ANALISIS PERBANDINGAN ALGORITMA CATBOOST DAN LIGHTGBM DALAM PREDIKSI *MULTI-LABEL* SERANGAN HAMA PADA TEBU

Oleh

MUTIARA CINTIA RAINY

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRAK

ANALISIS PERBANDINGAN ALGORITMA CATBOOST DAN LIGHTGBM DALAM PREDIKSI *MULTI-LABEL* SERANGAN HAMA PADA TEBU

Oleh

MUTIARA CINTIA RAINY

Penelitian ini bertujuan membangun model prediksi *multi-label* tingkat keparahan serangan hama tebu (penggerek batang, penggerek pucuk, dan kutu perisai) menggunakan algoritma CatBoost dan LightGBM. Dataset pengamatan lapangan diperkaya dengan fitur temporal berbasis *lag* (1–3) serta fitur kategorikal yang diproses secara native oleh masing-masing algoritma. Pembagian data dilakukan berbasis blok untuk mencegah *data leakage* dan merepresentasikan skenario prediksi pada blok yang belum diamati. Evaluasi menggunakan *F1-score macro average* dan AUC *macro* pada tiga skema pembagian data menunjukkan bahwa kedua model memiliki performa yang relatif sebanding, dengan *F1-score* pada kisaran 0,59–0,69 dan AUC *macro* 0,80–0,86. Pada tingkat kelas, kemampuan deteksi terhadap kelas keparahan tinggi konsisten dengan AUC di atas 0,90. LightGBM unggul dalam waktu pelatihan, sedangkan CatBoost lebih stabil dan memiliki waktu prediksi lebih cepat. Model yang direkomendasikan diharapkan mendukung sistem pemantauan risiko serangan hama secara dini dan pengambilan keputusan yang lebih terarah.

Kata kunci: Klasifikasi *Multi-label*, LightGBM, CatBoost, Serangan Hama Tebu.

ABSTRACT

COMPARATIVE ANALYSIS OF CATBOOST AND LIGHTGBM ALGORITHMS FOR MULTI-LABEL PREDICTION OF PEST INFESTATION IN SUGARCANE

By

MUTIARA CINTIA RAINY

This study aims to develop a multi-label prediction model for the severity levels of sugarcane pest infestations (stem borer, top borer, and scale insects) using the CatBoost and LightGBM algorithms. Field observation data were enriched with temporal lag-based features (1–3 periods) and categorical features processed natively by each algorithm. Data splitting was performed at the block level to prevent data leakage and to represent prediction scenarios for previously unseen blocks. Evaluation using macro-averaged F1-score and macro AUC across three data-splitting schemes indicates that both models achieve comparable performance, with F1-scores ranging from 0,59 to 0,69 and macro AUC between 0,80 and 0,86. At the class level, detection performance for the high-severity class is consistently strong, with AUC values above 0,90. LightGBM demonstrates superior training speed, while CatBoost shows greater stability and faster inference time. The recommended model is expected to support early pest risk monitoring systems and enable more targeted decision-making in sugarcane cultivation.

Keywords: Multi-label Classification, LightGBM, CatBoost, Sugarcane Pest Infestation.

Judul Skripsi	:	**ANALISIS PERBANDINGAN ALGORITMA CATBOOST DAN LIGHTGBM DALAM PREDIKSI *MULTI-LABEL* SERANGAN HAMA PADA TEBU**
Nama Mahasiswa	:	Mutiara Cintia Rainy
NPM	:	2217051100
Program Studi	:	S1 Ilmu Komputer
Jurusan	:	Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Dewi Asiah Shofiana, S.Komp., M.Kom.	**Ridho Sholehurrohman, M.Mat.**
NIP. 199509292020122030	NIK. 232111970128101

2. Mengetahui

Ketua Jurusan Ilmu Komputer	Ketua Program Studi Ilmu Komputer
Dwi Sakethi, S.Si., M.Kom.	**Tristiyanto, S.Kom., M.I.S., Ph.D**
NIP. 196806111998021001	NIK. 198104142005011001

MENGESAHKAN

1. Tim Penguji

Ketua Penguji : **Dewi Asiah Shofiana, S.Komp., M.Kom.** .....

Sekretaris Penguji : **Ridho Sholehurrohman, M.Mat.** .....

Penguji Utama : **Tristiyanto, S.Kom., M.I.S., Ph.D** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 9 Maret 2026

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mutiara Cintia Rainy

NPM : 2217051100

Dengan ini menyatakan bahwa skripsi saya yang berjudul **"Analisis Perbandingan Algoritma Catboost Dan Lightgbm Dalam Prediksi *Multi-Label* Serangan Hama Pada Tebu"** merupakan karya saya sendiri, bukan karya orang lain. Semua tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa karya tulis ilmiah saya terbukti hasil menjiplak karya orang lain, maka saya siap menerima sanksi berupa pencabutan gelar yang saya peroleh.

Bandar Lampung, 9 April 2026

Mutiara Cintia Rainy
NPM. 2217051100

RIWAYAT HIDUP

Lahir pada 5 Januari 2004. Anak kedua dari Bapak Wahyu Rihadhi dan Ibu Sri Supadmi. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SD Negeri 2 Sukadana Pasar pada Tahun 2015, lalu pendidikan menengah pertama di SMP Negeri 1 Purbolinggo Lampung Timur dan lulus pada Tahun 2018. Kemudian melanjutkan pendidikan menengah atas di SMA Negeri 1 Purbolinggo Lampung Timur yang diselesaikan pada Tahun 2021. Pada tahun 2022 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa penulis melakukan beberapa kegiatan antara lain.

1. Berpartisipasi dalam lomba Gemastik (Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi) Divisi UI/UX pada Tahun 2023 dan Tahun 2024.
2. Menjadi Asisten Dosen Jurusan Ilmu Komputer mata kuliah Komunikasi Data dan Jaringan Komputer pada Tahun 2024, dan mata kuliah Kecerdasan Buatan pada Tahun 2025.
3. Mengikuti program Magang dan Studi Independen Bersertifikat (MSIB) di MIKTI *Learning Path Data Analyst with Artificial Intelligence* (AI) *Expertise* pada Tahun 2024.
4. Melaksanakan Kerja Praktik (KP) di Badan Perencanaan Pembangunan Daerah Kabupaten Lampung Timur pada Desember 2024 hingga Februari 2025.
5. Melaksanakan Kuliah Kerja Nyata (KKN) di Desa Nusantara Permai, Kecamatan Sukabumi, Kota Bandar Lampung pada Juli – Agustus 2025.

MOTTO

”Setiap rasa sakit memberi kita pelajaran dan setiap pelajaran mengubah seseorang. Kalau hanya fokus ke rasa sakit, maka kamu akan terus menderita, tapi jika kamu fokus dengan pelajarannya kamu akan terus tumbuh.”
(Ust. Hanan Attaki)

”Lumut” [Karya Yuan Mei]
Di tempat yang tidak terjangkau sinar matahari,
Kehidupan muncul dengan sendirinya.
Bunga lumut itu sekecil biji beras,
Namun ia belajar mekar layaknya bunga Peony.
(Dalam Film: *Big World* - 2024)

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji syukur kehadirat Allah Subhanahu Wa Ta'ala atas segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan sebaik-baiknya. Shalawat serta salam selalu tercurahkan kepada junjungan Nabi Agung Muhammad Shallallahu 'Alaihi Wasallam.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Yang selalu mendukung, memberikan cinta dan kasih sayang yang tak terhingga, serta do'a yang selalu menyertaiku. Kuucapkan terima kasih sebesar-besarnya atas pengorbanan dan perjuangan dalam mendidik dan membesarkanku yang tak akan dapat terbalaskan.

Seluruh Keluarga Besar Ilmu Komputer 2022

Yang senantiasa memberikan semangat dan dukungan.

Almamater Tercinta, Jurusan Ilmu Komputer FMIPA Universitas Lampung

Tempat bernaung mengemban semua ilmu untuk menjadi bekal hidup.

SANWANCANA

Puji syukur kehadirat Allah Subhanahu Wa Ta'ala atas limpahan nikmat, rahmat dan karunia-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Muhammad Shallallahu 'Alaihi Wasallam, sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Analisis Perbandingan Algoritma Catboost Dan Lightgbm Dalam Prediksi *Multi-Label* Serangan Hama Pada Tebu"** dengan baik dan lancar.

Terima kasih penulis ucapkan kepada pihak-pihak yang telah memberi dukungan, bimbingan dan membantu penulis dalam menyelesaikan penyusunan skripsi ini. Ucapan terima kasih ini penulis tujukan kepada:

1. Allah SWT yang telah memberikan hidayah kesehatan dan kemampuan untuk menyelesaikan skripsi ini.
2. Kedua orang tua penulis, Bapak Wahyu Rihadhi dan Ibu Sri Supadmi yang tidak henti-hentinya memanjatkan doa serta selalu memberikan dukungan instrumental, emosional, dan informasional.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung dan Pembimbing Akademik penulis yang selalu mendukung peningkatan akademik penulis.
6. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Lampung dan Pembahas yang telah memberikan masukan serta saran yang bermanfaat untuk perbaikan skripsi ini.

7. Dewi Asiah Shofiana, S.Komp., M.Kom. selaku Pembimbing Utama yang telah memberikan arahan, ide, kritik, serta saran dalam menyelesaikan penelitian ini.
8. Bapak Ridho Sholehurrohman, M. Mat. selaku Pembimbing Kedua yang selalu dapat meluangkan waktunya untuk membimbing, memberikan arahan dan bantuan ketika mengalami kesulitan dalam menyelesaikan penelitian ini.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu, motivasi dan pengalaman hidup selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
10. Ibu Ade Nora Maela dan seluruh staf di Jurusan Ilmu Komputer yang telah membantu segala urusan administrasi di Jurusan Ilmu Komputer.
11. Kakak penulis Gerry Wednes Argipala yang sudah menjadi panutan penulis, serta adik penulis Intan Delisha Cesarindy yang telah memberi penghiburan.
12. Sahabat penulis Adinda, Ica, Tasya, Isma dan Salsa. Terima kasih telah menemani sejak masa SMA hingga sekarang.
13. Teman-teman grup "*strict parents*", Tamara, Suci dan Tata. Terima kasih karena telah mengisi hari-hari di tahun pertama perkuliahan.
14. Teman-teman KKN, Sintia, Opi, Dinda, Nisa, Rido, Doni dan Heber. Terima kasih atas kesan dan cerita seru yang kalian berikan semasa KKN dan setelahnya.
15. Keluarga Ilmu Komputer 2022 yang telah memberikan pengalaman berharga. Terima kasih telah menjadi rekan kelompok, rekan diskusi, dan rekan berjuang selama menjalankan studi di Jurusan Ilmu Komputer Universitas Lampung.
16. Seluruh pihak yang telah membantu dan memberikan dukungan, baik secara langsung maupun tidak langsung selama perkuliahan hingga penyelesaian skripsi ini.
17. Diri saya sendiri, Mutiara Cintia Rainy. Apresiasi sebesar-besarnya atas segala kerja keras dan semangatnya, sehingga tidak pernah memutuskan untuk menyerah, sesulit apapun penyusunan skripsi ini. Terima kasih karena

telah mampu mengendalikan diri dari berbagai tekanan luar dan bertahan sampai sejauh ini, serta bertanggung jawab untuk menyelesaikan apa yang telah dimulai dan senantiasa menikmati setiap prosesnya.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Namun, penulis sangat mengharapkan skripsi ini dapat bermanfaat bagi para civitas akademik Universitas Lampung, khususnya mahasiswa Ilmu Komputer.

Bandar Lampung, 9 April 2026

Mutiara Cintia Rainy

NPM. 2217051100

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

Gambar Halaman

DAFTAR LAMPIRAN

I. PENDAHULUAN

1.1. Latar Belakang

Perkebunan merupakan sektor penting dalam perekonomian Indonesia karena berperan sebagai penyedia bahan baku industri dan sumber penghidupan bagi masyarakat. Salah satu komoditas strategis adalah tebu (*Saccharum officinarum L.*), yang menjadi bahan utama produksi gula. Hal ini didukung oleh data dari Badan Pusat Statistik Indonesia (BPS) bahwa luas areal perkebunan tebu terus bertambah setiap tahun, dan pada tahun 2024 luasnya tercatat sekitar 520,82 ribu hektar. Pentingnya komoditas ini juga dipertegas oleh Kementerian Perindustrian, yang menyatakan bahwa industri gula merupakan salah satu sektor strategis yang memiliki peran vital bagi upaya ketahanan pangan dan peningkatan pertumbuhan perekonomian nasional (Nugroho, 2021). Namun peningkatan luas areal perkebunan tebu tidak otomatis meningkatkan jumlah produksi gula. Data BPS Indonesia (2023) menunjukkan bahwa produksi gula Indonesia cenderung fluktuatif, menurun dari 2,58 juta ton pada tahun 2014 hingga mencapai titik terendah 2,13 juta ton pada tahun 2020, sebelum kembali meningkat mendekati 2,48 juta ton pada tahun 2024. Variabilitas produksi ini mengindikasi adanya faktor pembatas penting dalam sistem budidaya tebu yang perlu dipahami secara lebih mendalam. Salah satu faktor risiko yang paling sering dikaitkan dengan penurunan produktivitas adalah serangan hama.

Berbagai penelitian di Indonesia mendukung temuan tersebut. Survei yang dilakukan di Provinsi Jambi menemukan bahwa hama penggerek batang jenis *Rhabdoscelus obscurus* merupakan hama utama yang menyerang beberapa lokasi perkebunan tebu, dengan gejala berupa batang berlubang, tumbang, hingga kematian tanaman (Adrian *et al.*, 2019). Penelitian di PT PG Rajawali II Jatitujuh

melaporkan insidensi hama seperti penggerek batang *Chilo sacchariphagus* dengan tingkat kejadian 1,26 %, serta penggerek pucuk *Scirpophaga excerptalis* dengan tingkat kejadian 0,86 %, yang berkaitan dengan penurunan produktivitas dari 6,27 ton per hektar pada 2016 menjadi 3,94 ton per hektar pada 2018 (Muliasari & Trilaksono, 2020). Studi lain di Kabupaten Sleman menemukan bahwa serangan hama uret memiliki hubungan kuat dengan penurunan rendemen, tercermin dari nilai koefisien determinasi (R^2) sebesar 0,79, selain itu serangan uret juga menunjukkan korelasi kuat dengan curah (Utami *et al.*, 2024). Temuan tersebut sejalan dengan kajian Pramono (2025), yang menunjukkan bahwa intensitas serangan penggerek batang cenderung meningkat pada fase vegetatif–generatif awal, terutama ketika kelembapan tanah tinggi dan pola rotasi tanaman tidak optimal. Jika ditarik bersama, berbagai hasil penelitian ini mengindikasikan bahwa dinamika serangan hama tebu merupakan hasil interaksi kompleks antara spesies hama, fase pertumbuhan tanaman, serta kondisi lingkungan yang berubah sepanjang musim.

Di sisi lain, kondisi lapangan menunjukkan bahwa pola serangan hama pada tanaman tebu cenderung kompleks. Dalam satu blok lahan, beberapa jenis hama seperti penggerek batang, penggerek pucuk, dan kutu perisai dapat muncul secara bersamaan, dengan tingkat keparahan yang berubah mengikuti fase pertumbuhan tanaman dan dinamika agroklimat. Fenomena multi-hama ini tidak hanya mencerminkan dinamika biologis tanaman tebu, tetapi juga menimbulkan tantangan bagi Perusahaan Gula A yang mulai mengembangkan sistem prediksi untuk mendukung pengambilan keputusan pengendalian hama. Berdasarkan informasi teknis dari pihak perusahaan, sistem prediksi yang digunakan saat ini masih berfokus pada pendekatan *multi-class* untuk satu jenis hama dalam satu waktu serta mengandalkan model bawaan perangkat analitik internal. Pendekatan tersebut belum mampu merepresentasikan kombinasi serangan yang terjadi secara simultan dan belum efektif dalam menangkap pola hubungan nonlinier antara variabel agroklimat dan perkembangan tanaman. Keterbatasan ini menunjukkan adanya kebutuhan akan pendekatan analitik yang mampu memodelkan serangan

dari beberapa jenis hama pada periode yang sama agar hasil prediksi lebih mencerminkan kondisi lapangan.

Sejalan dengan tantangan tersebut, perkembangan pertanian modern menunjukkan pergeseran menuju pemanfaatan *early warning system* berbasis data untuk memprediksi risiko produksi. Wadhwa & Malik (2024) menekankan bahwa data agroklimat, dan model pembelajaran mesin menjadi fondasi utama dalam mendeteksi potensi gangguan tanaman sejak dini, sehingga tindakan mitigasi dapat dilakukan lebih cepat dan efisien. Sementara itu, Gidiglo *et al.* (2024) menunjukkan bahwa model berbasis *machine learning* mampu menangkap pola-pola nonlinier kompleks dalam sistem produksi tanaman dan menghasilkan prediksi risiko yang lebih akurat dibandingkan metode tradisional.

Beberapa hasil penelitian sebelumnya memperlihatkan bahwa model prediksi hama pada tebu umumnya masih berfokus pada satu jenis serangan dalam satu waktu. Sistem prediksi yang dikembangkan Pituckwanich *et al.* (2025) misalnya, hanya memodelkan tingkat kerusakan akibat *stem borer* menggunakan pendekatan *hybrid machine learning*, sehingga belum mencakup keterkaitan antarjenis hama maupun variasi serangan yang muncul secara bersamaan. Pendekatan serupa juga terlihat pada penelitian Nadeem *et al.* (2022), yang mengembangkan sistem *early warning* berbasis algoritma *Naive Bayes* untuk memprediksi serangan *stem borer* pada tebu berdasarkan suhu, kelembapan, dan curah hujan. Meskipun sistem tersebut menunjukkan tingkat akurasi yang cukup baik, yaitu sekitar 83% pada pengujian model serta 77% hingga 91% pada evaluasi lapangan tahunan, pemodelan yang dilakukan tetap bersifat *single-label* atau berfokus pada satu jenis hama. Temuan serupa juga terlihat pada studi peringatan dini penyakit tanaman yang memanfaatkan data lingkungan, namun masih menggunakan pendekatan *single-label* (Wadhwa & Malik, 2024). Kondisi ini menunjukkan bahwa sebagian besar penelitian terdahulu belum mengakomodasi kompleksitas fenomena multi-hama pada tebu, khususnya ketika beberapa jenis hama muncul secara simultan dan dipengaruhi oleh dinamika lingkungan serta fase pertumbuhan tanaman yang berbeda.

Pendekatan *multi-label* pada konteks pertanian sebenarnya telah terbukti efektif di komoditas lain. Studi Garba *et al.* (2025) memodelkan beberapa hama utama pada serealia di Niger seperti *fall armyworm*, penggerek batang lokal, dan *aphids* yang muncul secara bersamaan pada fase fenologi tanaman tertentu. Pola kemunculan serempak tersebut menyebabkan label hama saling bergantung, baik melalui respons yang sama terhadap variabel agroklimat maupun interaksi biologis antarhama. Secara teoretis, prediksi pada kondisi multi-hama membutuhkan kerangka pemodelan yang mampu menangkap lebih dari satu label secara bersamaan. Zhang & Zhou (2013) menegaskan bahwa *multi-label learning* dirancang untuk merepresentasikan dependensi antarlabel serta hubungan laten yang dipengaruhi faktor lingkungan, menjadikannya lebih tepat untuk fenomena yang bersifat simultan. Selain itu, sistem prediksi yang diterapkan pada data pertanian idealnya mampu mengenali pola nonlinier, menangani ketidakseimbangan distribusi kelas, serta tetap stabil pada data yang dipengaruhi variasi musiman (Simeone, 2018; Younes, 2024).

Dengan karakteristik permasalahan *multi-label* tersebut, strategi pemodelan tidak hanya bergantung pada kerangka klasifikasi, tetapi juga pada pemilihan algoritma pembelajaran mesin yang sesuai dengan sifat data. Data prediksi serangan hama umumnya memiliki ketidakseimbangan antarlabel, jumlah fitur yang relatif besar, serta variasi temporal yang dipengaruhi faktor musiman. Kompleksitas ini menuntut algoritma yang mampu memodelkan hubungan nonlinier dan interaksi antarvariabel secara fleksibel. LightGBM (*Light Gradient Boosting Machine*) dan CatBoost (*Category Boosting*) merupakan algoritma berbasis *gradient boosting*, yaitu pendekatan pembelajaran yang membangun model secara bertahap melalui penggabungan sejumlah pohon keputusan (Friedman, 2001). Kedua algoritma tersebut dikenal efektif dalam menangani data berskala besar, distribusi kelas yang tidak seimbang, serta struktur data kompleks, sehingga relevan untuk digunakan pada permasalahan prediksi serangan hama dengan skema *multi-label* (Ke *et al.*, 2017; Prokhorenkova *et al.*, 2018).

Keunggulan CatBoost sebagai metode *boosting* modern ditunjukkan oleh Mahesh & Soundrapandiyan (2024), yang melaporkan nilai akurasi prediksi mencapai 99,1% pada prediksi hasil panen berbasis variabel lingkungan dan penggunaan pestisida. Keunggulan serupa juga terlihat pada penelitian prediksi kelembapan udara berbasis data meteorologi harian yang mencakup suhu, curah hujan, durasi penyinaran, dan karakteristik angin. Dalam studi tersebut, CatBoost mencapai nilai *Mean Absolute Error* (MAE) sebesar 0,0570, sementara LightGBM pada *dataset* yang sama memperoleh nilai MAE sebesar 0,0617 dengan kesalahan prediksi yang lebih tinggi (Wibawa *et al.*, 2025). Hasil ini mengindikasikan bahwa CatBoost memiliki keunggulan dalam menangani data dengan pola temporal dan variasi musiman yang kuat.

Sementara itu, LightGBM juga menunjukkan keunggulan dalam menangani data lingkungan berskala besar dan berdimensi tinggi. Huang *et al.* (2024) melaporkan bahwa LightGBM mencapai nilai *Area Under the Curve* (AUC) hingga 0,97 dalam pemodelan distribusi potensial hama. Nilai AUC yang tinggi menunjukkan kemampuan model dalam membedakan kondisi berisiko dan tidak berisiko secara konsisten. Keunggulan LightGBM dalam menangani data pertanian yang heterogen juga diperkuat oleh Kumar *et al.* (2024), yang melaporkan akurasi prediksi sebesar 94,7% pada data dengan distribusi tidak seragam dan hubungan nonlinier antarfitur. Kinerja tersebut menunjukkan kemampuan LightGBM dalam mengekstraksi informasi penting dari kombinasi variabel lingkungan yang saling berinteraksi. Karakteristik ini sejalan dengan kebutuhan pemodelan prediksi serangan hama *multi-label*, di mana kemunculan beberapa jenis hama dipengaruhi oleh faktor agroklimat, kondisi lahan, dan dinamika musiman.

Ketahanan algoritma *boosting* terhadap data tidak lengkap juga ditunjukkan dalam penelitian Rezk *et al.* (2025) yang menggunakan data sensor pertanian dengan tingkat *missing values* yang tinggi. Dalam studi tersebut, CatBoost dan LightGBM masing-masing mencapai akurasi sebesar 90,50% dan 90,23%. Berdasarkan temuan-temuan tersebut menunjukkan bahwa LightGBM dan CatBoost memiliki karakteristik yang sesuai untuk memodelkan sistem prediksi serangan hama *multi-*

label, terutama pada data yang bersifat tidak seimbang, berdimensi tinggi, dan memiliki pola temporal.

Berdasarkan uraian tersebut, penelitian ini difokuskan pada pengembangan model prediksi serangan hama tebu dengan pendekatan *multi-label*, sehingga mampu merepresentasikan kemunculan beberapa jenis hama secara bersamaan. Berbeda dari pendekatan *single-label* yang memodelkan satu jenis serangan secara terpisah, pendekatan ini diharapkan dapat menangkap keterkaitan antarjenis hama serta pengaruh bersama variabel agroklimat dan dinamika musiman pada tingkat blok lahan. Pemilihan algoritma *boosting* seperti LightGBM dan CatBoost didasarkan pada kemampuan keduanya dalam menangani data yang tidak seimbang, memiliki pola temporal, serta mengandung ketidaksempurnaan pengamatan, yang merupakan karakteristik umum pada data serangan hama.

Urgensi penelitian ini semakin kuat mengingat kebutuhan industri gula terhadap sistem peringatan dini yang lebih akurat dan representatif untuk mendukung pengambilan keputusan pengendalian hama. Dengan memodelkan serangan beberapa jenis hama secara bersamaan, penelitian ini diharapkan dapat memberikan kontribusi metodologis dalam penerapan pembelajaran mesin *multi-label* di sektor pertanian, sekaligus kontribusi praktis berupa dasar pengembangan sistem prediksi yang lebih adaptif terhadap kompleksitas kondisi lapangan. Pendekatan yang diusulkan diharapkan mampu menjembatani kesenjangan antara fenomena multi-hama yang terjadi secara nyata dan sistem prediksi yang selama ini masih bersifat parsial, sehingga relevan untuk mendukung pengelolaan hama tebu secara lebih efektif dan berbasis data.

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah bagaimana membangun model prediksi tingkat keparahan serangan hama pada tanaman tebu dengan pendekatan *multi-label* berbasis data historis, cuaca, dan

kondisi agronomis, serta bagaimana kinerja algoritma CatBoost dan LightGBM dalam memprediksi tingkat keparahan serangan hama tersebut.

1.3. Batasan Masalah

Pada penelitian ini ditetapkan beberapa batasan masalah, antara lain:

1. Kumpulan data yang digunakan dalam penelitian ini mencakup data serangan hama pada tanaman tebu tahun 2022 hingga 2023.
2. Jenis hama yang diteliti dibatasi pada tiga kategori, yaitu penggerek batang, penggerek pucuk, dan kutu perisai.
3. Model yang dikembangkan hanya menggunakan algoritma LightGBM dan CatBoost dengan pendekatan *multi-label classification*.
4. Penelitian ini tidak membahas secara mendalam aspek agronomis, biologis hama, maupun efektivitas metode pengendalian di lapangan.

1.4. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun model prediksi *multi-label* tingkat keparahan serangan hama tebu dengan algoritma Catboost dan LightGBM.
2. Mengevaluasi performa model Catboost dan LightGBM dalam beberapa skema pembagian data menggunakan *Confusion Matrix*, dan *Receiver Operating Characteristic* (ROC) *curve*.
3. Membandingkan kedua algoritma berdasarkan evaluasi performa, efisiensi waktu komputasi, aspek arsitektur, serta konfigurasi model.
4. Menentukan algoritma dengan performa terbaik dan memberikan rekomendasi model yang optimal bagi Perusahaan Gula A dalam mendukung manajemen hama.

1.5. Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi pada pengembangan penerapan metode *machine learning* berbasis *gradient boosting*, khususnya algoritma CatBoost dan LightGBM, dalam permasalahan prediksi pada sektor pertanian.
2. Memberikan gambaran empiris mengenai performa dan karakteristik CatBoost dan LightGBM pada data pertanian yang memiliki kombinasi fitur numerik, kategorikal, serta komponen temporal.
3. Memberikan rekomendasi model prediksi tingkat keparahan serangan hama tebu (*stem borer*, *top borer*, dan kutu perisai) yang dapat digunakan sebagai dasar pengembangan sistem pendukung keputusan pada Perusahaan Gula A.
4. Mendukung pengambilan keputusan berbasis data dalam manajemen perkebunan tebu, khususnya dalam menentukan prioritas *monitoring*, pengendalian hama, serta alokasi sumber daya pengamatan di lapangan.
5. Membantu perusahaan mengidentifikasi potensi risiko serangan hama lebih dini, sehingga tindakan pengendalian dapat dilakukan lebih tepat waktu dan tepat sasaran.
6. Memberikan dasar awal bagi pengembangan sistem prediksi operasional yang dapat terintegrasi dengan proses *monitoring* hama di lingkungan perkebunan.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Bagian ini menyajikan rangkuman beberapa penelitian terdahulu yang relevan dengan topik prediksi serangan hama pada tanaman tebu serta penggunaan algoritma *boosting* modern, khususnya CatBoost dan LightGBM, dalam permasalahan klasifikasi yang kompleks. Pemilihan penelitian terdahulu ini bertujuan untuk memberikan gambaran mengenai perkembangan metode prediksi di bidang pertanian, khususnya yang melibatkan data lingkungan, serangan hama, dan pendekatan *multi-label*. Ringkasan penelitian tersebut disajikan pada Tabel 1.

Tabel 1. Penelitian terdahulu.

No.	Penulis	Judul	Metode	Hasil
1.	Pituckwanich *et al.* (2025)	*The Implementation of a Prediction System for Sugarcane's Destruction rate From Sugarcane Stem borer via Hybrid Machine Learning*	- *Random forest* - *Long Short Term Memory* (LSTM) - PatchTST - XGBoost - LightGBM	Penelitian menggunakan data kerusakan tebu akibat *stem borer* dari kebun tebu Thailand. Tujuannya memprediksi tingkat kerusakan secara akurat. Model LightGBM (*Mean Absolute Error* 0,205) dan XGBoost (0,124) lebih baik daripada *Random Forest* dan LSTM. Memberi bukti bahwa *boosting* efektif untuk prediksi hama tebu.

Tabel 1 (Lanjutan)

No.	Penulis	Judul	Metode	Hasil
2.	Silva *et al.* (2024)	*Boosting algorithms for prediction in agriculture: An application of Feature Importance and Feature Selection*	- LightGBM - XGBoost - CatBoost - *Gradient Boosting* - AdaBoost - BORUTA untuk *feature selection* - SHAP *untuk interpretasi.*	Menggunakan *dataset* pertanian multivariat (cuaca, tanah, kondisi tanaman). Tujuannya mengevaluasi algoritma *boosting* dalam prediksi kesehatan tanaman. Hasil evaluasi LightGBM dan CatBoost mendapat akurasi (0,85) dengan nilai *Area Under the curve* (AUC) 0,81. Mendukung penggunaan keduanya untuk data pertanian.
3.	Kumar *et al.* (2024)	*Light Gradient Boosting Machine for Optimization and Forecasting in Agriculture*	- *Random forest* - *Support Vector Machine* - XGBoost - LightGBM	Menggunakan data hasil panen dan perawatan tanaman dari lahan pertanian skala besar. Tujuannya mengoptimalkan prediksi persentase keuntungan dan keputusan perawatan. LightGBM mencapai akurasi 94,7%, menunjukkan performa tinggi pada data tabular pertanian dan efisien digunakan untuk prediksi berbasis lingkungan.
4.	Wadhwa & Malik (2024)	*A Generalizeble Model For Early Warning Of Crop Diseases Using Environmental*	- *Random forest* - *Balanced* RF - XGBoost - CatBoost - SMOTE-ENN *(Resampling)*	Memakai data lingkungan (cuaca, kelembapan, suhu) dan tingkat infestasi hama/*disease*. Tujuannya membuat sistem peringatan dini penyakit tanaman.

Tabel 1 (Lanjutan)

No.	Penulis	Judul	Metode	Hasil
		And Pest Infestation Data	- Optuna *(Hyperparameter tuning)*	CatBoost memberikan performa terbaik (AUC 0,99 ; F1 0,94) dan bekerja sangat baik pada data tabular yang tidak seimbang. Menunjukkan kekuatan CatBoost dalam prediksi hama/*disease*.
5.	Garba *et al.* (2025)	*Multilabel classification for Predicting Crop Pests in Niger*	- 9 *multilabel classifier (MLkNN, RAkEL, BR, CC, dll).* Evaluasi bulanan & tahunan dengan *Hamming Loss.*	Menggunakan data serangan hama tanaman di Niger dengan beberapa jenis hama muncul bersamaan (*multi-label*). Tujuannya memprediksi kombinasi hama secara simultan, dan menunjukkan efektivitas metode *multi-label* dalam kasus prediksi hama yang terjadi bersamaan. Dari 9 algoritma multilabel, RAkEL menghasilkan performa terbaik dengan *Hamming loss* 3,63% (bulanan) dan 5,1% (tahunan).
6.	Huang *et al.* (2024)	*Predicting the Global Potential Suitable Distribution of Fall Armyworm and Its Host Plants Based on Machine*	- *Random forest* - CatBoost - XGBoost - LightGBM - *Stacking Ensemble Learning* (SEL)	Penelitian memanfaatkan data penginderaan jauh dan survei perlindungan tanaman untuk memprediksi distribusi potensial *fall armyworm* dan beberapa tanaman inangnya pada berbagai skenario iklim. LightGBM

Tabel 1 (Lanjutan)

No.	Penulis	Judul	Metode	Hasil
		Learning Models		menunjukkan performa optimal pada prediksi distribusi 47 tanaman inang, membuktikan kemampuannya dalam menangani data lingkungan berskala besar dan kompleks. Memberikan performa terbaik untuk prediksi dengan nilai AUC 0,98.
7.	Mahesh & Soundrapandiyan (2024)	*Yield Prediction for Crops by Gradient-Based Algorithms*	- CatBoost - LightGBM - XGBoost	Penelitian membandingkan algoritma *boosting* untuk prediksi hasil panen berbasis fitur lingkungan dan penggunaan pestisida. CatBoost menunjukkan performa terbaik dengan akurasi 99,12%, RMSE (*Root Mean Squared Error*) 0,24, dan koefisien determinasi (R^2) tertinggi dibandingkan LightGBM dan XGBoost. Hasil ini menegaskan keunggulan CatBoost dalam memodelkan hubungan kompleks dan nonlinier pada data pertanian.
8.	Wibawa *et al.* (2025)	*Comparison of CatBoost and LightGBM Models for Air Humidity Prediction*	- CatBoost - LightGBM	Penelitian menggunakan data cuaca harian BMKG yang mencakup suhu, curah hujan, durasi penyinaran, dan karakteristik angin untuk

Tabel 1 (Lanjutan)

No.	Penulis	Judul	Metode	Hasil
				memprediksi kelembapan udara. CatBoost menghasilkan performa lebih baik dengan nilai MAE (Mean Absolute Error) 0,057, dibandingkan LightGBM dengan MAE sebesar 0,0617. Hasil ini menunjukkan keunggulan CatBoost dalam menangani data dengan pola temporal dan musiman yang kuat.

2.2. Landasan Teori

Landasan teori menyediakan dasar ilmiah yang menjelaskan konsep dan metode yang relevan dengan penelitian sehingga analisis berjalan terarah. Pada penelitian ini, landasan teori mencakup tiga komponen utama, di antaranya konsep *machine learning* dan algoritma yang digunakan, pemanfaatan Python beserta pustaka pendukungnya, serta teori mengenai hama tebu sebagai objek prediksi.

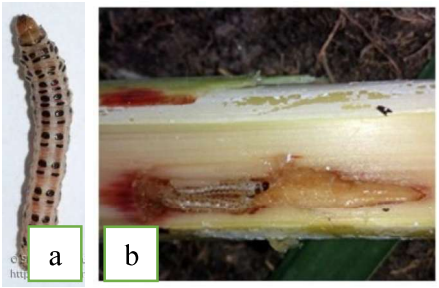
2.2.1. Hama Tebu

Hama tebu merupakan organisme pengganggu tanaman yang dapat menurunkan produktivitas melalui kerusakan pada batang, pucuk, maupun daun. Keberadaan hama sangat dipengaruhi oleh kondisi lingkungan seperti suhu, kelembapan, intensitas hujan, dan fase pertumbuhan tanaman. Berbagai penelitian menunjukkan bahwa beberapa hama tebu di Indonesia meliputi penggerek batang, penggerek pucuk, dan kutu perisai, yang masing-masing

memiliki pola serangan dan dampak yang berbeda (Adrian *et al.*, 2019; Muliasari & Trilaksono, 2020). Pemahaman mengenai karakteristik tiap hama ini diperlukan sebagai dasar dalam membangun model prediksi yang lebih akurat dan responsif terhadap kondisi lapangan.

2.2.1.1. Penggerek Batang (*Stem Borer*)

Penggerek batang tebu (*Chilo sacchariphagus* Bojer) merupakan salah satu hama utama pada budidaya tebu karena menyerang jaringan internal batang pada fase awal pertumbuhan. Serangan biasanya terjadi pada umur tanaman 1,5–2 bulan ketika larva mulai menggerek batang dan merusak jaringan pembuluh yang berfungsi mengangkut air serta nutrisi. Gangguan pada sistem transportasi ini menyebabkan penurunan kualitas batang, berkurangnya rendemen gula, serta kerugian produksi yang signifikan (Muliasari & Trilaksono, 2020). Gambar 1 menunjukkan hama penggerek batang pada tebu.



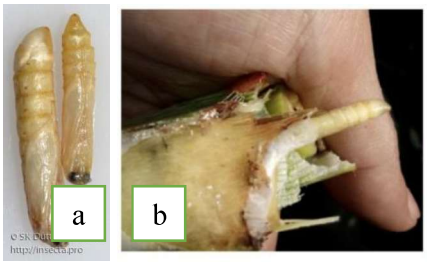
Gambar 1. (a) Larva dan (b) serangan dari *C. sacchariphagus*.
(Alimin, 2022)

Pada bagian (a) terlihat stadium larva (imago) dari penggerek batang yang berbentuk ulat dengan tubuh memanjang dan bersegmen. Sementara pada bagian (b) adalah serangan pada batang tebu, berupa lubang-lubang kecil. Karena serangan penggerek batang terjadi pada fase awal pertumbuhan dan sering belum menunjukkan gejala visual yang jelas, pengendalian konvensional cenderung dilakukan setelah

kerusakan berlangsung. Oleh sebab itu, prediksi intensitas serangan menjadi penting untuk mengidentifikasi periode risiko tinggi secara lebih dini, sehingga strategi pengendalian dapat dilakukan secara preventif berbasis data (Wadhwa & Malik, 2024).

2.2.1.2. Penggerek Pucuk (*Top Borer*)

Penggerek pucuk (*Scirpophaga excerptalis* Walker) menyerang tanaman tebu pada fase awal hingga pertengahan pertumbuhan ketika pucuk masih muda dan rentan. Larva masuk melalui jaringan muda di bagian titik tumbuh dan menggerek bagian dalam pucuk. Serangan ini menimbulkan gejala khas *dead heart*, yaitu matinya pucuk yang berubah warna menjadi coklat dan mudah ditarik. Kondisi ini menghambat pertumbuhan vegetatif dan mengurangi jumlah anakan produktif yang dapat dipanen. Pada Gambar 2 diperlihatkan larva penggerek pucuk pada tebu.



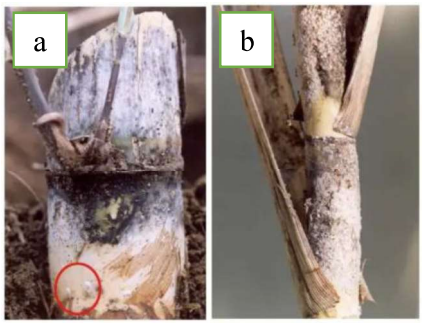
Gambar 2. (a) Pupa dan (b) serangan dari *S. excerptalis* (Alimin, 2022)

Serangan penggerek pucuk sangat berbahaya karena mengganggu fase awal pertumbuhan tebu. Penggerek pucuk mulai menyerang tebu saat usia 2 minggu hingga dewasa (Muliasari & Trilaksono, 2020). Intensitas serangan yang tinggi dapat mengakibatkan penurunan produktivitas secara signifikan, terutama bila terjadi pada lahan dengan kondisi iklim yang mendukung perkembangan populasi hama. Mengingat penggerek pucuk menyerang titik tumbuh tanaman dan

menyebabkan kerusakan yang bersifat permanen, keterlambatan pengendalian dapat berdampak langsung pada penurunan produktivitas. Oleh karena itu, prediksi kemunculan dan intensitas serangan berbasis data lingkungan menjadi krusial untuk menentukan waktu pengendalian yang optimal dan meminimalkan kehilangan hasil (Pituckwanich *et al.*, 2025).

2.2.1.3. Kutu Perisai (*Scale Insect*)

Kutu perisai (*Aulacaspis tegalensis*) memiliki ciri khas berupa tubuh kecil yang dilindungi lapisan seperti perisai, sehingga sering sulit terdeteksi pada tahap awal serangan. Kutu perisai dapat berkembang biak dengan cepat dan menyebabkan penurunan produksi secara signifikan apabila populasinya tidak dikendalikan (Fradzan, 2014). Penelitian yang dilakukan oleh Sunaryo & Hasibuan (2003) menunjukkan bahwa populasi kutu perisai mulai meningkat tajam ketika tanaman memasuki usia 8 bulan dan mencapai puncaknya di usia sekitar 11 bulan, dengan kisaran hingga 300 individu per batang tebu. Kutu perisai termasuk hama yang sulit dikendalikan oleh pestisida karena lokasinya di bawah pelepah daun tebu (Pramono, 2025). Pada Gambar 3 diperlihatkan serangan kutu perisai pada bagian keprasan dan pelepah tebu.



Gambar 3. Serangan kutu perisai pada (a) keprasan (b) pelepah.
(Fradzan, 2014)

Kutu perisai menempel pada permukaan daun dan menyedot cairan sel tanaman. Serangan kutu perisai biasanya ditandai dengan munculnya bintik-bintik atau bercak kuning pada daun, daun tampak mengering, dan lama-kelamaan dapat mengganggu proses fotosintesis. Distribusi hama ini telah ditemukan pada berbagai wilayah perkebunan tebu di Indonesia dan memiliki pola kemunculan yang dipengaruhi kondisi lingkungan dan ketersediaan inang (Pramono, 2025). Penelitian sebelumnya menunjukkan bahwa intensitas serangan kutu perisai berkorelasi langsung dengan penurunan hasil tebu, sehingga keberadaannya penting untuk diprediksi sebagai bagian dari manajemen risiko serangan hama (Adrian *et al.*, 2019; Sunaryo & Hasibuan, 2003).

2.2.2. *Machine Learning*

Machine Learning (ML) merupakan cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma yang mampu belajar dari data untuk melakukan prediksi atau pengambilan keputusan tanpa harus diprogram secara eksplisit (Simeone, 2018). Secara umum, metode ML dibagi menjadi *supervised learning, unsupervised learning, dan reinforcement learning*, di mana masing-masing memiliki karakteristik berbeda sesuai dengan ketersediaan label dan tujuan pembelajaran (Janiesch *et al.*, 2022). *Supervised learning* merupakan pendekatan yang paling relevan dalam penelitian ini, karena digunakan pada data berlabel untuk tugas klasifikasi maupun regresi, termasuk prediksi serangan hama. Berbagai algoritma ML modern bekerja dengan prinsip meminimalkan kesalahan prediksi melalui mekanisme optimisasi dan penyesuaian parameter secara iteratif (Younes, 2024). Dalam konteks data tabular seperti data lingkungan dan serangan hama, *machine learning* banyak digunakan karena mampu mengenali pola nonlinier serta interaksi antar variabel yang sulit dianalisis secara manual (Janiesch *et al.*, 2022). Algoritma berbasis pohon keputusan dan turunannya menjadi pilihan

populer pada data jenis ini karena fleksibilitasnya dalam menangani fitur numerik maupun kategorikal dengan sedikit kebutuhan prapemrosesan (Myles *et al.*, 2004). Penelitian di bidang pertanian juga menunjukkan bahwa *machine learning* efektif dalam memodelkan variabilitas lingkungan, risiko penyakit, dan potensi serangan hama, sehingga mampu memberikan prediksi yang lebih akurat dibandingkan metode konvensional (Pituckwanich *et al.*, 2025; Wadhwa & Malik, 2024). Oleh karena itu, pemahaman konsep *machine learning* menjadi dasar penting dalam pengembangan model prediksi untuk mendukung sistem peringatan dini pada tanaman tebu.

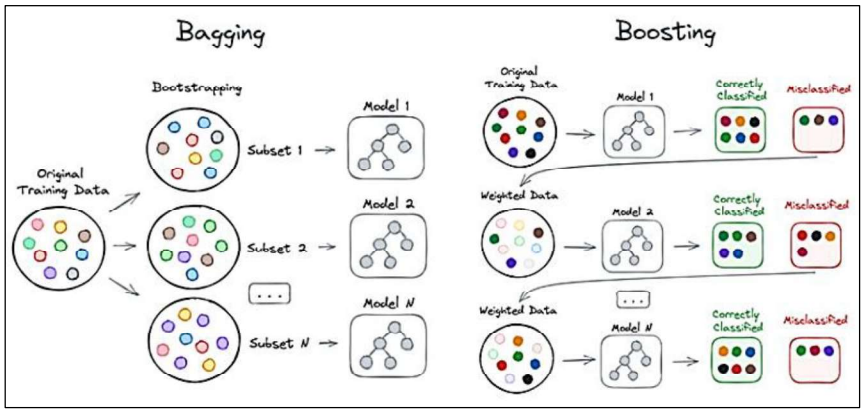
2.2.2.1. *Decision Tree*

Decision tree merupakan algoritma klasifikasi berbasis struktur pohon yang diperkenalkan melalui pendekatan *Classification and Regression Trees* (CART) oleh Breiman *et al.* pada tahun 1984. Algoritma ini bekerja dengan membagi data secara rekursif berdasarkan atribut yang memberikan pemisahan paling informatif, sehingga membentuk rangkaian *node* keputusan hingga mencapai *node* daun sebagai keluaran prediksi (Myles *et al.*, 2004). Setiap pemisahan dilakukan dengan mengukur tingkat *impurity* data menggunakan metrik seperti *gini index* atau *entropy*, sehingga atribut dengan kemampuan pemisahan terbaik dapat dipilih sebagai dasar pembentukan cabang. Keunggulan utama *decision tree* terletak pada interpretabilitasnya yang tinggi, karena struktur pohon memungkinkan pengguna memahami alasan di balik sebuah prediksi melalui jalur keputusan yang terbentuk. Selain itu, algoritma ini dapat menangani fitur numerik maupun kategorikal tanpa perlu banyak tahap prapemrosesan. Namun, penggunaan satu pohon keputusan cenderung rentan terhadap *overfitting* karena model sangat sensitif terhadap perubahan kecil pada data. Keterbatasan ini menjadi alasan dikembangkannya berbagai metode *ensemble* yang menggabungkan banyak pohon untuk meningkatkan stabilitas dan

akurasi, seperti *bagging* dan *boosting* (Breiman, 2001; Friedman, 2001).

2.2.2.2. *Gradient Boosting*

Dalam metode *ensemble learning*, terdapat dua pendekatan utama yang sering digunakan, yaitu *bagging* dan *boosting*. Perbedaan antara *bagging* dan *boosting* dapat dilihat pada Gambar 4.



Gambar 4. Ilustrasi perbedaan *bagging* dan *boosting*.
(Jain, 2024)

Bagging (*Bootstrap Aggregating*) bekerja dengan melatih banyak model secara paralel pada sampel *bootstrap* yang berbeda, kemudian menggabungkan prediksinya untuk mengurangi varians dan meningkatkan kestabilan, seperti pada algoritma *random forest* (Breiman, 2001). Sebaliknya, *boosting* membangun model secara berurutan, di mana setiap model baru difokuskan untuk memperbaiki kesalahan model sebelumnya sehingga menghasilkan prediktor akhir yang lebih akurat (Friedman, 2001).

Gradient boosting merupakan salah satu metode *boosting* yang paling berpengaruh dan banyak digunakan. Algoritma ini menggabungkan sejumlah *weak learners*, biasanya pohon keputusan berukuran kecil,

yang ditambahkan secara bertahap untuk meminimalkan fungsi *loss*. Proses pembaruan model pada iterasi ke-m dapat dituliskan sebagai:

$$F_m(x) = F_{m-1}(x) + v.h_m(x) \quad (1)$$

Di mana $F_m(x)$ adalah model pada iterasi ke-*m,* $h_m(x)$ merupakan pohon regresi yang dilatih untuk mendekati *negative gradient* dari fungsi *loss*, dan v adalah *learning rate.* Persamaan 2 menunjukan fungsi objektif yang diminimalkan.

$$L = \sum_{i=1}^{n} \ell(y_i, F(x_i)) \quad (2)$$

L (*Loss function total*) yaitu nilai kesalahan keseluruhan dari model. Dengan ℓ sebagai fungsi *loss*, misalnya *squared error* atau *log loss*. Fungsi ini digunakan untuk mengukur perbedaan antara nilai sebenarnya (y_i) dengan nilai prediksi ($F(x_i)$) pada setiap data i. Total *loss* dihitunng dengan menjumlahkan seluruh nilai kesalahan dari data ke-1 hingga data ke- n, sehingga semakin kecil nilai L, semakin baik model dalam melakukan prediksi terhadap data. Pendekatan bertahap ini membuat *gradient boosting* unggul dalam menangani hubungan nonlinier dan pola data kompleks, sehingga banyak diterapkan pada data tabular di berbagai domain seperti lingkungan, pertanian, dan bioinformatika (Friedman, 2001).

2.2.2.3. LightGBM

LightGBM (*Light Gradient Boosting Machine*) merupakan pengembangan dari algoritma GBDT (*Gradient Boosted Decision Tree*) yang dirancang untuk memberikan efisiensi tinggi pada proses pelatihan. Berbeda dari implementasi GBDT tradisional yang mencari *split* secara seksama, LightGBM menggunakan pendekatan *histogram-*

based untuk mempercepat proses pemisahan *node* dan mengurangi kompleksitas komputasi (Ke *et al.*, 2017). Selain itu, LightGBM menerapkan strategi *leaf-wise tree growth*, yaitu memilih daun dengan potensi penurunan *loss* terbesar pada setiap iterasi, sehingga model dapat menangkap pola nonlinier. Salah satu keunggulan utama LightGBM adalah integrasi dua teknik optimasi, yaitu GOSS (*Gradient-based One-Side Sampling*) dan EFB (*Exclusive Feature Bundling*). GOSS mempertahankan sampel dengan nilai gradien besar dan mengurangi sampel dengan gradien kecil, sehingga mempercepat pelatihan tanpa menghilangkan informasi penting. EFB menggabungkan fitur yang saling eksklusif untuk mengurangi dimensi efektif, sehingga meminimalkan beban komputasi tanpa menurunkan kualitas pemodelan. Fungsi objektif GBDT pada iterasi ke-*t* dapat direpresentasikan melalui pendekatan deret Taylor orde kedua sebagai berikut:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \tag{3}$$

Persamaan 3 merupakan bentuk pendekatan dari fungsi objektif pada iterasi ke-*t* dalam algoritma LightGBM. Nilai $\mathcal{L}^{(t)}$ menyatakan fungsi *loss* yang diekspansi menggunakan deret Taylor hingga orde kedua. Komponen g_i dan h_i masing-masing merepresentasikan turunan pertama (*gradient*) dan turunan kedua (*hessian*), dari fungsi *loss* terhadap prediksi pada sampel ke-*i.* Sementara itu, $f_t(x_i)$ adalah kontribusi pohon ke-*t* pada sampel ke-*i*, dan $\Omega(f_t)$ merupakan fungsi regulasi untuk mengendalikan kompleksitas model. Pendekatan ini memungkinkan LightGBM memperbarui struktur pohon secara iteratif dengan mempertimbangkan keseimbangan antara penurunan *loss* dan kontrol kompleksitas, sehingga pembelajaran tetap stabil. Berdasarkan kerangka objektif tersebut, proses pencarian *split* pada setiap *node* dilakukan dengan memanfaatkan informasi gradien dan hessian secara efisien melalui pendekatan *histogram-based*. Mekanisme ini tidak

hanya mempercepat proses komputasi, tetapi juga menjadi dasar bagi kemampuan LightGBM dalam menangani fitur numerik maupun kategorikal secara langsung. Untuk fitur kategorikal, LightGBM tidak bergantung pada *one-hot encoding* yang dapat meningkatkan dimensi data secara signifikan. Sebaliknya, kategori diurutkan berdasarkan statistik *gradient* dan *hessian*, kemudian dilakukan pencarian partisi optimal (*optimal partitioning*) guna menentukan kombinasi kategori yang menghasilkan penurunan *loss* terbesar (Ke *et al.*, 2017). Pendekatan ini menjaga efisiensi komputasi sekaligus mengurangi risiko *sparsity* dan *overfitting* akibat ekspansi dimensi yang berlebihan.

Dalam konteks aplikasi, LightGBM telah banyak dimanfaatkan pada penelitian di bidang pertanian dan lingkungan yang melibatkan variabel heterogen serta hubungan nonlinier antar fitur. Penelitian oleh Kumar *et al.* (2024) serta Mahesh & Soundrapandiyan (2024) menunjukkan bahwa LightGBM mampu memodelkan pengaruh variabel lingkungan terhadap kondisi tanaman secara efektif, sementara Silva *et al.* (2024) menegaskan keunggulan algoritma *boosting* dalam menangkap interaksi fitur pada data pertanian. Karakteristik tersebut sejalan dengan kebutuhan penelitian ini, yang berfokus pada prediksi intensitas serangan hama tebu berdasarkan variabel lingkungan dan data pengamatan lapangan.

2.2.2.4. CatBoost

CatBoost (*Category Boosting*) merupakan algoritma *boosting* yang dikembangkan untuk mengatasi dua permasalahan utama dalam *gradient boosting* tradisional, yaitu *prediction shift* akibat *target leakage* serta kesulitan dalam menangani fitur kategorikal secara efisien. CatBoost memperkenalkan mekanisme *ordered boosting*, yaitu proses pembaruan model secara berurutan dengan memanfaatkan

urutan permutasi data agar informasi dari masa depan tidak bocor ke dalam proses pelatihan (Prokhorenkova *et al.*, 2018). Pembaharuan model pada iterasi ke-*m* dirumuskan pada Persamaan 4.

$$F_m(x) = F_{m-1}(x) + \eta . T_m(x) \tag{4}$$

Pada persamaan 4, $F_m(x)$ menyatakan model pada iterasi ke-*m*, $T_m(x)$ adalah pohon keputusan pada literasi ke-*m,* dan η sebagai *learning rate* yang mengontrol besarnya kontribusi pohon baru. Selain itu, CatBoost menggunakan *ordered target statistics* yang menghitung nilai statistik target berdasarkan urutan permutasi sehingga informasi label tidak digunakan sebelum waktunya. Pendekatan ini diformulasikan pada Persamaan 5.

$$\hat{y}_i = \frac{\sum_{j<i} y_j}{i-1} \tag{5}$$

$\hat{y}_i$: Nilai statistik target untuk sampel ke-*i*

y_j : *Label* target dari sampel ke-*j*, dengan syarat $j<i$

i : Indeks sampel dalam urutan permutasi data

Melalui penerapan *ordered boosting* dan *ordered target statistics*, CatBoost mampu mengatasi masalah *label leakage* dan menghasilkan estimasi yang lebih tidak bias. Inovasi ini membuat CatBoost secara konsisten lebih unggul dibandingkan algoritma *boosting* lain, khususnya pada *dataset* dengan proporsi fitur kategorikal yang besar (Prokhorenkova *et al.*, 2018).

CatBoost telah diaplikasikan dalam berbagai studi prediktif berbasis data lingkungan dan pertanian. Penelitian oleh Silva *et al.* (2024) menunjukkan bahwa algoritma boosting, termasuk CatBoost, efektif dalam memodelkan interaksi fitur pada data pertanian yang kompleks, sementara Wibawa *et al.* (2025) membuktikan stabilitas performa

CatBoost pada tugas prediksi variabel lingkungan. Karakteristik tersebut selaras dengan penelitian ini yang melibatkan data lingkungan dan pengamatan lapangan dengan struktur fitur heterogen, sehingga CatBoost relevan digunakan untuk memodelkan intensitas serangan hama pada tebu.

2.2.2.5. Optuna (*Hyperparameter Tuning*)

Hyperparameter tuning merupakan proses mencari kombinasi nilai *hyperparameter* terbaik untuk model *machine learning* agar performanya optimal. Tahap ini penting karena nilai *hyperparameter* sangat memengaruhi kemampuan model dalam mempelajari pola data serta mengontrol risiko *overfitting* maupun *underfitting*. Optuna adalah *framework* optimasi *hyperparameter* berbasis Python yang dirancang untuk melakukan pencarian *hyperparameter* secara otomatis dan efisien. Optuna menggunakan pendekatan *bayesian optimization*, khususnya algoritma TPE (*Tree-structured Parzen Estimator*), untuk menentukan kombinasi *hyperparameter* yang optimal berdasarkan hasil percobaan sebelumnya (Akiba *et al.*, 2019).

Optuna secara iteratif mempelajari distribusi probabilitas *hyperparameter* yang menghasilkan performa baik dan buruk, sehingga proses pencarian menjadi lebih terarah dan hemat komputasi. Keunggulan utama Optuna terletak pada fleksibilitas dan efisiensinya. *Framework* ini mendukung *define-by-run*, yaitu mekanisme yang memungkinkan ruang pencarian *hyperparameter* didefinisikan secara dinamis selama proses optimasi berlangsung. Pendekatan ini sangat sesuai untuk model kompleks seperti *ensemble tree* dan *boosting*, termasuk LightGBM dan CatBoost, yang memiliki banyak *hyperparameter* saling bergantung. Selain itu, Optuna juga menyediakan fitur *pruning*, yang memungkinkan penghentian dini pada

percobaan dengan performa rendah sehingga sumber daya komputasi dapat dialokasikan secara lebih optimal (Akiba *et al.*, 2019).

Sejumlah penelitian menunjukkan bahwa Optuna efektif digunakan untuk mengoptimalkan *hyperparameter* pada model berbasis *boosting*, termasuk CatBoost dan LightGBM. Satria *et al.* (2025) membuktikan bahwa integrasi Optuna pada model CatBoost mampu menghasilkan konfigurasi *hyperparameter* yang lebih optimal dibandingkan pengaturan *default*, sehingga meningkatkan kinerja dan stabilitas model klasifikasi. Selain itu, studi oleh Imani (2023) menunjukkan bahwa penerapan Optuna pada berbagai algoritma *boosting* seperti CatBoost dan LightGBM secara konsisten memperbaiki performa prediktif dengan menemukan konfigurasi yang menghasilkan performa model optimal. Temuan-temuan tersebut mendukung pemilihan Optuna dalam penelitian ini, yang menggunakan model CatBoost dan LightGBM untuk prediksi serangan hama tebu dengan karakteristik data yang melibatkan fitur kategorikal, ketidakseimbangan kelas, dan dependensi temporal, sehingga diperlukan metode optimasi *hyperparameter* yang adaptif dan efektif untuk memperoleh performa model yang optimal.

2.2.3. *Multi-Label Clasification*

Multi-label classification merupakan pendekatan pembelajaran mesin di mana satu *instance* dapat memiliki lebih dari satu label keluaran secara bersamaan. Berbeda dengan *single-label* atau *multi-class classification* yang hanya menghasilkan satu label per *instance*, pendekatan ini memungkinkan prediksi beberapa kategori secara paralel. Dalam konteks penelitian ini, setiap baris data mewakili kondisi satu blok tanaman tebu yang memiliki tiga target sekaligus, yaitu tingkat serangan penggerek pucuk, penggerek batang, dan kutu perisai. Ketiga target ini bersifat multi-kelas sehingga secara teknis penelitian ini termasuk dalam kategori *multi-output multi-class classification*,

namun tetap relevan dalam kerangka *multi-label* karena setiap target diprediksi secara independen dalam satu struktur keluaran. Pada penelitian Zhang & Zhou (2013) ditegaskan bahwa sebuah *instance* dapat memiliki lebih dari satu label dan bahwa setiap label dapat berupa variabel kategorikal yang memiliki lebih dari dua kelas. Permasalahan *multi-label* dapat ditangani melalui pendekatan *problem transformation*, yaitu mengubah permasalahan *multi-label* menjadi beberapa permasalahan klasifikasi terpisah. Pendekatan ini merupakan generalisasi dari metode *binary relevance*, di mana setiap label diperlakukan sebagai target mandiri, tetapi pada penelitian ini setiap target merupakan variabel multi-kelas sehingga model yang dibangun bukan *classifier biner*. Pendekatan per-label seperti ini digunakan secara luas pada domain biologis dan agrikultur yang memiliki karakteristik data terbatas dan distribusi label yang tidak merata. Gidiglo *et al.* (2024) dan Garba *et al.* (2025) menunjukkan bahwa pemisahan model menghasilkan performa yang lebih stabil ketika kombinasi label jarang, hubungan antar label tidak konsisten, atau jumlah sampel tidak besar.

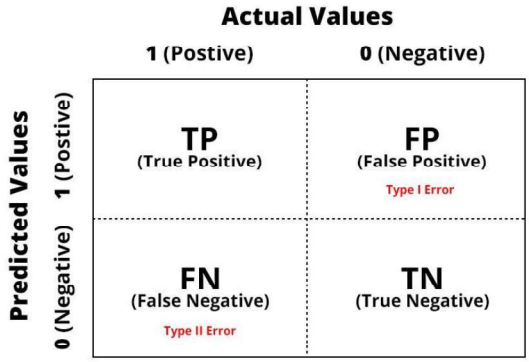
Struktur *multi-label* pada kasus serangan hama tebu bersifat dinamis karena setiap jenis hama dapat muncul secara bersamaan maupun terpisah, bergantung pada kondisi lingkungan dan musim. Mengingat korelasi antar label tidak selalu stabil serta keterbatasan jumlah dan distribusi data, pendekatan *problem transformation* berbasis pemodelan per-label menjadi lebih sesuai untuk konteks penelitian ini. Pendekatan tersebut sejalan dengan temuan Zhang & Zhou (2013) serta studi empiris oleh Gidiglo *et al.* (2024) dan Garba *et al.* (2025), yang menekankan bahwa pemodelan terpisah memberikan stabilitas dan interpretabilitas yang lebih baik pada kasus *multi-label* dengan kombinasi label yang jarang.

2.2.4. Metrik Evaluasi

Metrik evaluasi digunakan untuk menilai sejauh mana model *machine learning* mampu menghasilkan prediksi yang akurat dan sesuai dengan karakteristik data. Pada tugas klasifikasi, khususnya yang bersifat *multi-label* dan *imbalanced*, pemilihan metrik menjadi penting karena akurasi saja tidak cukup mewakili performa model secara menyeluruh (Nugroho, 2019; Sathyanarayanan & Tantri, 2024). Berbagai metrik, baik yang berbasis *confusion matrix* maupun probabilitas, diperlukan untuk memberikan gambaran yang lebih komprehensif mengenai kualitas prediksi (Wilimitis & Walsh, 2023). Dalam penelitian ini, metrik yang digunakan meliputi *confusion matrix* dan ROC AUC, yang semuanya tersedia melalui pustaka Scikit-learn.

2.2.4.1. *Confusion Matrix*

Confusion matrix merupakan salah satu metode evaluasi performa model klasifikasi yang menggambarkan jumlah prediksi benar dan salah dalam bentuk tabel yang memuat *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). Gambar 5 menunjukan representasi hubungan antara prediksi dan kondisi aktual.



Gambar 5. *Confusion matrix*.
(Nugroho, 2019)

Confusion matrix menjadi dasar perhitungan metrik evaluasi utama seperti akurasi, presisi, *recall*, dan *F1-score*, yang memberikan gambaran menyeluruh mengenai kualitas prediksi pada distribusi kelas yang seimbang maupun tidak seimbang (Sathyanarayanan & Tantri, 2024). Perhitungan tersebut dijabarkan dalam bentuk persamaan-persamaan berikut.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$precision = \frac{TP}{TP+FP} \quad (7)$$

$$recall = \frac{TP}{TP+FN} \quad (8)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (9)$$

Akurasi menghitung proporsi seluruh prediksi yang benar. *Precision* mengukur ketepatan prediksi positif dengan membandingkan TP terhadap seluruh prediksi positif. *Recall* menunjukkan kemampuan model dalam mendeteksi kasus positif sebenarnya. *F1-score* memberikan ukuran kinerja yang seimbang pada situasi ketika *precision* dan *recall* perlu dinilai secara bersamaan. Penggunaan *confusion matrix* dan metrik turunannya juga umum diterapkan pada penelitian klasifikasi berbasis *boosting* dan *multi-label* di domain pertanian (Garba *et al.*, 2025; Silva *et al.*, 2024). Pendekatan evaluasi ini relevan dengan penelitian ini karena setiap jenis hama diprediksi sebagai target multi-kelas tersendiri, sehingga diperlukan metrik yang mampu mengevaluasi performa model pada masing-masing kelas secara rinci.

2.2.4.2. ROC & AUC

Receiver Operating Characteristic (ROC) dan *Area Under the Curve* (AUC) merupakan metrik evaluasi yang mengukur kemampuan model

dalam membedakan kelas positif dan negatif berdasarkan skor probabilitas prediksi. Dalam *multi-label classification*, AUC dihitung secara terpisah untuk setiap label sehingga menghasilkan penilaian per-label yang lebih detail. Keunggulan utama AUC adalah kemampuannya untuk tetap memberikan penilaian yang valid meskipun data bersifat *imbalanced*, karena metrik ini berfokus pada *trade-off* antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR), bukan pada jumlah sampel tiap kelas (Gao *et al.*, 2019). AUC secara umum dihitung sebagai luas area di bawah kurva TPR–FPR seperti pada Persamaan 10.

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(t)\ d\big(\mathrm{FPR}(t)\big) \tag{10}$$

Dalam bentuk diskrit:

$$\mathrm{AUC} = \sum_{k=1}^{K-1}(FPR_{k+1} - FPR_k)\frac{TPR_{k+1}+TPR_k}{2} \tag{11}$$

$$TPR = \frac{TP}{TP+FN} \tag{12}$$

$$FPR = \frac{FP}{FP+TN} \tag{13}$$

Variabel TPR dan FPR masing-masing dihitung berdasarkan nilai *True Positive*, *False Positive*, dan *False Negative*. Perubahan nilai *treeshold* t menghasilkan pasangan (TPR, FPR) pada berbagai titik sepanjang kurva ROC, sehingga luas area di bawah kurva tersebut (AUC) menjadi ukuran kemampuan model dalam membedakan kelas positif dan negatif secara keseluruhan. Metrik ini sangat efektif digunakan pada *dataset imbalanced*, karena menilai kemampuan model dalam memeringkatkan *instance* positif lebih tinggi daripada negatif, bukan sekadar menghitung akurasi. Dalam penelitian *multi-label* bidang medis, Zhou *et al.* (2021) menunjukkan bahwa perhitungan AUC setiap label memberikan gambaran yang lebih informatif mengenai performa model pada kasus-kasus *imbalanced*. Prinsip ini juga dapat diterapkan pada prediksi serangan hama, di mana beberapa jenis hama muncul lebih

jarang dibanding lainnya. Dengan demikian, ROC dan AUC menjadi metrik pendukung yang penting untuk mengevaluasi kemampuan model dalam menghasilkan prediksi probabilistik yang tahan terhadap variasi data dan *noise*.

2.2.5. Python

Python adalah bahasa pemrograman tingkat tinggi yang sederhana, dinamis, dan mudah dipelajari. Bahasa ini banyak digunakan dalam komputasi ilmiah karena memiliki pustaka yang luas serta mampu terintegrasi dengan bahasa lain seperti C dan Fortran. Keunggulan tersebut membuat Python efisien sekaligus berperan sebagai penghubung antarpustaka dalam penelitian dan rekayasa ilmiah. Dengan dukungan pustaka ilmiah, Python berkembang dari bahasa skrip menjadi *platform* lengkap untuk perhitungan numerik, visualisasi, serta integrasi perangkat lunak ilmiah (Oliphant, 2007). Dalam konteks penelitian ini, Python berperan sebagai fondasi komputasi untuk seluruh proses analisis, mulai dari pengolahan data, konstruksi model *machine learning*, hingga evaluasi. Dukungan pustaka seperti Pandas, NumPy, Matplotlib, dan Scikit-learn membuat proses analisis lebih konsisten, terstruktur, dan reprodusibel, sehingga sesuai dengan kebutuhan pemodelan *multi-label multi-class* berbasis data tabular.

2.2.5.1. Pandas

Pandas merupakan pustaka Python yang menyediakan struktur data berorientasi tabel melalui objek Series dan DataFrame, sehingga proses pembersihan, transformasi, dan manipulasi data dapat dilakukan secara efisien. Mckinney (2010) memperkenalkan Pandas sebagai fondasi analisis data modern karena mendukung operasi seperti *filtering*, *grouping*, *join/merge*, penanganan nilai hilang, hingga transformasi

kolom dalam skala besar. Dalam konteks penelitian ini, Pandas digunakan untuk seluruh tahap awal pengolahan data, mulai dari membaca *dataset* mentah, menyelaraskan tipe data, rekayasa fitur, hingga membentuk struktur *multi-label* yang sesuai dengan keperluan pemodelan. Fungsinya yang fleksibel memungkinkan peneliti mengekstrak pola dari data agroklimat dan serangan hama secara sistematis sebelum dilakukan pemodelan *machine learning*. Pandas menjadi komponen utama dalam menyiapkan *dataset* yang bersih, konsisten, dan siap diproses oleh model.

2.2.5.2. NumPy

NumPy digunakan sebagai fondasi komputasi numerik dalam ekosistem Python. NumPy menyediakan struktur *array* multidimensi dengan kemampuan operasi vektorisasi, komputasi aljabar linier, serta manipulasi matriks yang efisien. Harris *et al.* (2020) menegaskan bahwa NumPy menjadi dasar bagi sebagian besar *library* ilmiah karena mendukung operasi vektorisasi, manipulasi matriks, transformasi numerik, hingga komputasi aljabar linier dengan efisiensi tinggi. NumPy digunakan dalam penelitian ini untuk mempercepat perhitungan matematis pada tahap pra-pemrosesan, seperti normalisasi fitur, konversi tipe data, serta pembuatan matriks input untuk model. NumPy mendukung kebutuhan penelitian ini karena sebagian besar algoritma *machine learning* dan pengolahan data bergantung pada operasi numerik berbasis *array*.

2.2.5.3. Matplotlib

Matplotlib merupakan pustaka visualisasi yang digunakan untuk mempresentasikan pola distribusi data, hubungan antar variabel, dan

tren waktu. Visualisasi berperan penting pada tahap eksplorasi data karena membantu peneliti memahami karakteristik tiap label sebelum model dibangun. Hunter (2007) menunjukkan bahwa Matplotlib menjadi komponen visualisasi fundamental dalam analisis ilmiah berbasis Python karena fleksibilitas tinggi dan kompatibilitasnya dengan *library* lain. Matplotlib digunakan dalam penelitian ini untuk mendukung eksplorasi data serta menampilkan hasil analisis secara informatif.

2.2.5.4. Scikit-learn

Scikit-learn merupakan pustaka utama dalam pemodelan *machine learning*. *Library* ini menyediakan implementasi algoritma klasifikasi, regresi, pemilihan fitur, serta *pipeline* pemrosesan data yang terstandarisasi. Pedregosa *et al.* (2011) menjelaskan bahwa Scikit-learn dirancang untuk penelitian ilmiah karena konsistensi API dan *reproducibility* hasil pemodelan. Fungsi klasifikasi *multi-class* dalam Scikit-learn digunakan dalam penelitian ini untuk memodelkan masing-masing label tingkat serangan hama secara terpisah dalam pendekatan *multi-output*. *Library* ini mendukung seluruh proses pemodelan mulai dari pelatihan, validasi, hingga penyusunan prediksi.

III. METODOLOGI

3.1. Tempat dan Waktu Penelitian

3.1.1. Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Komputasi Dasar, Program Studi Ilmu Komputer, Universitas Lampung. Pemilihan lokasi penelitian didasarkan pada ketersediaan fasilitas yang memadai untuk mendukung jalannya penelitian, meliputi perangkat komputer dengan spesifikasi yang sesuai, perangkat lunak analisis data, serta jaringan internet untuk menunjang kebutuhan literatur maupun pengolahan informasi. Selain itu, Laboratorium Komputasi Dasar juga dipilih karena memiliki lingkungan akademik yang kondusif, sehingga mendukung dalam menjalankan aktivitas penelitian secara terarah, sistematis, dan sesuai dengan tujuan yang telah ditetapkan.

3.1.2. Waktu Penelitian

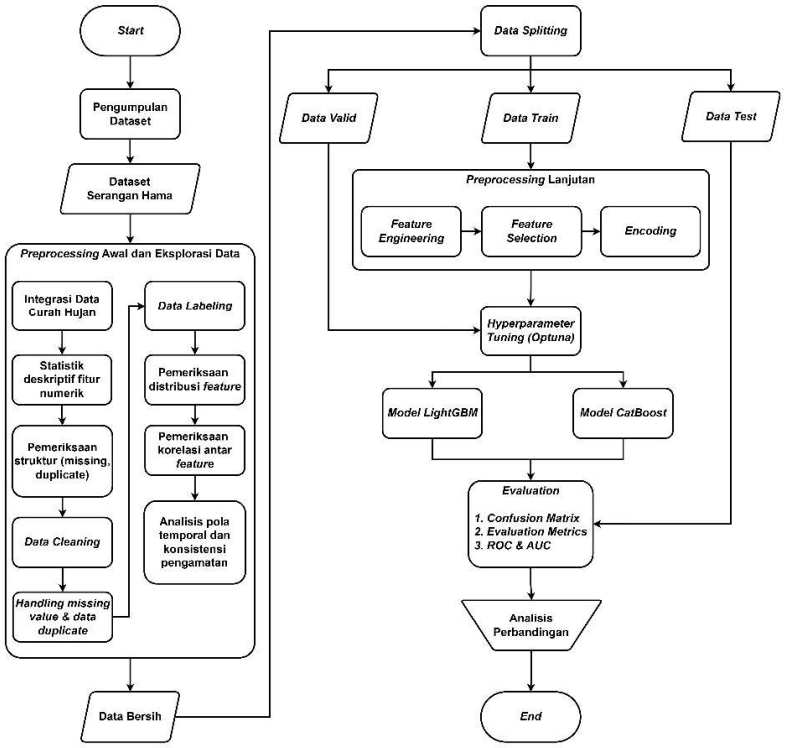
Pelaksanaan penelitian telah dilaksanakan pada periode Agustus 2025 hingga Januari 2026. Rentang waktu tersebut mencakup tahap persiapan, pengumpulan data, analisis, serta penyusunan laporan penelitian. Studi literatur dilakukan sebelum dan selama proses penelitian berlangsung sebagai dasar dalam pemilihan metode serta interpretasi hasil penelitian. Rincian jadwal pelaksanaan penelitian disajikan pada Tabel 2.

Tabel 2. Waktu penelitian.

Aktivitas	2025																				2026			
	Agustus				September				Oktober				November				Desember				Januari			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengumpulan Data																								
Exploratory Data Anaysis																								
Penyusunan Proposal																								
Preprocessing																								
Modeling																								
Evaluation																								
Penyusunan Laporan																								

3.2. Tahapan Penelitian

Tahapan penelitian ini dilakukan secara sistematis agar hasil yang diperoleh lebih terarah. Secara umum, alur penelitian dapat dilihat pada Gambar 6 berikut.



Gambar 6. Alur penelitian.

Preprocessing awal dan eksplorasi data dilakukan sebelum pembagian *dataset* untuk memastikan kualitas data. Proses *encoding* dipelajari dari data latih, kemudian diterapkan secara konsisten pada data validasi dan data uji.

3.2.1. Pengumpulan *dataset*

Dataset pertama yang digunakan dalam penelitian ini adalah *dataset* serangan hama yang berisi 9.954 baris dengan 23 kolom. *dataset* tersebut mencakup informasi identitas lahan, waktu pengamatan, karakteristik penanaman, hingga tingkat serangan hama dan perlakuan pengendalian. Pada Tabel 3 telah dijabarkan atribut yang terdapat dalam *dataset* serangan hama.

Tabel 3. Fitur *dataset* serangan hama

No.	Atribut	Keterangan
1.	Unique Code	Kode unik untuk tanaman tebu yang sedang diamati. masing-masing kode terdiri dari nomor divisi, tahun pengamatan, varietas, bulan pengamatan, dan angka acak.
2.	Divisi	Nama divisi ditandai dengan angka romawi dari I–VII.
3.	Tahun Pengamatan	Pengamatan tebu dimulai pada pertengahan tahun 2022 hingga pertengahan tahun 2023.
4.	Bulan Pengamatan	Pengamatan tebu dilakukan selama 12 bulan penuh yang diawali pada bulan Agustus. Kolom ini berisi nilai dengan tipe data *integer* 1-12.
5.	Kawasan	Nama kawasan ditandai dengan angka dari 1-31, yang membuat kolom ini memiliki tipe data *float*.

Tabel 3 (Lanjutan)

No.	Atribut	Keterangan
6.	Varietas Tebu	Tebu yang diamati terdiri dari 15 varietas di antaranya G11, GMP 3, GMP 5, GMP 6, GMP 7, R02, R06, R08, R2, R34, R54, R69, R7, R86, dan Lain-lain
7.	Bulan Tanam2	Berisi nilai dengan tipe data *integer* dari 5 – 11 di mana masa penanaman tebu di mulai pada bulan Mei
8.	Nama Blok	Terdiri dari 1.230 nilai unik kombinasi huruf dan angka.
9.	Tanggal Pengamatan	Berisi tanggal pengamatan tebu dalam format MM/DD/YYYY.
10.	Luas Tanah Produktif (Ha)	Berisi nilai dengan tipe data *float* yang merepresentasikan luas lahan perkebunan dalam satuan hektar.
11.	Bulan Tanam	Berisi tanggal penanaman tebu dalam format MM/DD/YYYY.
12.	Tahun Tanam	Tahun ketika tebu ditanam hanya tersedia di tahun 2022.
13.	Category Tanam	Berisi 5 kategori tebu yang dibedakan berdasarkan masa panen. PC mewakili tebu yang ditanam dengan bibit baru. RC mewakili tebu yang sudah dipanen dan dibiarkan tumbuh lagi setelah penebangan. RC sendiri terdiri dari RC1 – RC5, yang menandai intensitas penebangan yang dilakukan pada tebu tersebut.
14.	Category 2	Berisi 3 kategori tebu saja, yaitu PC, RC1, dan RC2.

Tabel 3 (Lanjutan)

No.	Atribut	Keterangan
15.	Umur	Terdiri dari usia tebu ketika dilakukan pengamatan, yaitu mulai usia 3 – 11 bulan.
16.	Ruas Terserang Stem borer (%)	Persentase ruas batang tebu yang terserang hama *stem borer*. Nilai berupa angka desimal dengan tipe data *float*.
17.	Batang Terserang Top borer (%)	Persentase batang tebu yang mengalami serangan hama *top borer*. Data disajikan dalam bentuk *float*.
18.	Populasi Kutu Perisai (%)	Persentase populasi kutu perisai yang menyerang tanaman tebu. Nilai berbentuk desimal dengan tipe data *float*.
19.	Dosis pias	Jumlah pelepasan atau aplikasi agen pengendali hayati pias yang diberikan, dengan tipe data *integer*.
20.	Dosis lalat	Banyaknya pelepasan lalat sebagai agen pengendali, dicatat dalam *integer*.
21.	Dosis telenomus	Jumlah penggunaan parasitoid *Telenomus* untuk pengendalian hama, dinyatakan dalam *integer*.
22.	Dosis Tetras	Banyaknya pelepasan parasitoid *Tetrastichus* pada tanaman tebu. Data bertipe *integer*.
23.	Dosis cecopet	Takaran atau dosis penggunaan cecopet (agen hayati tertentu), dicatat dalam nilai desimal dengan tipe data *float*.nn

Dataset kedua merupakan *dataset* yang berisi informasi terkait curah hujan di setiap divisi. Informasi curah hujan nantinya akan diintegrasikan pada *dataset* pertama dengan menyesuaikan nama divisi dan tanggal observasi. Pada Tabel 4 telah dijabarkan atribut yang terdapat dalam *dataset* curah hujan.

Tabel 4. Fitur *dataset* curah hujan.

No.	Atribut	Keterangan
1.	Tahun	Pengamatan tebu dimulai pada pertengahan tahun 2017 hingga pertengahan tahun 2023.
2.	Bulan Pengamatan	Pengamatan tebu dilakukan selama 12 bulan penuh yang diawali pada bulan Agustus. Kolom ini berisi nilai dengan tipe data *integer* 1-12.
3.	Divisi	Nama divisi ditandai dengan angka romawi dari I–VII.
4.	Curah Hujan	Intensitas hujan di wilayah perkebunan setiap bulan selama Agustus 2022 hingga Juli 2023.

3.2.2. Eksplorasi Data

Tahap EDA (*Exploratory Data Analysis*) dilakukan untuk memahami struktur awal *dataset* serangan hama tebu serta memastikan kualitas data sebelum masuk ke proses *preprocessing* dan pemodelan. Analisis dilakukan pada data mentah dan data setelah pembersihan awal untuk mengidentifikasi masalah input, distribusi variabel, serta pola temporal yang relevan dengan proses pemantauan hama.

a. Pemeriksaan stuktur data:

   Langkah pertama dalam EDA adalah pemeriksaan struktur *dataset* untuk mengidentifikasi tipe fitur numerik dan kategorikal, kelengkapan data, serta konsistensi pencatatan. Pemeriksaan ini bertujuan untuk mendeteksi potensi permasalahan seperti nilai hilang, duplikasi data, serta inkonsistensi format input akibat proses pencatatan manual. Analisis struktur ini menjadi dasar dalam menentukan kebutuhan pembersihan data dan standarisasi kategori sebelum tahap *preprocessing* lanjutan.

b. Analisis statistik deskriptif dan distribusi fitur numerik:

   Untuk memahami karakteristik dasar fitur numerik, digunakan analisis statistik deskriptif melalui fungsi 'df.describe()' , yang mencakup ukuran pemusatan dan penyebaran data. Selain itu, histogram digunakan untuk memvisualisasikan distribusi setiap fitur numerik. Tujuan dari analisis ini adalah untuk mengidentifikasi rentang nilai, kecenderungan distribusi, serta potensi keberadaan nilai ekstrem yang perlu dipertimbangkan dalam proses penanganan *outlier* atau transformasi data.

c. Analisis distribusi fitur kategorikal:

   Distribusi fitur kategorikal seperti dianalisis menggunakan diagram batang. Visualisasi ini digunakan untuk memahami proporsi data pada setiap kategori serta mengidentifikasi ketidakseimbangan frekuensi antar kategori. Informasi ini penting dalam menentukan strategi *encoding* yang sesuai serta dalam mengevaluasi potensi bias akibat dominasi kategori tertentu.

d. Analisis hubungan antar variabel:

   Untuk mengevaluasi hubungan antar fitur numerik serta tingkat keterkaitan antar variabel, digunakan *heatmap* korelasi. Selain itu, heatmap juga dimanfaatkan untuk memvisualisasikan tingkat keparahan serangan hama pada berbagai kombinasi fitur. Analisis ini bertujuan untuk memberikan gambaran awal mengenai independensi atau keterkaitan antar fitur, yang menjadi pertimbangan dalam pemilihan fitur dan perancangan model prediksi.

e. Analisis pola temporal:

   Pola temporal dianalisis menggunakan diagram garis untuk memvisualisasikan tren variabel berbasis waktu, khususnya curah hujan dan intensitas serangan masing-masing jenis hama dalam rentang satu tahun pengamatan. Analisis ini bertujuan untuk memahami dinamika musiman serta konsistensi pencatatan data berbasis waktu, yang relevan dalam pemodelan prediksi berbasis temporal. Selain itu, kondisi kelengkapan data umur tanaman pada setiap blok pengamatan turut

diperhatikan, di mana ringkasan jumlah blok dengan data pengamatan yang tidak lengkap disajikan pada Lampiran 1.

3.2.3. *Preprocessing*

Tahap *preprocessing* dilakukan untuk menyiapkan data agar siap digunakan pada proses pemodelan. *Preprocessing* mencakup pembersihan data, integrasi sumber data tambahan, pembentukan label dan fitur, serta transformasi lanjutan yang dilakukan setelah pembagian data untuk menjaga validitas evaluasi model. Secara umum, *preprocessing* dibagi menjadi dua tahapan, yaitu *preprocessing* awal dan *preprocessing* lanjutan, sebagaimana ditunjukkan pada diagram alur penelitian.

a. *Preprocessing* awal dan eksplorasi data.
   i. Integrasi data curah hujan:
      Dataset utama serangan hama digabungkan dengan data curah hujan bulanan berdasarkan kesesuaian waktu pengamatan. Integrasi ini bertujuan untuk menambahkan konteks lingkungan yang relevan, mengingat curah hujan merupakan salah satu faktor eksternal yang dapat memengaruhi dinamika populasi hama pada tanaman tebu. Proses ini dilakukan menggunakan skrip Python, sebagaimana ditunjukkan pada Lampiran 2.
   ii. Pembersihan data (*data cleaning*):
      Pembersihan data dilakukan untuk mengatasi ketidakteraturan yang umum ditemukan pada data lapangan. Proses ini meliputi penyesuaian pada nama fitur (Lampiran 3). Selain itu, baris data yang teridentifikasi sebagai duplikasi dihapus untuk mencegah bias dalam pelatihan model. Beberapa nama blok dengan pola pengamatan yang tidak logis, seperti umur tanaman yang tidak runtut atau jumlah pengamatan yang sangat terbatas, dikeluarkan agar struktur temporal data tetap konsisten (Lampiran 4). Nilai ”lain-lain” dan ”Lain-lain” pada kolom ”Varietas” (Lampiran 5)

juga dihapus karena tidak merepresentasikan suatu varietas secara jelas.

iii. Penanganan nilai hilang (*handling missing value*):

Penanganan nilai hilang dilakukan secara hati-hati dengan imputasi berdasarkan informasi yang sama. Untuk kasus nilai hilang pada kolom ”Kawasan” yang terdapat pada Lampiran 6 dilakukan imputasi dengan mengisi bagian tersebut berdasarkan nilai kawasan pada nama blok yang sama (Lampiran 7). Kemudian proses imputasi terhadap data pengamatan yang tidak lengkap dilakukan menggunakan beberapa pendekatan, termasuk imputasi dengan teknik *forward fill* (Lampiran 8), serta berbasis kawasan dan nama blok (Lampiran 9). Selain itu, dilakukan perbaikan tanggal pengamatan untuk menjaga konsistensi temporal data, dengan implementasi fungsi ditunjukkan pada Lampiran 10.

iv. Pelabelan data:

Setelah data dinyatakan bersih dan konsisten, dilakukan pembentukan label target. Proses pelabelan tingkat keparahan serangan hama dilakukan menggunakan fungsi sebagaimana ditunjukkan pada Lampiran 11. Kelas intensitas serangan hama ditentukan berdasarkan aturan klasifikasi yang digunakan dalam penelitian ini. Rincian aturan pelabelan disajikan pada Tabel 5 dan 6 berikut.

Tabel 5. Aturan *label* serangan penggerek batang dan penggerek pucuk.

Kategori	**Umur Tanaman (bulan)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tinggi	> 5.5	> 6.0	> 6.5	> 7.0	> 7.5	> 8.0	> 8.5	> 9.0	> 9.5	> 10
Sedang	0.5 - 5.5	1.0 - 6	1.5 - 6.5	2.0 - 7.0	2.5 - 7.5	3.0 - 8.0	3.5 - 8.5	4.0 - 9.0	4.5 - 9.5	5.0 - 10
Rendah	< 0.5	<1.0	< 1.5	< 2.0	< 2.5	< 3.0	< 3.5	< 4.0	< 4.5	< 5.0

Tabel 6. Aturan *label* serangan kutu perisai.

Kategori	Umur Tanaman (bulan)									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Tinggi	> 10	> 10	> 10	> 10	> 10	> 10.5	> 11.0	> 15	> 20	> 40
Sedang	0.1 – 10.0	0.1 – 10.0	0.1 – 10.0	0.1 – 10.0	0.1 – 10.0	0.5 - 10.5	1.0 - 11.0	5.0 - 15.0	10.0 - 20.0	20.0 - 40.0
Rendah	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.5	< 1	< 5	< 10	< 20

b. *Preprocessing* Lanjutan.

Preprocessing lanjutan dilakukan setelah proses pembagian data menjadi data latih, data validasi, dan data uji, sebagaimana ditunjukkan pada diagram alur penelitian. Tahapan ini bertujuan untuk menyiapkan fitur agar dapat digunakan secara optimal oleh algoritma pembelajaran mesin, sekaligus mencegah terjadinya *data leakage* (kebocoran data).

i. *Feature engineering*:

Pada tahap ini rekayasa fitur temporal berupa *lag feature* diterapkan untuk menangkap ketergantungan antar waktu pengamatan, dengan implementasi kode disajikan pada Lampiran 12. Fitur *lag* dibentuk dengan memanfaatkan nilai pengamatan pada tiga bulan sebelumnya untuk memprediksi kondisi pada bulan pengamatan berikutnya, sehingga dinamika perubahan serangan hama dalam jangka pendek dapat direpresentasikan secara lebih eksplisit dalam data. Pemilihan rentang tiga bulan didasarkan pada karakteristik *dataset* yang hanya mencakup satu periode pengamatan, yaitu dari usia tebu 3 hingga 11 bulan, sehingga tidak memungkinkan untuk menangkap pola musiman atau tren tahunan. Penelitian yang dilakukan Shafiuzzaman *et al.* (2025) menunjukkan bahwa integrasi *multi-lag feature engineering* pada model berbasis *boosting* mampu meningkatkan akurasi peramalan jangka pendek dengan menangkap hubungan nonlinier dan dependensi temporal yang tidak tercermin dari nilai saat ini saja.

ii. *Feature selection*:

Seleksi fitur dilakukan secara spesifik untuk setiap jenis hama dengan mempertimbangkan perbedaan strategi pengendalian yang diterapkan di lapangan. Setiap hama dikendalikan menggunakan agen hayati yang berbeda, baik dari sisi jenis organisme pengendali maupun dosis aplikasi yang diberikan. Dalam praktik lapangan, dosis agen hayati ditentukan berdasarkan karakteristik biologis masing-masing hama serta responsnya terhadap perlakuan tertentu. Oleh karena itu, fitur-fitur yang relevan untuk memodelkan satu jenis hama belum tentu informatif untuk hama lainnya. Pendekatan ini menghasilkan tiga *set* fitur yang berbeda, yang masing-masing digunakan untuk membangun model klasifikasi tersendiri.

iii. *Encoding* fitur kategorikal:

Fitur kategorikal pada penelitian ini diperlakukan sesuai dengan karakteristik masing-masing algoritma. Seluruh kolom bertipe objek terlebih dahulu dikonversi ke tipe *category* pada pandas, termasuk variabel Kawasan yang direpresentasikan sebagai angka namun secara konseptual bersifat kategorikal. Pada CatBoost, fitur kategori diproses secara native melalui parameter 'cat_features' dengan mekanisme *ordered target statistics*. Pada LightGBM, kolom kategori diinformasikan melalui parameter 'categorical_feature' dan diproses secara internal saat pembentukan *split* tanpa asumsi hubungan ordinal. Berbeda dengan fitur, variabel target tetap ditransformasikan menggunakan 'LabelEncoder' dari *library* Scikit-learn. Proses ini mengonversi label kelas ("Rendah", "Sedang", "Tinggi") menjadi representasi numerik *integer* agar kompatibel dengan kebutuhan algoritma klasifikasi multi-kelas. Transformasi ini tidak mengubah makna kelas, melainkan hanya mengubah representasi label agar dapat diproses oleh model.

3.2.4. Pembagian Data

Pembagian data dilakukan untuk memastikan proses pelatihan, pemilihan model, dan evaluasi dilakukan secara objektif serta terhindar dari *data leakage*. Pada penelitian ini, pembagian data tidak dilakukan secara acak per baris, melainkan berbasis unit blok (*block-based splitting*), di mana nama blok diperlakukan sebagai unit observasi yang independen. Pendekatan ini dipilih karena data serangan hama memiliki keterkaitan spasial dan temporal dalam satu blok yang sama. Jika data dari blok yang sama muncul pada lebih dari satu *subset*, maka model berpotensi mempelajari pola spesifik blok tersebut, sehingga evaluasi performa menjadi tidak representatif terhadap kondisi lapangan yang sesungguhnya. Untuk memastikan nama blok yang mewakili divisi – kawasan pada data validasi dan data uji sudah pernah dipelajari oleh model, dibuat fungsi seperti pada Lampiran 13. Fungsi tersebut mengelompokkan divisi – kawasan yang memiliki nama blok terbatas ke dalam data latih. Pembagian data dilakukan dengan tujuan berikut:

a. Data latih (*training set*) digunakan untuk melatih model pembelajaran mesin serta membangun pola hubungan antara fitur *input* dan target. Proses *encoding* dipelajari (*fit*) hanya dari data latih bertujuan untuk memastikan model tidak memperoleh informasi dari data di luar proses pelatihan.
b. Data validasi (*validation set*) dimanfaatkan dalam proses *hyperparameter tuning* menggunakan metode Optuna. Evaluasi pada data validasi memungkinkan pemilihan konfigurasi model terbaik tanpa mengekspos data uji, sehingga objektivitas evaluasi akhir tetap terjaga.
c. Data uji (*test set*) digunakan secara eksklusif untuk mengevaluasi performa akhir model yang telah dilatih dan ditetapkan hiperparameternya. Data ini tidak terlibat dalam proses pelatihan maupun pemilihan model, sehingga hasil evaluasi pada data uji mencerminkan kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya.

3.2.5. Pemodelan

Pada tahap ini dilakukan pembangunan model *machine learning* menggunakan dua algoritma berbasis *gradient boosting*, yaitu CatBoost dan LightGBM. Kedua algoritma dipilih karena mampu menangani kombinasi fitur numerik dan kategorikal, serta dikenal memiliki performa yang baik pada permasalahan klasifikasi dengan data tabular dan struktur kompleks. Pemodelan dilakukan dengan pendekatan *multi-label multi-class classification*, di mana setiap jenis hama diprediksi menggunakan model yang dilatih secara terpisah. Dengan pendekatan ini, setiap model dapat mempelajari karakteristik spesifik dari masing-masing jenis hama secara lebih fokus, sekaligus mempermudah analisis dan interpretasi hasil prediksi. Dua algoritma digunakan dengan penyesuaian terhadap karakteristik masing-masing metode.

a. Pada model CatBoost, fitur kategorikal diproses secara *native* dengan mendefinisikan indeks kolom kategori melalui parameter 'cat_features'. CatBoost kemudian menerapkan mekanisme *ordered target statistics* dalam proses pembelajaran. Untuk mengatasi ketidakseimbangan distribusi kelas, CatBoost dikonfigurasi menggunakan 'auto_class_weights' dengan skema 'SqrtBalanced'. Pendekatan ini secara otomatis menyesuaikan bobot setiap kelas berdasarkan akar kuadrat dari frekuensi kemunculannya, sehingga memberikan penalti lebih besar pada kelas minoritas namun tetap menjaga stabilitas pelatihan dan mencegah pemberian bobot yang terlalu ekstrem pada kelas dengan jumlah sampel sangat sedikit. Skema ini dipilih untuk mencapai keseimbangan antara sensitivitas terhadap kelas minoritas dan stabilitas model selama proses pembelajaran.
b. Pada model LightGBM, fitur kategorikal juga tidak ditransformasikan menggunakan *label encoding*. Seluruh kolom kategori dikonversi ke tipe *category* dan diinformasikan melalui parameter 'categorical_feature' pada saat pelatihan. Dengan pendekatan ini, LightGBM menangani pemisahan kategori secara internal dalam proses

pembentukan *split* pohon keputusan tanpa mengasumsikan hubungan ordinal antar kategori. Untuk menangani permasalahan ketidakseimbangan kelas, LightGBM dikonfigurasi dengan parameter class_weight="balanced", yang secara otomatis menghitung bobot kelas berdasarkan proporsi terbalik dari frekuensi kemunculan masing-masing kelas pada data latih. Pendekatan ini memungkinkan model memberikan perhatian yang lebih besar pada kelas minoritas tanpa perlu perhitungan bobot secara manual, serta sesuai dengan mekanisme pembelajaran LightGBM yang sensitif terhadap distribusi kelas pada proses pembentukan pohon.

Kemudian penentuan *hyperparameter* dilakukan menggunakan metode Optuna dengan memanfaatkan data validasi untuk mencari konfigurasi terbaik bagi model CatBoost dan LightGBM. Optuna dipilih karena merupakan kerangka kerja optimasi *hyperparameter* yang dirancang untuk menjelajahi ruang parameter secara adaptif dan efisien. Optuna menerapkan algoritme *sampling* yang adaptif dan mekanisme *early stopping* (*pruning*) untuk menghentikan percobaan parameter yang berkinerja buruk lebih awal, sehingga mengurangi komputasi yang tidak perlu dan mempercepat proses optimasi. Daftar parameter yang digunakan kedua model dapat dilihat pada Tabel 7.

Tabel 7. Fungsi parameter model.

LightGBM	**CatBoost**	**Fungsi Parameter**
objective	loss_function	Menentukan fungsi objektif / jenis masalah (*multiclass classification*).
num_class	—	Jumlah kelas target (CatBoost infer otomatis dari data).
max_depth	depth	Kedalaman maksimum pohon.
num_leaves	—	Jumlah daun per pohon (kontrol kompleksitas di LightGBM).

Tabel 7 (Lanjutan)

LightGBM	CatBoost	Fungsi Parameter
learning_rate	learning_rate	Besar langkah pembaruan model di setiap iterasi *boosting*.
n_estimators	iterations	Jumlah pohon (*tree*) yang dibangun selama *boosting.*
reg_lambda	l2_leaf_reg	Regularisasi L2 untuk menekan bobot besar dan mengurangi *overfitting*.
reg_alpha	—	Regularisasi L1 untuk mendorong *sparsity* bobot fitur (khusus LightGBM).
min_child_ samples	—	Minimum jumlah data pada satu *leaf* untuk mencegah *split* yang terlalu spesifik.
subsample	—	Proporsi data untuk tiap *tree.*
colsample_ bytree	—	Proporsi fitur pada tiap *tree*.
class_weight	auto_class_ weights	Penanganan *data imbalance.*
early_stopping _rounds	early_stopping _rounds	Hentikan *training* jika validasi tidak membaik.
categorical_ feature	cat_features	Penanda fitur kategorikal.

Rentang nilai untuk setiap *hyperparameter* ditentukan berdasarkan praktik umum pada algoritma *boosting* dan disesuaikan dengan karakteristik data. Parameter yang diuji selama proses optimasi untuk masing-masing algoritma disajikan pada Tabel 8.

Tabel 8. Nilai yang diuji dalam *hyperparameter tuning*.

Algoritma	Parameter	Nilai
CatBoost	depth	4, 6, 8
	learning_rate	0.01 – 0.1
	l2_leaf_reg	5 – 20
	iterations	200, 400, 600
LightGBM	max_depth	3, 4, 5
	num_leaves	7, 15, 31
	learning_rate	0.01 – 0.1
	n_estimators	200, 400, 600
	reg_alpha	0.5 – 5.0
	reg_lambda	0.0 – 1.0
	min_child_samples	20, 50, 100
	feature_fraction / colsample_bytree	0.6, 0.7, 0.8
	bagging_fraction / subsample	0.6, 0.8

3.2.6. Evaluasi Model

Evaluasi model dilakukan untuk menilai kualitas prediksi yang dihasilkan masing-masing model pada setiap target hama. Proses evaluasi disusun agar sesuai dengan alur penelitian serta menggambarkan performa model pada kondisi lapangan sebenarnya. Evaluasi dilakukan menggunakan *test set*, sehingga model benar-benar diuji pada nama blok yang berbeda. Dengan cara ini, performa model merepresentasikan kemampuan prediksi ke depan (*forward prediction*) dan tidak dipengaruhi kebocoran informasi dari masa depan. Model dievaluasi menggunakan beberapa metrik untuk memberikan gambaran menyeluruh terhadap performanya:

a. *Macro-average F1-score* digunakan sebagai metrik utama karena lebih representatif pada data dengan distribusi kelas tidak seimbang. Perhitungan dilakukan dengan menghitung *F1-score* untuk setiap kelas

secara terpisah, kemudian dirata-ratakan tanpa mempertimbangkan proporsi jumlah sampel tiap kelas. Pendekatan ini memastikan bahwa kelas minoritas memiliki kontribusi yang setara dalam evaluasi, sehingga performa model tidak bias terhadap kelas mayoritas.

b. *Confusion matrix* digunakan untuk menganalisis pola kesalahan klasifikasi. Melalui matriks ini dapat diamati kecenderungan model dalam salah mengklasifikasikan tingkat keparahan tertentu, serta mengidentifikasi kelas yang paling sulit diprediksi.
c. *Area Under the Curve* (AUC) dari ROC *curve* digunakan untuk mengukur kemampuan model dalam membedakan antar kelas pada berbagai ambang keputusan. Selain nilai AUC per kelas, penelitian ini juga menggunakan AUC *macro* untuk setiap label hama, sehingga diperoleh gambaran performa agregat yang tidak dipengaruhi dominasi kelas mayoritas. Pendekatan ini memungkinkan analisis yang lebih adil pada data dengan distribusi tidak seimbang.
d. Grafik *loss* dan akurasi selama proses pelatihan turut dianalisis untuk mengevaluasi dinamika pembelajaran model. Grafik ini digunakan untuk mengamati konvergensi, stabilitas proses training, serta mendeteksi potensi *overfitting* atau *underfitting* melalui perbandingan performa pada data latih dan validasi. Informasi ini mendukung interpretasi terhadap hasil akhir pada *test set*.

V SIMPULAN DAN SARAN

5.1. Simpulan

Berdasarkan hasil penelitian dan evaluasi performa model yang telah dilakukan, diperoleh simpulan sebagai berikut:

1. Model klasifikasi *multi-label* tingkat keparahan serangan hama tebu berhasil dibangun menggunakan algoritma LightGBM dan CatBoost pada tiga jenis hama, yaitu penggerek batang, penggerek pucuk, dan kutu perisai. Baik LightGBM maupun CatBoost, terbukti mampu menangani karakteristik data pertanian tebu yang bersifat heterogen, yaitu kombinasi fitur numerik (umur, curah hujan, intensitas serangan), fitur kategorikal (area tanam dan jenis tebu), serta komponen temporal berbasis *lag*.
2. Kedua model menunjukkan performa yang relatif sebanding. Nilai *F1-score macro average* berada pada kisaran 0,59–0,69, sedangkan AUC *macro* berada pada kisaran 0,79–0,86 pada seluruh skema pembagian data. CatBoost sedikit lebih unggul pada beberapa label, khususnya *stem borer* dan kutu perisai, sementara pada *top borer* performa kedua model hampir identik. Perbedaan arsitektur tidak menghasilkan *gap* performa yang ekstrem, namun memberikan variasi kecil yang konsisten.
3. Pada tingkat kelas, kedua model mampu membedakan tingkat keparahan serangan hama dengan baik pada kelas "Tinggi", dengan nilai AUC secara konsisten berada pada kisaran 0,87–0,95 di seluruh skema *splitting*. Sebaliknya, kelas "Sedang" memiliki nilai AUC yang lebih rendah (sekitar 0,70–0,78), mengindikasikan bahwa ambiguitas klasifikasi terutama terjadi pada kelas dengan batas interval yang berdekatan. Temuan ini memperkuat bahwa model lebih mudah mendeteksi kondisi serangan kritis dibandingkan membedakan tingkat sedang.

4. Variasi skema pembagian data (60:20:20, 70:15:15, dan 80:10:10) tidak menunjukkan perbedaan performa yang signifikan. Konsistensi ini mengindikasikan bahwa pendekatan *block-based splitting* yang diterapkan mampu menjaga stabilitas evaluasi dan mengurangi risiko *data leakage* antar blok.
5. Dari sisi efisiensi komputasi, LightGBM memiliki keunggulan pada waktu pelatihan, dengan durasi pelatihan berada pada kisaran kurang dari 3 detik, sedangkan CatBoost membutuhkan waktu pelatihan yang lebih lama (sekitar 17–35 detik). Namun, CatBoost menunjukkan waktu prediksi yang lebih cepat dan stabil (sekitar 0,005–0,016 detik), sehingga lebih unggul pada tahap inferensi ketika model telah siap digunakan.
6. Perbedaan arsitektur dan implementasi berkontribusi terhadap karakter performa dan kemudahan penggunaan model. LightGBM dengan pendekatan *leaf-wise* memberikan fleksibilitas tinggi namun lebih sensitif terhadap konfigurasi *hyperparameter*. Sebaliknya, CatBoost dengan struktur pohon simetris (*oblivious tree*), *ordered boosting*, serta mekanisme auto_class_weights="SqrtBalanced" menunjukkan stabilitas pembelajaran yang lebih konsisten pada data dengan fitur kategorikal dan distribusi kelas tidak seimbang. Selain itu, kedua model mampu menangani *missing value* secara internal tanpa memerlukan imputasi tambahan.
7. Berdasarkan kombinasi hasil performa, stabilitas antar skema *splitting*, serta kemudahan implementasi dalam menangani fitur kategorikal dan *imbalance*, CatBoost direkomendasikan sebagai model yang lebih sesuai untuk mendukung sistem prediksi tingkat keparahan serangan hama tebu pada Perusahaan Gula A, terutama pada skenario operasional yang membutuhkan *pipeline* yang lebih sederhana dan prediksi yang stabil.
8. Hasil prediksi tingkat keparahan serangan hama dapat dimanfaatkan sebagai komponen sistem pendukung keputusan dalam manajemen risiko hama. Informasi prediksi memungkinkan perusahaan mengidentifikasi blok dengan potensi risiko tinggi lebih awal, menetapkan prioritas *monitoring* lapangan, serta mengoptimalkan alokasi sumber daya pengamatan dan pengendalian secara lebih terarah dan berbasis data.

5.2. Saran

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya maupun penerapan hasil penelitian di lingkungan operasional.

1. Penelitian selanjutnya disarankan untuk menambahkan rentang data historis yang lebih panjang serta memperkaya fitur *input*, seperti penambahan variabel lingkungan atau pola musiman yang lebih detail. Hal ini berpotensi meningkatkan kemampuan model dalam menangkap dinamika serangan hama secara jangka panjang.
2. Selain LightGBM dan CatBoost, penelitian berikutnya dapat mengeksplorasi algoritma lain, seperti model *deep learning* berbasis *time series* atau pendekatan *ensemble* lintas model, untuk membandingkan performa prediksi pada kasus *multi-label* dengan data temporal.
3. Penggunaan teknik validasi tambahan, seperti *cross-validation* berbasis waktu (*time series split*), dapat dipertimbangkan untuk memperoleh estimasi performa model yang lebih *robust*, khususnya pada data yang memiliki ketergantungan temporal.
4. Hasil prediksi tingkat keparahan serangan hama yang dihasilkan oleh model dapat diintegrasikan ke dalam sistem pendukung keputusan. Sistem ini dapat dimanfaatkan untuk membantu proses pemantauan risiko, penentuan prioritas wilayah, serta perencanaan inspeksi lapangan secara lebih terstruktur dan berbasis data.

DAFTAR PUSTAKA

Adrian, R., Nasamsir, N., & Meilin, A. (2019). Survei Serangan hama Pada Perkebunan tebu (Saccharum officinarum L.) Di Provinsi jambi. *Jurnal Media Pertanian*, *4*(1), 1. https://doi.org/10.33087/jagro.v4i1.77

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., & Networks, P. (2019). *Optuna: A Next - generation hyperparameter Optimization Framework*. 1–10.

Alimin. (2022). *Pengendalian Tiga Hama Penting Pada Tebu*. https://ditjenbun.pertanian.go.id/pengendalian-tiga-hama-penting-pada-tebu/

BPS Indonesia. (2023). *Produksi Tanaman Perkebunan - Tabel Statistik - Badan Pusat Statistik Indonesia*. https://www.bps.go.id/id/statistics-table/2/MTMyIzI=/produksi-tanaman-perkebunan--ribu-ton-.html

Breiman, L. (2001). Random Forests. *International Journal of Advanced Computer Science and Applications*, *7*(6), 1–33.

Fradzan, R. (2014). *Pengenalan Hama Tebu: Kutu Perisai*. Scribd. https://id.scribd.com/presentation/439800996/Kutu-Perisai-Tebu

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gao, P., Xu, Q., Wen, P., Shao, H., He, Y., & Huang, Q. (2019). *Towards Decision-Friendly AUC : Learning Multi-Classifier with AUC μ*. *37*(6). https://doi.org/https://doi.org/10.1609/aaai.v37i6.25926

Garba, M. L. I., Naroua, H., Kadri, C., Garba, M., & Ali, M. A. (2025). *Multilabel Classification for Predicting Crop Pests in Niger*. *13*(1), 407–415.

Gidiglo, P. D., Njimbouom, S. N., Abdelkader, G. A., Mosalla, S., & Kim, J. D. (2024). Multi-Label Classification for Predicting Antimicrobial Resistance on E. coli. *Applied Sciences (Switzerland)*, *14*(18). https://doi.org/10.3390/app14188225

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Huang, Y., Dong, Y., Huang, W., Guo, J., Hao, Z., & Zhao, M. (2024). *Predicting the Global Potential Suitable Distribution of Fall Armyworm and Its Host Plants Based on Machine Learning Models*. *16*(12). https://doi.org/https://doi.org/10.3390/rs16122060

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Imani, M. (2023). *hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction : A Comparative Analysis*. *11*(6). https://doi.org/https://doi.org/10.3390/technologies11060167

Jain, A. (2024). *A Comprehensive Guide to Ensemble Techniques: Bagging and Boosting*. https://medium.com/@abhishekjainindore24/a-comprehensive-guide-to-ensemble-techniques-bagging-and-boosting-fa276e28da9f

Janiesch, C., Zschech, P., & Heinrich, K. (2022). Machine Learning And Deep Learning. *Elgar Encyclopedia of Technology and Politics*, *31*, 114–118. https://doi.org/https://doi.org/10.1007/s12525-021-00475-2

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3147–3155.

Kumar, S., Sohail, M. A., Jadhav, S., & Gupta, R. K. (2024). *Light Gradient Boosting Machine For Optimizing Crop Maintenance And Yield Prediction In Agriculture*. *6956*(October), 3551–3555. https://doi.org/10.21917/ijsc.2024.0495

Mahesh, P., & Soundrapandiyan, R. (2024). *Yield prediction for crops by gradient-based algorithms*. *Fig 1*, 1–20. https://doi.org/10.1371/journal.pone.0291928

Mckinney, W. (2010). *Data Structures for Statistical Computing in Python*. *1*(Scipy), 56–61.

Muliasari, A. A., & Trilaksono, R. (2020). Insidensi Hama dan Penyakit Utama Tebu (Saccharum officinarum L) di PT PG Rajiwali II Jati Majalengka. *Jurnal Sains Terapan*, *10*(1), 40–52.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics*, *18*(6), 275–285. https://doi.org/10.1002/cem.873

Nadeem, R. M., Jaffar, A., & Saleem, R. M. (2022). IoT and machine learning based stem borer pest prediction. *Intelligent Automation and Soft Computing*, *31*(3), 1377–1392. https://doi.org/10.32604/IASC.2022.020680

Nugroho, A. (2021, October 5). Tekan Impor, Kemenperin Genjot Produksi Industri Gula. *RM.Id*. https://rm.id/baca-berita/ekonomi-bisnis/93883/tekan-impor-kemenperin-genjot-produksi-industri-gula

Nugroho, K. S. (2019). *Confusion Matrix untuk Model Evaluasi pada Supervised Learning*. https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science and Engineering*, *9*(3), 10–20. https://doi.org/10.1109/MCSE.2007.58

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2011). Scikit-learn: Machine Learning in Python. *Environmental Health Perspectives*, *127*(9), 2825–2830. https://doi.org/10.1289/EHP4713

Pituckwanich, W., Hormdee, D., Boonkong, A., Kaewfoongrungsi, P., Tintarasara Na Ratchaseema, M., & Veerachit, V. (2025). The Implementation of a Prediction System for Sugarcane's Destruction Rate From Sugarcane Stem Borer via Hybrid Machine Learning. *IEEE Access*, *13*(March), 45594–45608. https://doi.org/10.1109/ACCESS.2025.3549453

Pramono, S. (2025). Distribution of Sugarcane Shield Scale (Aulacaspis Tegalensis) and Coccinellid Predators Aggregation on Sugarcane Plants. *Plant Protection*, *9*(1), 25–30. https://doi.org/10.33804/pp.009.01.5444

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural*

Information Processing Systems, *2018-Decem*(Section 4), 6638–6648.

Rezk, N. G., Attia, A., El-rashidy, M. A., & El-sayed, A. (2025). *An efficient IoT-based crop damage prediction framework in smart agricultural systems*. 1–15.

Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion Matrix-Based Performance Evaluation Metrics. *African Journal of Biomedical Research*, *November*, 4023–4031. https://doi.org/10.53555/ajbr.v27i4s.4345

Satria, Y., Harmoni, B., Hindrayani, K. M., & Prasetya, D. A. (2025). *Optimizing Categorical Boosting Model with Optuna for Anti-Tuberculosis Drugs Classification*. *7*(2), 401–414.

Shafiuzzaman, M., Islam, S., Bashar, T. M. R., Munem, M., Nahiduzzaman, M., Ahsan, N., & Haider, J. (2025). Enhanced very short-term load forecasting with multi-lag feature engineering and prophet-XGBoost-CatBoost architecture. *Energy*, *335*(June), 137981. https://doi.org/10.1016/j.energy.2025.137981

Silva, V. C., Rocha, M. S., Faria, G. A., Alves, S. F., Junior, X., Oliveira, T. A. De, Patricia, A., & Peixoto, B. (2024). *Boosting algorithms for prediction in agriculture: An application of Feature importance and Feature Selection*. *13*(4), 339–348. https://doi.org/https://doi.org/10.29327/2520355.13.4-31

Simeone, O. (2018). A brief introduction to machine learning for engineers. *Foundations and Trends in Signal Processing*, *12*(3–4), 200–431. https://doi.org/10.1561/2000000102

Sunaryo, & Hasibuan, R. (2003). Perkembangan Populasi Kutu Perisai Aulacaspis Tegalensis Zehntner (Homoptera: Diaspididae) Dan Pengaruh Tingkat Serangannya Terhadap Penurunan Hasil Tebu Di Pt Gunung Madu Plantations, Lampung Tengah. *Jurnal Hama Dan Penyakit Tumbuhan Tropika*, *3*(1), 1–5. https://doi.org/10.23960/j.hptt.131-5

Utami, I. D., Muningsih, R., & Ciptadi, G. (2024). Identifikasi tingkat serangan hama uret (Lepidiota stigma. F) pada tanaman tebu (Saccharum officinarum L) di Kabupaten Sleman. *Jurnal Pengelolaan Perkebunan Vol. 5, No. 1, Maret 2024, Pp. 7-17 ISSN*, *5*(1), 7–17.

Wadhwa, D., & Malik, K. (2024). *A Generalizeble Model For Early Warning Of Crop Diseases Using Environmental And Pest Infestation Data*. 1–34.

https://doi.org/10.2139/ssrn.4924884

Wibawa, T. S., Ningrum, N. K., & Syahreza, A. (2025). *Comparison of CatBoost and LightGBM Models for Air Humidity Prediction*. *9*(3), 803–809.

Wilimitis, D., & Walsh, C. G. (2023). Practical Considerations and Applied Examples of Cross-Validation for Model Development and Evaluation in Health Care: Tutorial. *Jmir Ai*, *2*(1). https://doi.org/10.2196/49023

Younes, L. (2024). Introduction to Machine Learning. *Studies in Computational Intelligence*, *1169*, 51–94. https://doi.org/10.1007/978-981-97-5624-7_2

Zhang, M., & Zhou, Z. (2013). *A Review on Multi-Label Learning Algorithms*. 1–59.

Zhou, L., Zheng, X., Yang, D., Wang, Y., Bai, X., & Ye, X. (2021). *Application of multi-label classification models for the diagnosis of diabetic complications.pdf*.