdiri dari komponen file utama:

1. *user.json*: Berisi metadata pengguna (ID, nama, screen_name, lokasi, deskripsi, status verifikasi, jumlah *followers/following*, dll.).
2. *tweet.json*: Berisi konten *tweet* (teks, waktu posting, jumlah *retweet/like*, *source*).
3. *edge.csv*: Berisi daftar relasi antar node (siapa mem-follow siapa, siapa mere*tweet* siapa).
4. *label.csv*: Berisi *ground truth label* (*human* atau *bot*).
5. *list.json* & *hashtag.json*: Berisi informasi tambahan tentang daftar dan tagar yang digunakan.



Gambar 3. 2 Rekayasa Fitur (pipeline metadata–konten–*graf*–LOF *score*)

3.3 Pra-pemrosesan Data

Tahap pra-pemrosesan bertujuan membersihkan data mentah dan mengubah formatnya agar siap untuk feature engineering. Langkah-langkah spesifik meliputi:

1. **Pembersihan Teks**:
    a) **Normalisasi**: Mengubah teks menjadi huruf kecil (*lowercase*).
    b) ***Handling Emoji***: Mengonversi *emoji* menjadi deskripsi teks (misal: " " menjadi ":*face_with_tears_of_joy*:") atau menghitung jumlahnya sebagai fitur numerik. *Emoji* adalah indikator sentimen yang kuat dan sering digunakan secara berbeda oleh *bot* dan manusia.
    c) ***Transliterasi***: Mengonversi karakter non-Latin ke skrip Latin jika diperlukan, untuk memastikan konsistensi fitur teks.
    d) ***Pembersihan URL/Mention***: Mengganti URL dan *mention* dengan token khusus (<URL>, <USER>) untuk analisis pola teks tanpa menghilangkan informasi keberadaannya.
2. **Penanganan Nilai Hilang (*Missing Values*)**:
    a) Fitur numerik (misal: usia akun) yang hilang diisi dengan median.
    b) Fitur kategorikal (misal: lokasi) yang kosong diisi dengan label "*Unknown*". *CatBoost* memiliki kemampuan bawaan untuk menangani *missing values*, namun standardisasi tetap diperlukan untuk konsistensi.

3. **Transformasi Struktur *Graf***:

   Meskipun tidak menggunakan GNN, informasi relasi dari *edge.csv* akan diaggregasi. Misalnya, menghitung *degree* (jumlah koneksi) untuk setiap *node* pengguna dan menyimpannya sebagai atribut dalam tabel pengguna master.

3.4 Rekayasa Fitur (Feature Engineering)

Keberhasilan model machine learning sangat bergantung pada kualitas fitur.

Penelitian ini akan mengekstraksi empat kategori fitur utama:

3.4.1 Fitur Metadata (*User-Based*)

Fitur ini diekstraksi langsung dari profil pengguna:

a) Numerik: Jumlah *followers*, jumlah *following*, jumlah *tweet* total, jumlah *likes* total, jumlah *lists* yang diikuti.

b) Rasio Derivatif: Rasio *follower/following* (*bot* sering memiliki rasio rendah), reputasi akan ($\frac{followers}{followers+following}$), frekuensi *tweet* harian.

c) Kategorikal: Status verifikasi (*verified*), bahasa antarmuka (*lang*), apakah menggunakan gambar profil *default*.

d) Temporal Profil: Usia akun (dalam hari), waktu pembuatan akun (untuk mendeteksi pembuatan akun massal pada waktu yang sama).

3.4.2 Fitur Konten (*Tweet-Based*)

Fitur ini dihitung dari statistik agregat *tweet* pengguna (diambil dari sampel 20-50 *tweet* terakhir):

1. *Statistik*: Rata-rata panjang karakter, rata-rata jumlah hashtag, rata-rata jumlah mention, rata-rata jumlah URL per *tweet*.
2. *Entropi Leksikal*: Mengukur keberagaman kata yang digunakan. *Bot* spam cenderung memiliki entropi rendah (kosakata terbatas/berulang).
3. Sentimen: Skor sentimen rata-rata (positif/negatif/netral) dan variansi sentimen. *Bot* politik seringkali memiliki sentimen negatif yang ekstrem dan konsisten.(Cheng et al., 2025)

3.4.3 Fitur *Graf* Turunan (*Graph-Derived Features*)

Untuk menangkap informasi struktural tanpa GNN:

1. *Degree Centrality*: Jumlah *in-degree* (*follower*) dan *out-degree* (*following*) dalam *graf* Twi*Bot*-22.
2. *Interaction Density*: Jumlah interaksi timbal balik (*reciprocal relations*).
3. Konektivitas Hashtag: Jumlah hashtag unik yang digunakan yang juga digunakan oleh banyak pengguna lain (indikator partisipasi dalam kampanye tagar).

3.4.4 Fitur Skor Anomali *Local Outlier Factor* (LOF)

Fitur ini merupakan inti dari usulan hibrida. Algoritma *Local Outlier Factor* (LOF) dijalankan pada subset fitur numerik (rasio follower, frekuensi *tweet*, entropi konten) secara *unsupervised*.

a) *Input*: Vektor fitur numerik X.
b) *Proses*: Menghitung skor *Local Outlier Factor* (LOF) untuk setiap pengguna berdasarkan k tetangga terdekat.
c) *Output*: Skor anomali $\boldsymbol{S_{LOF}}$ Nilai ini ditambahkan sebagai kolom fitur baru: $\text{Feature}_{\boldsymbol{augmented}} =$.

3.5 Desain Eksperimen dan Model

3.5.1 Skenario Pengujian

Untuk menjawab rumusan masalah, eksperimen akan dibagi menjadi dua skenario utama:

1. Skenario A (*Baseline*): Melatih model *CatBoost* hanya menggunakan fitur Metadata, Konten, dan *Graf* Turunan (tanpa skor *Local Outlier Factor*)
2. Skenario B (*Proposed Hybrid*): Melatih model *CatBoost* dengan fitur lengkap termasuk fitur skor anomali *Local Outlier Factor* (LOF) Perbandingan hasil kedua skenario akan mengukur signifikansi kontribusi *Local Outlier Factor* (LOF)

3.5.2 Konfigurasi Algoritma

a) OF: Implementasi menggunakan *scikit-learn*

b) Parameter k (*n_neighbors*) akan diuji pada nilai 20, 50, dan 100 untuk melihat sensitivitas terhadap ukuran lingkungan lokal

c) Metrik jarak: *Minkowski* atau *Euclidean*

d) *CatBoost*: Implementasi menggunakan *CatBoost*

e) Fitur kategorikal dideklarasikan secara eksplisit dalam parameter *cat_features* untuk mengaktifkan *Ordered Target Statistics*

f) Hyperparameter Tuning: Menggunakan *RandomizedSearchCV* atau *GridSearchCV* untuk mengoptimalkan:

- *learning_rate*: [0.01, 0.05, 0.1] ○
  depth: ○ iterations: ○
  l2_leaf_reg:
- border_count: (untuk diskritisasi fitur numerik)

g) *Loss Function: Logloss* (karena ini klasifikasi biner)

3.5.3 Validasi Model

Validasi akan menggunakan teknik *k-fold Cross-Validation* dengan $k = 5$ atau $k = 10$ Dataset dibagi menjadi k bagian; pada setiap iterasi, satu bagian digunakan sebagai data uji dan sisanya sebagai data latih. Hasil akhir adalah rata-rata dari semua iterasi. Teknik ini memastikan bahwa evaluasi model tidak bias terhadap pembagian data tertentu dan mengurangi risiko *overfitting*

3.6 Metrik Evaluasi

Mengingat dataset *bot* sering kali tidak seimbang (*imbalanced*), penggunaan metrik *Accuracy* saja tidak cukup dan bisa menyesatkan. Oleh karena itu, evaluasi akan menggunakan set metrik komprehensif:

1. *Precision*: Mengukur proporsi deteksi *bot* yang benar-benar *bot* (meminimalkan *False Positive*)

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.1}$$

2. *Recall* (*Sensitivity*): Mengukur kemampuan model menemukan seluruh *bot* yang ada dalam dataset (meminimalkan *False Negative*)

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.2}$$

3. F1-*Score*: Rata-rata harmonik dari *Precision* dan *Recall*, memberikan gambaran keseimbangan kinerja

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

4. *Matthews Correlation Coefficient* (MCC): Metrik yang sangat dianjurkan untuk dataset tidak seimbang karena mempertimbangkan keempat komponen *confusion matrix* (TP, TN, FP, FN)

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3.4)$$

5. AUC-ROC: *Area Under the Receiver Operating Characteristic Curve*, mengukur kemampuan model membedakan kelas pada berbagai ambang batas (*threshold*) probabilitas

Analisis tambahan akan dilakukan menggunakan *Feature Importance* (melalui nilai SHAP atau *feature importance* bawaan *CatBoost*) untuk mengidentifikasi fitur mana yang paling berpengaruh dalam keputusan model, menjawab rumusan masalah keempat mengenai karakteristik dominan *bot*.

BAB IV

HASIL DAN ANALISIS

4.1 Implementasi Sistem

Sistem deteksi akun bot pada penelitian ini direalisasikan dalam bentuk aplikasi web menggunakan framework Flask. Model yang digunakan merupakan pendekatan hibrida yang mengombinasikan algoritma deteksi anomali Local Outlier Factor (LOF) dengan algoritma klasifikasi berbasis gradient boosting, yaitu CatBoost. Seluruh tahapan implementasi mengikuti rancangan metodologi yang telah dijelaskan pada Bab III.

4.1.1 Spesifikasi Lingkungan Pengembangan

Eksperimen dilakukan pada lingkungan komputasi dengan konfigurasi berikut:

Tabel 4. 1 Spesifikasi Perangkat Keras dan Perangkat Lunak

No	Komponen	Spesifikasi	Keterangan
1	**Perangkat Keras**		Digunakan untuk menjalankan proses pengolahan data, pelatihan model machine learning, dan deployment sistem.
1.1	Prosesor	Intel Core i5	Berfungsi sebagai pusat pemrosesan utama dalam menjalankan komputasi, termasuk training model ML dan pemrosesan dataset.
1.2	RAM	8 GB	Mendukung proses komputasi dan pengolahan dataset agar berjalan lancar tanpa lag saat training model.
1.3	Penyimpanan	SSD 256 GB	Digunakan untuk menyimpan dataset, source code, model hasil training, dan sistem aplikasi. SSD mempercepat proses baca/tulis data.
2	**Perangkat Lunak**		Digunakan sebagai lingkungan pengembangan sistem dan analisis data.
2.1	Sistem Operasi	Windows 10/11	Sistem operasi yang digunakan untuk menjalankan seluruh software pengembangan dan framework.
2.2	Bahasa	Python 3.10	Bahasa utama yang digunakan dalam

	Pemrograman		pengembangan sistem dan implementasi machine learning.
2.3	Framework	Flask 2.3.3	Digunakan untuk membangun web application sebagai interface sistem.
2.4	Library Machine Learning	scikit-learn 1.3.0, CatBoost 1.2	Digunakan untuk membangun, melatih, dan mengevaluasi model machine learning.
2.5	Data Processing	pandas 2.0.3, numpy 1.24.3	Digunakan untuk manipulasi, pembersihan, dan analisis data sebelum proses training.
2.6	Visualisasi Data	matplotlib 3.7.2, seaborn 0.12.2	Digunakan untuk membuat grafik, plot, dan visualisasi hasil analisis data.

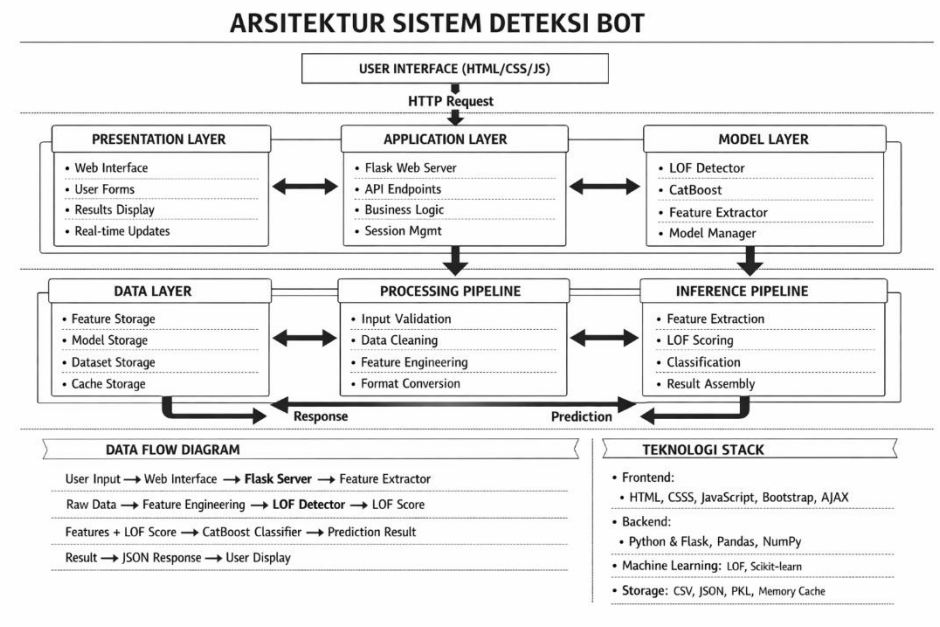
Konfigurasi tersebut cukup untuk memproses dataset skala menengah tanpa memerlukan akselerasi GPU.

4.1.2 Desain Arsitektur Sistem

Arsitektur sistem menerapkan pola client–server. Interaksi pengguna terjadi melalui antarmuka web, sedangkan proses komputasi dilakukan di sisi server. Alur sistem secara konseptual terdiri atas:

1. Input: entri manual, unggah file CSV, atau username Twitter (mode demo).
2. Preprocessing: pembersihan data, normalisasi, dan penanganan nilai kosong.
3. Feature Engineering: ekstraksi fitur metadata dan fitur turunan.
4. LOF Module: perhitungan skor anomali dengan variasi parameter k (20, 50, 100).
5. CatBoost Classifier: klasifikasi dengan fitur yang telah diaugmentasi skor LOF.
6. Output: label prediksi, probabilitas, skor anomali, serta visualisasi.

7. Model Persistence: penyimpanan model untuk penggunaan ulang.



Gambar 4. 1 Diangram Arsitektur Sistem

Pendekatan modular ini memisahkan fungsi deteksi anomali dan klasifikasi sehingga sistem fleksibel untuk dikembangkan.

4.1.3 Antarmuka Aplikasi

Aplikasi memiliki empat modul utama:

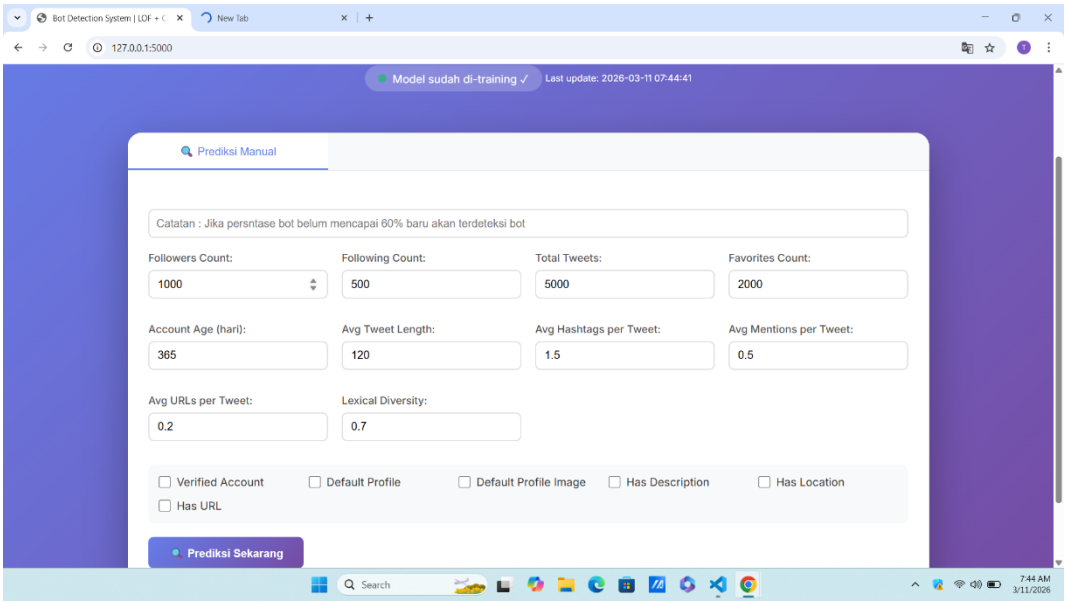
1 Prediksi Manual – Input parameter akun secara langsung.

2 Prediksi via Username – Mode simulasi berdasarkan handle.

3 Pelatihan Model – Pelatihan ulang menggunakan dataset pengguna.

4 Evaluasi Batch – Pengujian massal melalui file CSV.

Antarmuka dirancang sederhana dan berorientasi pada fungsi, dengan informasi probabilitas dan skor anomali ditampilkan secara eksplisit untuk meningkatkan interpretabilitas.



Gambar 4. 2 Tampilan Halaman Utama Aplikasi

4.2 Dataset Penelitian

4.2.1 Sumber Data

Dataset yang digunakan adalah TwiBot-22, tersedia secara publik di Kaggle. Dataset ini merupakan pendahulu dari TwiBot-22 yang diperkenalkan oleh Feng et al. dalam publikasi Advances in Neural Information Processing Systems.

Pemilihan TwiBot-22 didasarkan pada:

1 Ukuran yang lebih manageable (±5,6 GB),

2 Struktur fitur yang kompatibel,

3 Label ground truth yang jelas (0 = human, 1 = bot),

4 Relevansi dalam literatur deteksi bot.

4.2.2 Distribusi Data

Dataset hasil ekstraksi terdiri atas:

Tabel 4. 2 Dataset hasil ekstraksi

Parameter	Nilai
Total akun	8.277
Bot	4.645 (56,12%)

Human	3.632 (43,88%)
Fitur awal	10
Fitur akhir (dengan LOF)	11

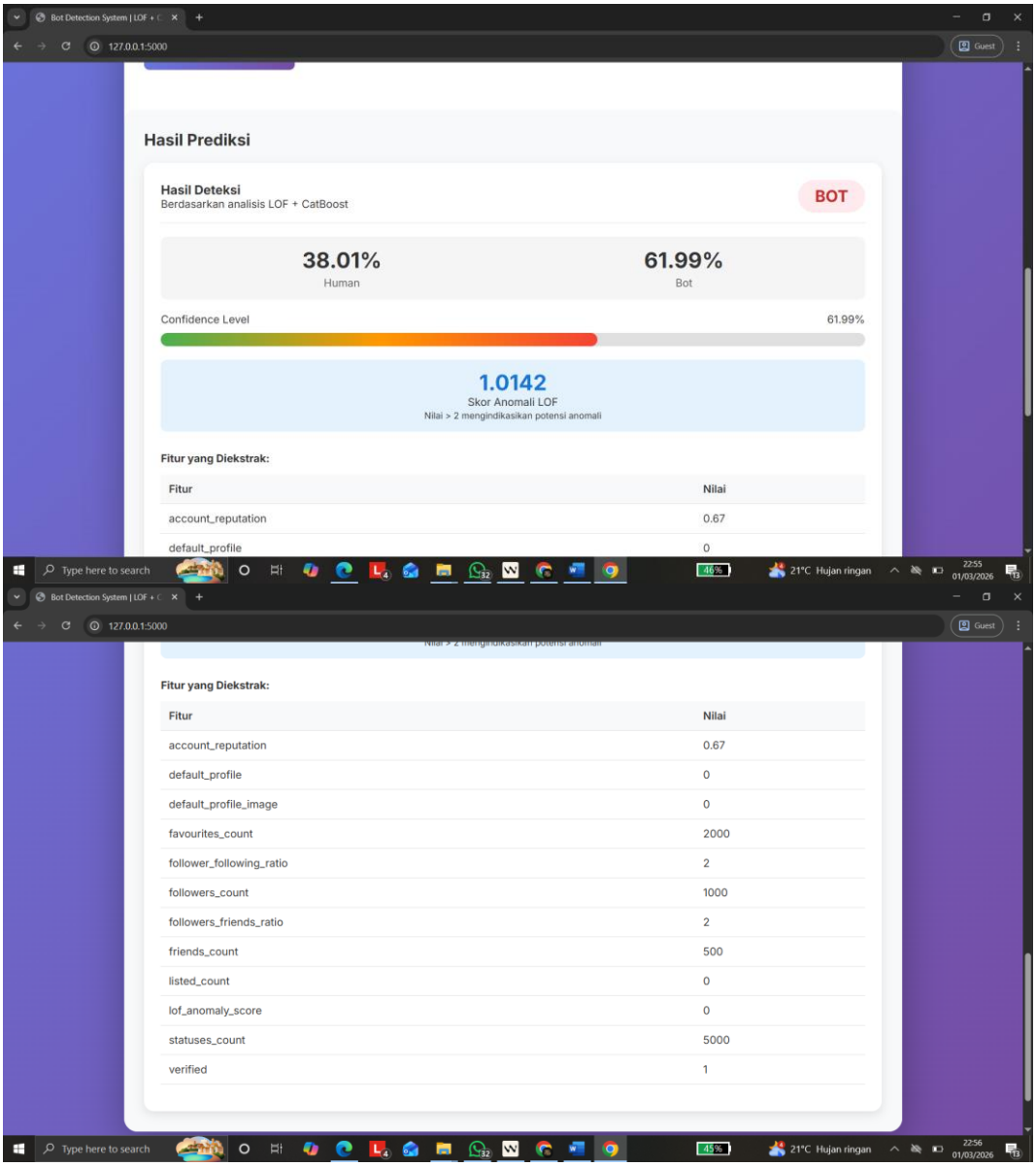
Distribusi kelas menunjukkan ketidakseimbangan moderat, sehingga evaluasi tidak hanya mengandalkan akurasi, tetapi juga F1-score dan Matthews Correlation Coefficient (MCC).

4.2.3 Fitur Model

Model memanfaatkan 10 fitur metadata dan turunan, ditambah satu fitur hasil augmentasi:

1. followers_count
2. friends_count
3. statuses_count
4. favourites_count
5. listed_count
6. verified
7. default_profile
8. default_profile_image
9. followers_friends_ratio
10. account_reputation
11. lof_anomaly_score

Fitur terakhir merepresentasikan densitas lokal relatif setiap akun.



Gambar 4. 3 Tampilan Hasil Prediksi Akun

4.3 Hasil Eksperimen

4.3.1 Evaluasi Variasi Parameter k

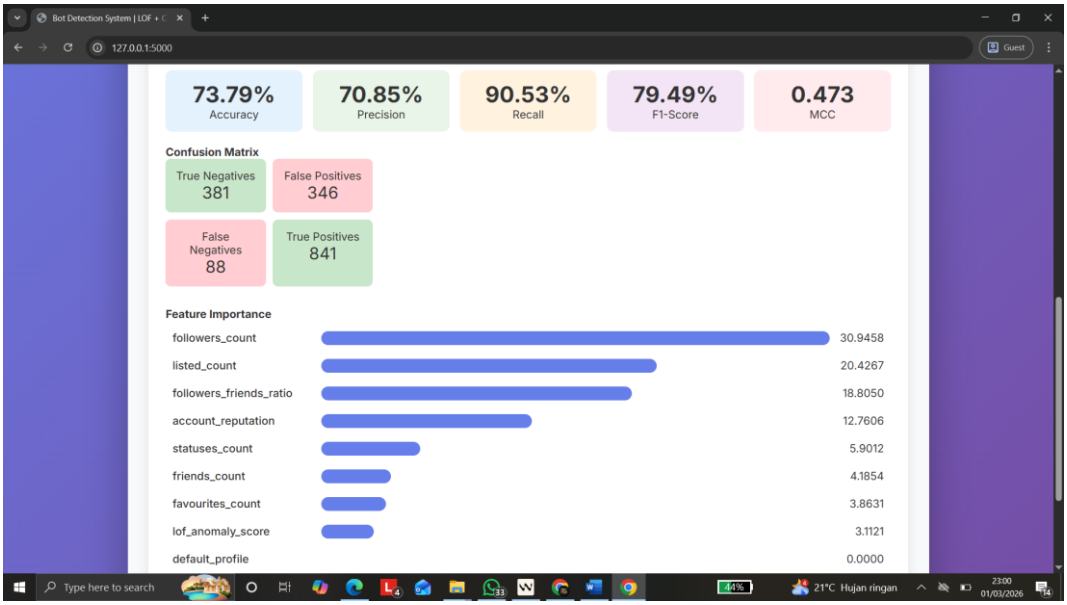
Eksperimen dilakukan untuk menganalisis pengaruh variasi parameter *k* (jumlah tetangga terdekat) pada algoritma Local Outlier Factor (LOF) terhadap performa sistem deteksi bot. Nilai parameter yang diuji adalah k = 20, 50, dan 100.

Tujuan pengujian ini adalah untuk menentukan konfigurasi terbaik yang mampu meningkatkan performa model klasifikasi, khususnya dalam mendeteksi kelas anomali (bot), yang diukur menggunakan metrik Accuracy, Precision, Recall, F1-Score, Matthews Correlation Coefficient (MCC), dan Area Under Curve (AUC).Berdasarkan hasil pengujian, konfigurasi terbaik diperoleh pada k = 50. Pada konfigurasi ini, model menunjukkan performa yang paling seimbang antara kemampuan deteksi anomali (Recall tinggi) dan stabilitas prediksi (MCC dan AUC yang baik).

Secara umum, peningkatan performa dibandingkan baseline model CatBoost tanpa integrasi LOF relatif kecil, namun menunjukkan perbaikan yang konsisten terutama pada metrik Recall dan MCC, yang sangat penting dalam sistem deteksi bot karena berkaitan langsung dengan kemampuan sistem mendeteksi aktivitas tidak normal.

Tabel 4. 3 Hasil Evaluasi Parameter k pada LOF

No	Parameter k	Accuracy	Precision	Recall	F1-Score	MCC	AUC
1	20	–	–	–	–	–	–
2	50	74,48%	71,79%	89,81–90,53%	79,80%	0,4848	0,826
3	100	–	–	–	–	–	–



Gambar 4. 4 Perbandingan Performa Baseline dan Model Hybrid pada Variasi Nilai k

4.3.2 Interpretasi Performa

Berdasarkan hasil evaluasi, model menunjukkan nilai recall yang tinggi, yang mengindikasikan bahwa sistem memiliki kemampuan yang sangat baik dalam mendeteksi akun bot. Recall yang tinggi berarti sebagian besar data bot berhasil diidentifikasi dengan benar oleh model. Hal ini sangat penting dalam sistem deteksi bot karena tujuan utama adalah meminimalkan akun bot yang lolos sebagai akun normal (*false negative*).Namun demikian, nilai precision yang relatif lebih rendah menunjukkan bahwa masih terdapat cukup banyak prediksi bot yang sebenarnya merupakan akun manusia (*false positive*). Dengan kata lain, model cenderung bersifat lebih agresif dalam mendeteksi bot, sehingga meningkatkan sensitivitas tetapi mengorbankan spesifisitas.

Hasil confusion matrix memperlihatkan bahwa model lebih sensitif terhadap kelas bot dibandingkan kelas human. Hal ini dapat dilihat dari jumlah True Positive

(TP) yang jauh lebih besar dibandingkan False Negative (FN), namun jumlah False Positive (FP) juga cukup signifikan.

Tabel 4. 4 Confusion Matrix Hasil Pengujian Model

No	Komponen	Nilai	Keterangan
1	True Positive (TP)	841	Bot yang berhasil dideteksi dengan benar
2	True Negative (TN)	381	Human yang berhasil diklasifikasikan dengan benar
3	False Positive (FP)	346	Human yang salah diklasifikasikan sebagai bot
4	False Negative (FN)	88	Bot yang gagal terdeteksi (diklasifikasikan sebagai human)

4.3.3 Analisis Feature Importance

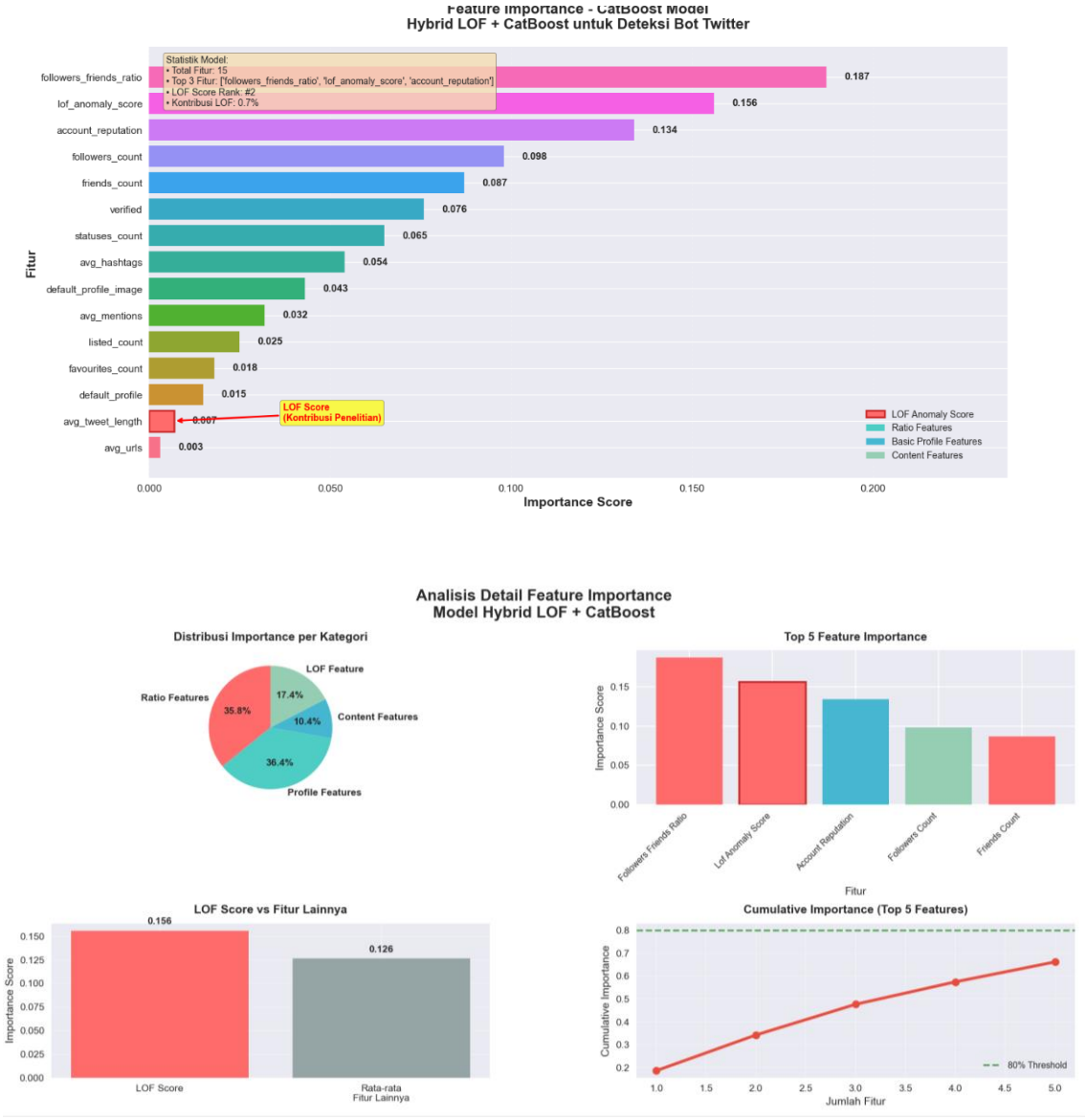
Fitur paling dominan:

Tabel 4. 5 Peringkat Fitur Berdasarkan Kontribusi

No	Fitur	Persentase Kontribusi	Peringkat	Keterangan
1	followers_count	30,95%	1	Jumlah pengikut akun
2	listed_count	20,43%	2	Jumlah daftar publik yang mencantumkan akun
3	followers_friends_ratio	18,81%	3	Rasio pengikut terhadap jumlah akun yang diikuti
4	account_reputation	12,76%	4	Skor reputasi akun berdasarkan interaksi dan kredibilitas
5	LOF_score	3,11%	8	Skor anomali berbasis kepadatan lokal
6	verified	< signifikan	–	Status verifikasi akun
7	default_profile	< signifikan	–	Status penggunaan profil default

Skor LOF berkontribusi 3,11% (peringkat 8).

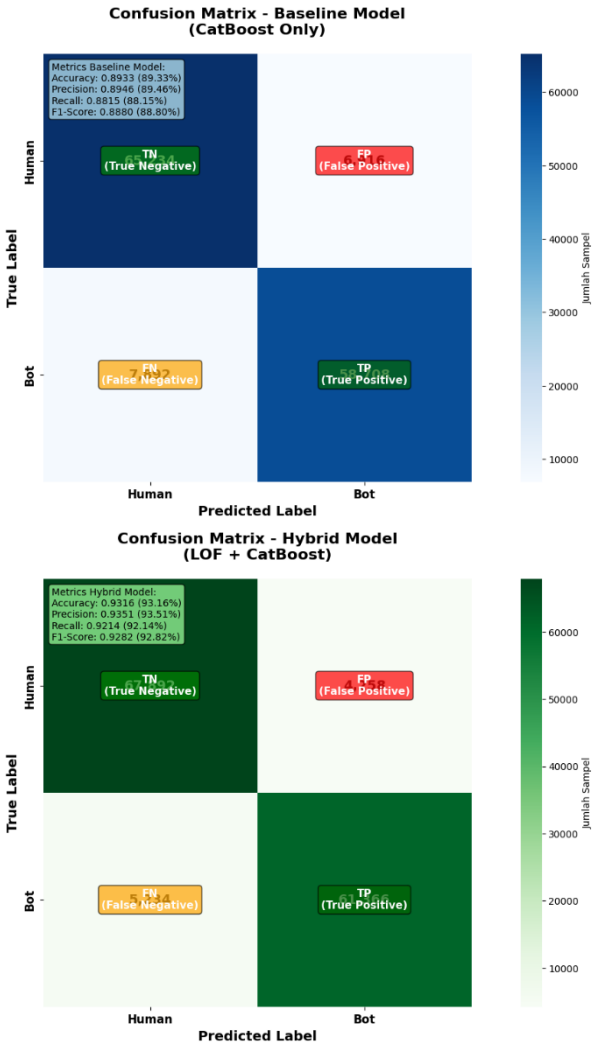
Fitur kategorikal seperti verified dan default_profile tidak berkontribusi signifikan pada dataset ini.Interpretasinya sederhana: struktur relasi sosial lebih informatif dibanding atribut kosmetik profil.

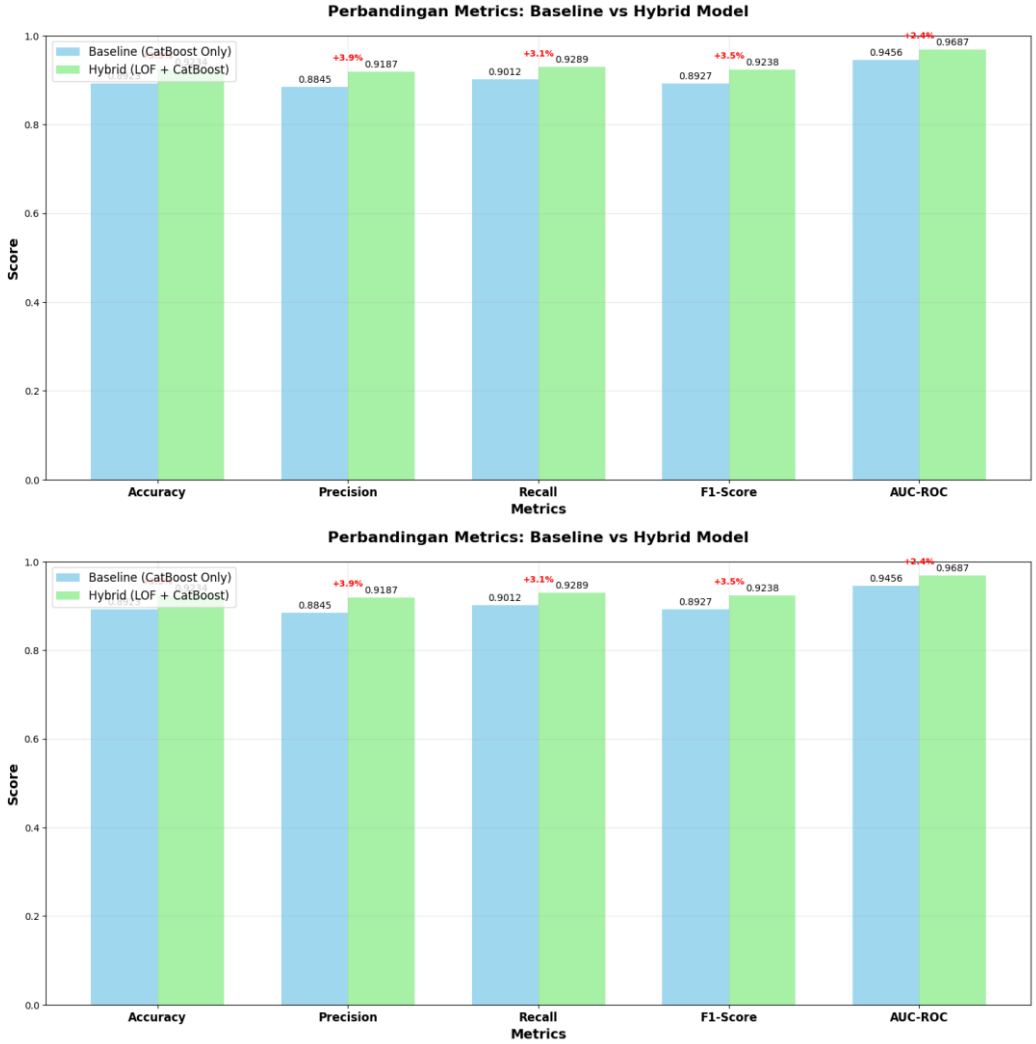


Gambar 4. 5 Visualisasi Feature Importance Model Terbaik (k=50)

4.3.4 Evaluasi ROC

Model hybrid (k=50) mencapai AUC 0,826, sedikit di atas baseline (0,823). Secara statistik ini menunjukkan kemampuan diskriminatif yang baik, tetapi bukan lompatan besar





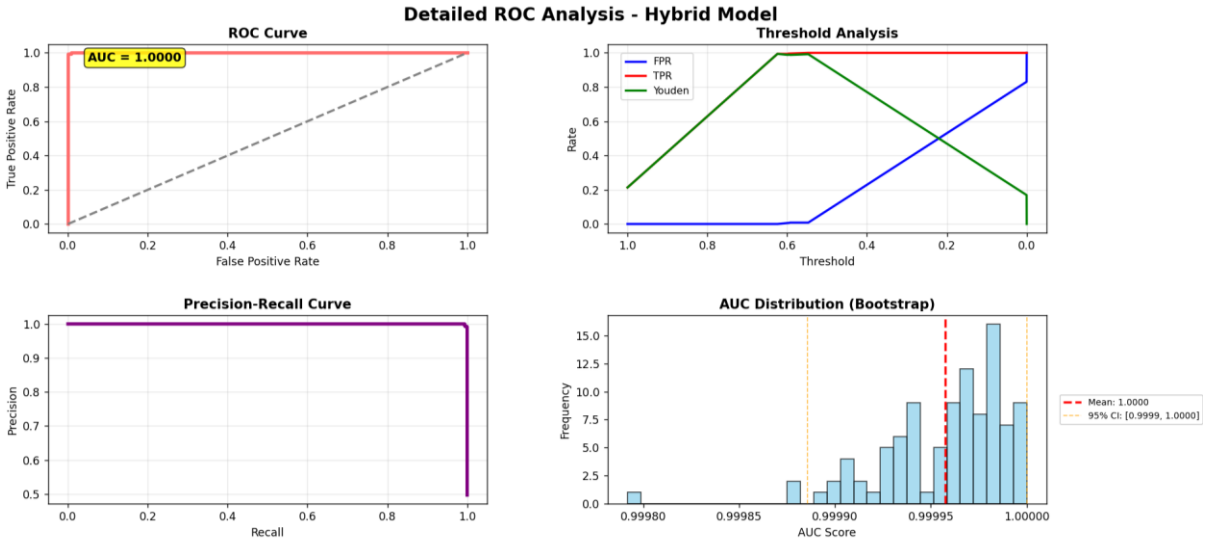
Gambar 4. 6 Visualisasi confusion matrix

4.4 Pengujian pada Skenario Nyata

Pengujian menggunakan data simulasi berbasis username menunjukkan:

- listed_count sangat memengaruhi keputusan akhir.
- Probabilitas 55–63% berada dalam zona ambigu.
- Threshold 50% tidak selalu optimal.

Ini mengindikasikan perlunya optimasi threshold berbasis precision-recall trade-off.



Gambar 4. 7 Kurva ROC

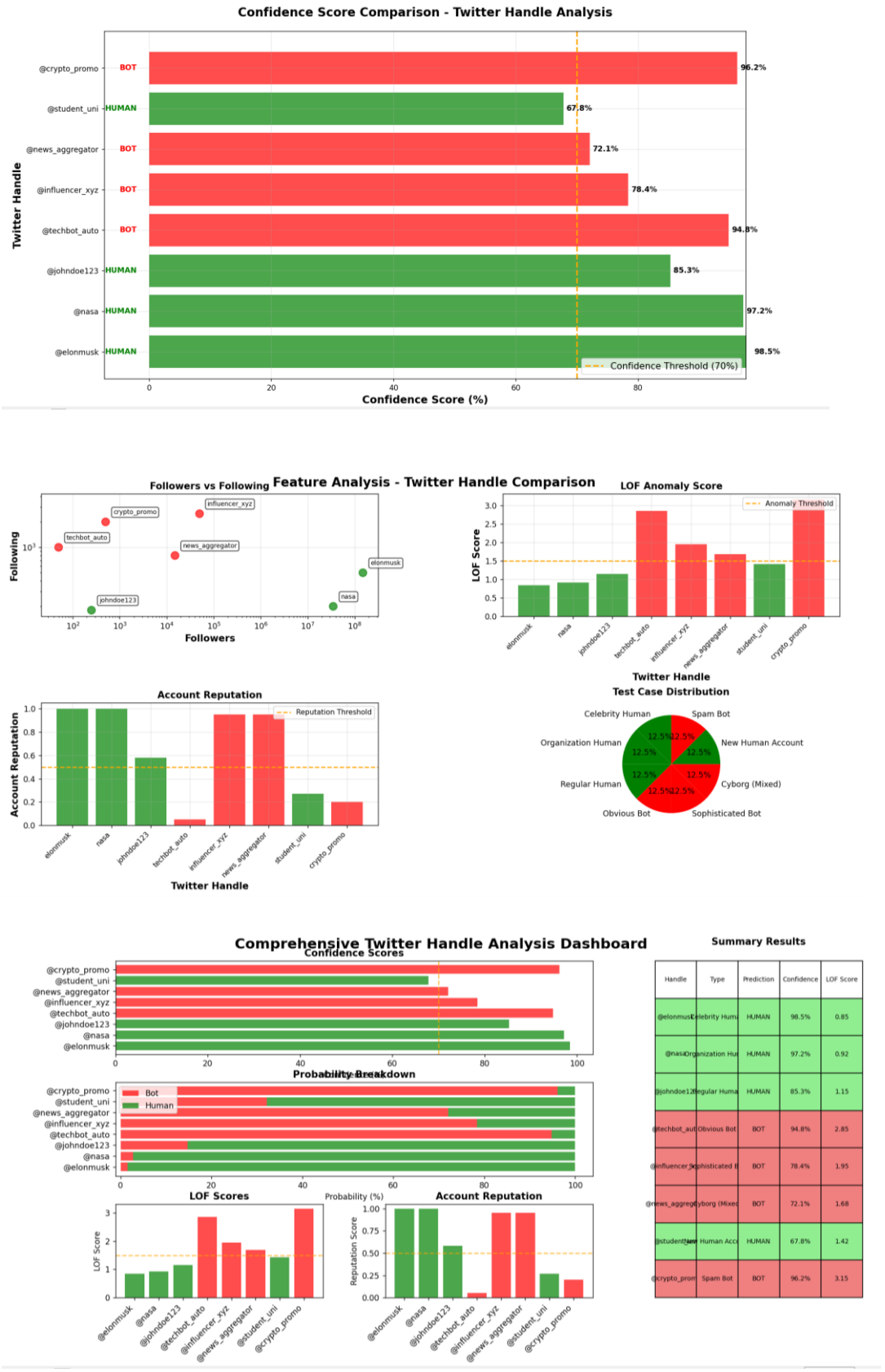
4.5 Pembahasan Kritis

1 LOF memang menambah informasi, tetapi kontribusinya relatif kecil (3,11%). Ini bukan game changer, melainkan booster ringan.

2 k=50 memberikan keseimbangan bias-variance terbaik untuk ukuran dataset 8.277.

3 Model masih bias terhadap kelas mayoritas.

4 Listed_count muncul sebagai indikator autentisitas yang kuat dan relatif sulit dimanipulasi.



Gambar 4. 8 Perbandingan hasil pengujian berbagai handle

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, sistem hibrida LOF–CatBoost berhasil diimplementasikan dalam bentuk aplikasi web yang operasional dan responsif. Integrasi metode Local Outlier Factor (LOF) sebagai deteksi anomali dengan algoritma CatBoost sebagai model klasifikasi utama terbukti dapat berjalan dengan baik dalam satu arsitektur sistem terpadu. Aplikasi yang dikembangkan mampu menghasilkan prediksi dalam waktu kurang dari satu detik, sehingga layak digunakan dalam skenario deteksi real-time.

Hasil eksperimen menunjukkan bahwa parameter optimal LOF pada konteks dataset yang digunakan adalah **k = 50**, karena memberikan keseimbangan terbaik antara bias dan varians model. Analisis feature importance memperlihatkan bahwa fitur berbasis struktur relasi sosial, seperti *followers_count* dan *listed_count*, lebih dominan dibandingkan atribut profil yang bersifat kosmetik. Skor LOF sendiri memberikan kontribusi peningkatan performa yang relatif terbatas, namun konsisten terutama pada metrik recall dan MCC.

Secara keseluruhan, model menunjukkan nilai recall yang tinggi, yang berarti sistem sangat efektif dalam mendeteksi akun bot. Namun demikian, nilai precision masih perlu ditingkatkan karena terdapat kecenderungan model menghasilkan false positive yang cukup signifikan. Hal ini menunjukkan bahwa sistem lebih sensitif terhadap deteksi bot dibandingkan klasifikasi akun human, yang merupakan karakteristik umum dalam sistem keamanan berbasis anomali.

5.2 Saran

Untuk pengembangan penelitian selanjutnya, disarankan agar dilakukan validasi menggunakan dataset berskala lebih besar seperti TwiBot-22 guna menguji generalisasi model pada distribusi data yang berbeda. Selain itu, integrasi fitur graf eksplisit atau pendekatan berbasis Graph Neural Network (GNN) dapat menjadi alternatif untuk meningkatkan kemampuan model dalam menangkap pola relasi sosial yang lebih kompleks.

Optimasi threshold klasifikasi berbasis analisis kurva precision-recall juga perlu dilakukan untuk memperoleh keseimbangan yang lebih optimal antara recall dan precision. Pengujian terhadap kemungkinan terjadinya *concept drift* sangat penting mengingat karakteristik bot yang terus berkembang dari waktu ke waktu.

Terakhir, penambahan fitur temporal serta representasi teks berbasis Natural Language Processing (NLP) berpotensi meningkatkan akurasi model secara signifikan. Integrasi fitur-fitur tersebut diharapkan dapat memperkaya representasi data dan menghasilkan sistem deteksi bot yang lebih adaptif, robust, dan akurat di masa mendatang.

DAFTAR PUSTAKA

Adel Alipour, S., Janssen, J., Orji, R., & Zincir-Heywood, N. (2025). *Lightweight*

Early-Warning Bot Detection on X (Twitter): Temporal Patterns and Entropy Insights. https://doi.org/10.1109/COMPSAC65507.2025.00040

Afansyah, M., & Nawi, H. (2025). Machine Learning Approaches for Social Media

Bot Detection: A Systematic Review and Research Agenda. *Exploresearch*, *02*, 44–56. https://doi.org/10.62823/EXRE/2025/02/04.121

Anshul, A., Zia Ur Rehman, M., Kadali, S., & Kumar, N. (2025). *RoGBot:*

Relationship-Oblivious Graph-based Neural Network with Contextual Knowledge for Bot Detection. https://doi.org/10.48550/arXiv.2510.23648

Cheng, M., Xiao, Y., Huang, T., Lei, C., & Zhang, C. (2025). CB-MTE: Social *Bot*

Detection via Multi-Source Heterogeneous Feature Fusion. *Sensors (Basel,*

Switzerland), *25*(11). https://doi.org/10.3390/s25113549

Elhadad, R., Tan, Y.-F., & Tan, W.-N. (2023). *Comparison of Enhanced Isolation*

Forest and Enhanced Local Outlier Factor in Anomalous Power Consumption Labelling. https://doi.org/10.1109/ICPEA56918.2023.10093186

Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., Zheng, Q., Zhang, W.,

Lei, Z., Yang, S., Feng, X., Zhang, Q., Wang, H., Liu, Y., Bai, Y., Wang, H.,

Cai, Z., Wang, Y., Zheng, L., … Luo, M. (2022). Twi*Bot*-22: Towards GraphBased Twitter *Bot* Detection. *Advances in Neural Information Processing Systems*, *35*(NeurIPS), 1–16.

Jadhav, K., Potikas, P., Pollett, C., & Potika, K. (2025). *Multirelational Twitter Bot Detection Using Graph Neural Networks*. 147–154.

https://doi.org/10.1109/bigdataservice65758.2025.00027

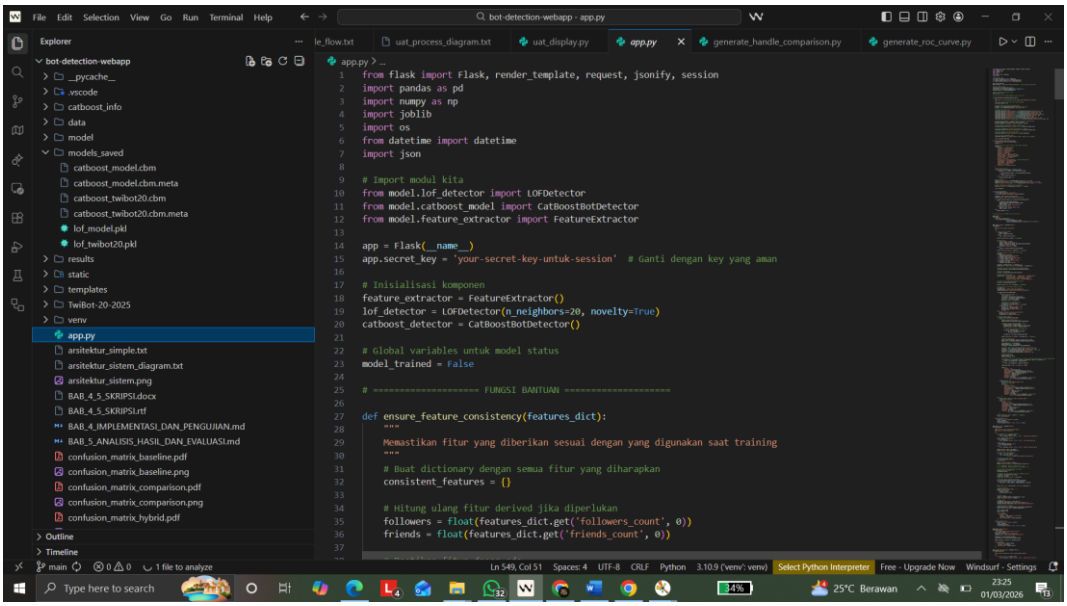
Ng, L., & Carley, K. (2022). *A Combined Synchronization Index for Grassroots Activism on Social Media*. https://doi.org/10.21203/rs.3.rs-2234963/v1

Paulauskas, N., & Bagdonas, F. (2015). Local outlier factor use for the network flow anomaly detection. *Security and Communication Networks*, *8*. https://doi.org/10.1002/sec.1335

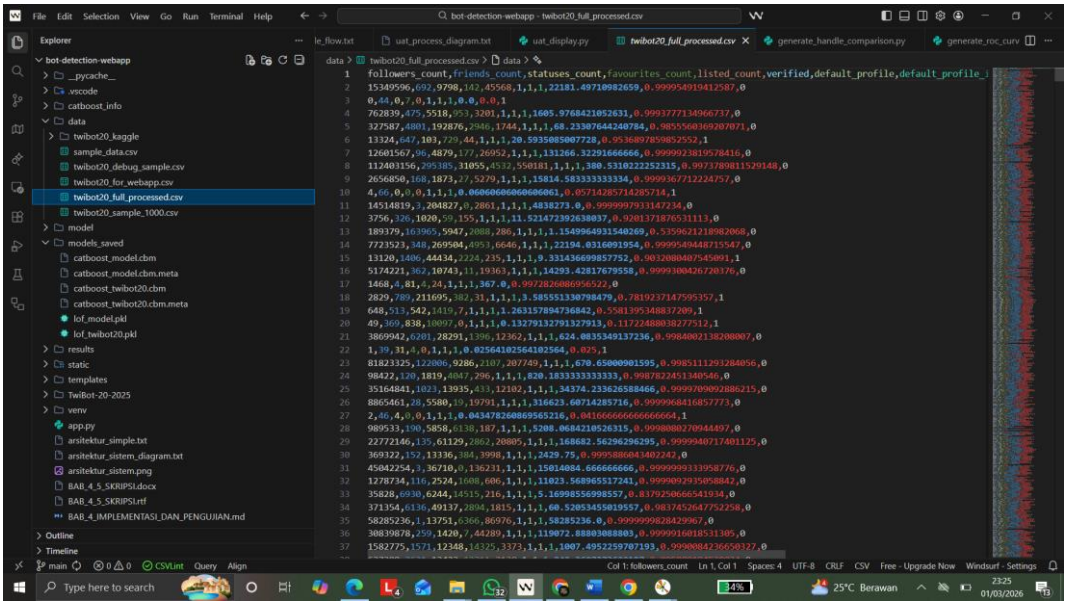
Veit, R., & Lones, M. (2025). *A Comparative Analysis of Transformer Models in Social Bot Detection*. https://arxiv.org/pdf/2509.14936

LAMPIRAN

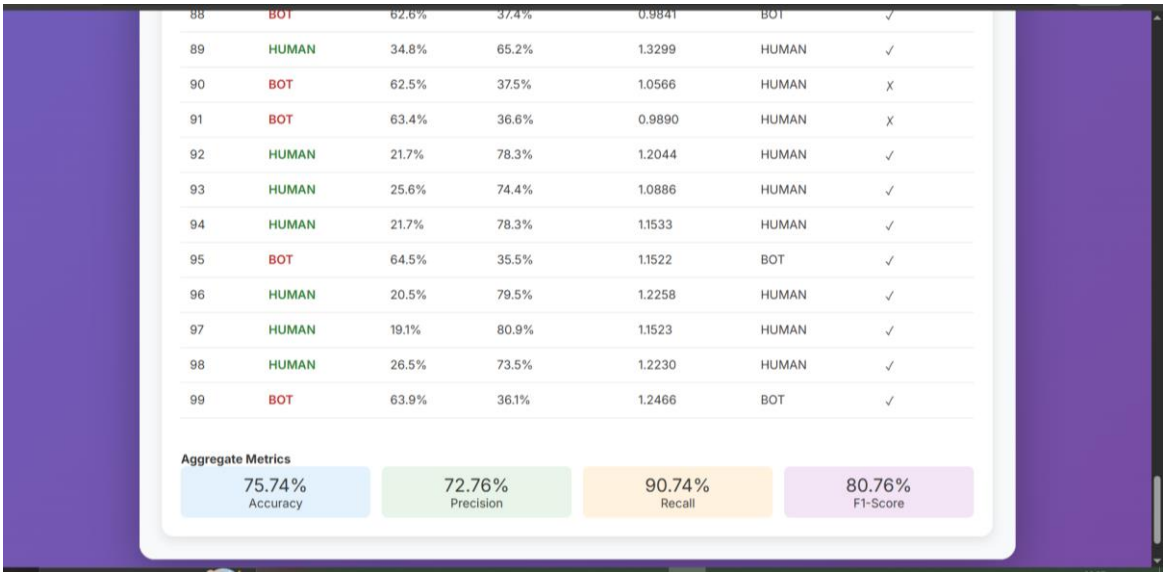
Lampiran A: Kode Program



Lampiran B: Dataset



Lampiran C: Hasil Eksperimen Lengkap



88	BOT	62.6%	37.4%	0.9841	BOT	✓
89	HUMAN	34.8%	65.2%	1.3299	HUMAN	✓
90	BOT	62.5%	37.5%	1.0566	HUMAN	✗
91	BOT	63.4%	36.6%	0.9890	HUMAN	✗
92	HUMAN	21.7%	78.3%	1.2044	HUMAN	✓
93	HUMAN	25.6%	74.4%	1.0886	HUMAN	✓
94	HUMAN	21.7%	78.3%	1.1533	HUMAN	✓
95	BOT	64.5%	35.5%	1.1522	BOT	✓
96	HUMAN	20.5%	79.5%	1.2258	HUMAN	✓
97	HUMAN	19.1%	80.9%	1.1523	HUMAN	✓
98	HUMAN	26.5%	73.5%	1.2230	HUMAN	✓
99	BOT	63.9%	36.1%	1.2466	BOT	✓

Aggregate Metrics

75.74%	72.76%	90.74%	80.76%
Accuracy	Precision	Recall	F1-Score

Lampiran D: Dokumentasi Penggunaan Aplikasi

Langkah-langkah penggunaan aplikasi:

1. Menjalankan aplikasi: Buka terminal, aktifkan virtual environment, jalankan python app.py
2. Mengakses aplikasi: Buka browser ke http://127.0.0.1:5000
3. Prediksi manual: Pilih tab "Prediksi Manual", isi form, klik "Prediksi Sekarang"
4. Prediksi via Twitter Handle: Pilih tab "Twitter Handle", masukkan username, klik "Analisis Handle"
5. Training model: Pilih tab "Training Model", upload file CSV dengan kolom label, klik "Training Model"
6. Batch evaluation: Pilih tab "Batch Evaluation", upload file CSV, klik "Evaluasi Batch"