Rancang Bangun Smart Composter Berbasis IOT dan Machine Learning Dengan Sistem Monitoring Berbasis Firebase dan Flutter

SUB JUDUL

Klasifikasi Kematangan Kompos Organik Menggunakan Metode Decission Tree Berbasiskan Edge Computing Pada Sistem Smart Composter

SKRIPSI

Rasyid Fathoni Syahputra

2207421027

Program Studi Teknik Multimedia Dan Jaringan Jurusan Teknik Informatika Dan Komputer

Politeknik Negeri Jakarta

2026

Klasifikasi Kematangan Kompos Organik Menggunakan Metode Decission Tree Berbasiskan Edge Computing Pada Sistem Smart Composter

SKRIPSI

Dibuat Untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

Rasyid Fathoni Syahputra

2207421027

Program Studi Teknik Multimedia Dan Jaringan

Jurusan Teknik Informatika Dan Komputer

Politeknik Negeri Jakarta

2026

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Rasyid Fathoni Syahputra

NIM : 2207421027

Jurusan/Program Studi : Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

Judul skripsi : Klasifikasi Kematangan Kompos Organik Menggunakan Metode Decission Tree Berbasiskan Edge Computing pada Sistem Smart Composter

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 28 Juli 2026

Yang membuat pernyataan

(Rasyid Fathoni Syahputra)

NIM. 2207421027

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga laporan tugas akhir berjudul "Klasifikasi Kematangan Kompos Organik Menggunakan Metode Decission Tree Berbasiskan Edge Computing pada Sistem Smart Composter" dapat diselesaikan. Laporan ini disusun sebagai salah satu syarat memperoleh gelar D4 pada Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta. Penulis menyadari bahwa penyusunan laporan ini tidak akan terselesaikan tanpa bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Dr. Indra Hermawan S.Kom., M.Kom., selaku Dosen Pembimbing, yang telah memberikan waktu, tenaga, dan pikiran untuk membimbing & mengarahkan penulis dalam penyusunan laporan tugas akhir ini.
2. Orang tua dan keluarga penulis, yang senantiasa memberikan dukungan doa, dan motivasi selama proses studi hingga penyusunan laporan ini.
3. Teman-teman sekelas, rekan smart composter, dan sahabat sekampus, yang selalu memberikan bantuan, semangat, dan kerja sama dalam menyelesaikan laporan ini.

Akhir kata, penulis berharap semoga Tuhan Yang Maha Esa senantiasa memberikan balasan yang berlimpah atas segala kebaikan dan bantuan yang telah diberikan oleh semua pihak. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun pihak lain yang membacanya.

ABSTRAK

Pengelolaan sampah organik rumah tangga melalui pengomposan umumnya masih dilakukan secara manual dengan mengandalkan pengamatan visual yang subjektif terhadap indikator fisik seperti warna, tekstur, dan bau untuk menentukan kematangan kompos. Subjektivitas tersebut meningkatkan risiko evaluasi yang tidak konsisten serta kegagalan proses. Penelitian ini mengusulkan sistem smart composter yang mengintegrasikan sensor Internet of Things (IoT) dengan model machine learning berbasis pohon keputusan (CatBoost, Random Forest, dan XGBoost) yang berjalan pada perangkat edge computing untuk mengklasifikasikan kematangan kompos berdasarkan nilai Germination Index (GI). Dua arsitektur prediksi dirancang dan dibandingkan secara head-to-head, yaitu Arsitektur A (Single-Stage) yang memetakan fitur sensor mentah langsung ke nilai GI, dan Arsitektur B (Two-Stage Soft-Sensor) yang terlebih dahulu mengestimasi empat parameter laboratorium yang sulit diukur langsung (rasio C/N, Electrical Conductivity, Total Nitrogen, dan Nitrate) sebagai keluaran soft-sensor antara sebelum memprediksi GI. Kedua arsitektur dievaluasi pada lima skema augmentasi data (baseline, C-Mixup, GMM, CTGAN, SMOTE) dan tiga metode optimasi hyperparameter (Bayesian Optimization, Random Search, Grid Search) menggunakan skema nested cross-validation berbasis batch pada 452 data asli dari dataset kompos publik, membentuk 270 kombinasi eksperimen. Hasil menunjukkan Arsitektur A secara signifikan mengungguli Arsitektur B (paired t-test, $p = 0{,}00022$). Konfigurasi terbaik, yaitu CatBoost dengan augmentasi GMM pada Arsitektur A yang dioptimalkan menggunakan Random Search, memperoleh RMSE = 17,729, MAE = 13,342, dan $R^2 = 0{,}628$. Saat diimplementasikan pada Raspberry Pi 4B dan diuji pada tujuh batch hold-out independen (52 sampel), model mencapai $R^2 = 0{,}745$, akurasi klasifikasi kematangan 75,0%, serta latensi inferensi rata-rata hanya 1,528 ms, jauh di bawah ambang batas real-time 50 ms. Temuan ini menunjukkan bahwa model tree-based single-stage mampu memberikan klasifikasi kematangan kompos yang objektif dan berlatensi rendah, sesuai untuk implementasi real-time pada perangkat edge dengan sumber daya terbatas

***Kata kunci**: smart composter, Internet of Things, edge computing, machine learning, Germination Index, CatBoost*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang

Berdasarkan data Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) tahun 2025, komposisi sampah di Indonesia didominasi oleh sisa makanan sebesar 40,61%, diikuti oleh plastik sebesar 20,53%, kayu/ranting sebesar 13,22%, dan kertas/karton sebesar 11,38% (Kementerian Lingkungan Hidup, 2025). Data tersebut menunjukkan bahwa sampah organik masih menjadi komponen terbesar dalam kumpulan sampah nasional. Oleh karena itu, pengelolaan sampah organik, khususnya di tingkat rumah tangga, perlu menjadi perhatian utama untuk mengurangi beban Tempat Pemrosesan Akhir (TPA) serta menekan dampak lingkungan yang ditimbulkan akibat proses dekomposisi sampah organik.

Salah satu metode yang umum digunakan untuk mengolah sampah organik adalah pengomposan, yaitu proses penguraian bahan organik secara biologis dengan mengendalikan aktivitas mikroorganisme hingga menghasilkan kompos yang matang yang dapat dimanfaatkan kembali untuk keperluan pertanian dan perkebunan. Meskipun pengomposan menjadi solusi yang berkelanjutan, keberhasilannya sangat bergantung pada beberapa parameter utama, antara lain suhu, kelembapan, pH, rasio karbon terhadap nitrogen (C/N), aerasi, serta kemunculan gas hasil dekomposisi seperti amonia yang menjadi indikator krusial kualitas proses. Li et al. (2023) menjelaskan bahwa parameter-parameter tersebut secara langsung memengaruhi kecepatan dekomposisi dan tingkat kematangan kompos, sementara proses pengomposan berbasis sensor umumnya dipantau melalui parameter fisik yang mencakup suhu, kelembapan, dan pH (Putri et al., 2025). Apabila parameter tersebut tidak berada pada kondisi optimal, proses pengomposan berpotensi menghasilkan kompos yang belum stabil serta belum mencapai tingkat kematangan yang memadai, padahal kematangan kompos merupakan salah satu indikator utama dalam menentukan kelayakan produk akhir karena berpengaruh terhadap efektivitas pemanfaatannya di lahan pertanian.

Namun, praktik di lapangan saat ini, seperti yang ditemukan di Bank Sampah Rumah Harum, Depok, masih dilakukan secara manual dan kurang termonitor dengan baik. Pengolahan yang mengandalkan perkiraan fisik dan pengamatan subjektif tersebut meningkatkan risiko ketidakstabilan proses dekomposisi serta memicu tingginya kegagalan deteksi masalah pada tumpukan kompos. Untuk mengatasi kendala tersebut, pemanfaatan teknologi Internet of Things (IoT) hadir menawarkan solusi pemantauan objektif secara real-time melalui penerapan arsitektur IoT terdistribusi yang dibagi menjadi empat layer utama, yaitu Sensor Acquisition Node, Edge Computing Processor berbasis Raspberry Pi 4B, Cloud Backend Database Firebase, dan User Application Interface, sehingga data kondisi fisik dari tumpukan sampah dapat dikumpulkan, diolah, dan dikirimkan secara kontinu tanpa perlu kehadiran operator secara fisik di area tumpukan.

Umumnya, kematangan kompos masih dinilai melalui sifat fisiknya. Seperti warna, tekstur, bau, dan kestabilan suhu. Putri et al. (2025) mengatakan bahwa indikator fisik tersebut merupakan pendekatan utama yang digunakan untuk mengevaluasi kondisi kompos, meskipun penilaian ini cukup subjektif dan hampir sepenuhnya bergantung pada pengalaman pengguna. Akibatnya, evaluasi kematangan kompos bersifat tidak konsisten, terutama pada skala rumah tangga atau jika pengguna tidak memiliki pengalaman teknis yang memadai.

Perkembangan Internet of Things (IoT) membuka peluang untuk meningkatkan pemantauan proses pengomposan secara lebih terukur dan presisi. Rais dan Rokhmah (2025) menjelaskan bahwa IoT memungkinkan objek fisik saling terhubung melalui internet untuk pemantauan secara real-time. Dalam konteks pengomposan, Putri et al. (2025) serta Mujiyanti et al. (2022) menunjukkan bahwa sensor seperti suhu, kelembapan, pH, dan gas dapat diintegrasikan dengan mikrokontroler atau perangkat komputasi untuk memantau kondisi kompos secara langsung. Dengan demikian, pengguna dapat memperoleh data proses secara kontinu tanpa sepenuhnya bergantung pada pengamatan manual.

Namun, penelitian *smart composter* berbasis IoT yang ada di Indonesia masih dominan pada fungsi pemantauan dan kontrol berbasis ambang batas, bukan secara otomatis mengklasifikasikan kematangan kompos (Rais & Rokhmah, 2025; Putri et al., 2025; Mujiyanti et al., 2022). Di sisi lain, penelitian machine

learning untuk prediksi kematangan kompos umumnya masih dievaluasi secara offline pada data historis dan belum terintegrasi dengan sistem IoT real-time maupun diuji pada perangkat edge computing (Li et al., 2023; Shi et al., 2025). Bahkan pada dataset publik yang sama dengan penelitian ini, Mullick et al. (2025) mencapai akurasi prediksi yang sangat tinggi (R^2 = 0,998-0,999), namun pendekatan tersebut memprediksi maturity Score menggunakan T-Value dan GI(%) itu sendiri sebagai bagian dari fitur input, bukan memprediksi GI hanya dari sensor fisik dasar yang tersedia pada perangkat IoT. Dengan demikian, terdapat gap implementasi yang belum terjawab: belum ada penelitian yang mengintegrasikan sistem IoT real-time dengan model machine learning untuk memprediksi kematangan kompos hanya dari sensor fisik, sekaligus diuji kelayakannya pada perangkat edge computing.

Integrasi antara sistem pemantauan IoT dan model *machine learning* berpotensi menghasilkan *smart composter* yang tidak hanya memantau kondisi kompos, tetapi juga mampu memprediksi tingkat kematangan kompos secara lebih objektif dan berbasis data. Dengan menggunakan data real-time dari sensor, sistem dapat mengenali pola perubahan kondisi selama proses dekomposisi berlangsung. Data tersebut kemudian diproses menggunakan algoritma *machine learning* untuk mengidentifikasi parameter proses yang berkorelasi dengan nilai *Germination Inde*x (GI), sehingga sistem dapat memberikan hasil prediksi kematangan kompos yang lebih terukur dibandingkan penilaian manual.

Dengan mempertimbangkan kondisi-kondisi tersebut, penelitian ini mengusulkan sistem smart composter yang menerapkan sensor IoT dengan model *edge computing* berbasis algoritma *tree-based* (*CatBoost*, *Random Forest*, dan *XGBoost*). Penelitian ini merancang dan membandingkan dua arsitektur prediksi secara head-to-head, yaitu arsitektur A (Single-Stage), yang memetakan fitur sensor langsung ke nilai GI(%), dan arsitektur B (Two-Stage Soft-Sensor), yang terlebih dahulu mengestimasi empat parameter laboratorium yang sulit diukur langsung oleh sensor fisik (rasio C/N, *Electrical Conductivity*/EC, Total Nitrogen/TN, dan Nitrate) sebagai representasi antara, sebelum memprediksi nilai GI akhir. Ketiga algoritma tersebut dibandingkan pada kedua arsitektur, dikombinasikan dengan enam skema augmentasi data dan tiga metode

hyperparameter optimization, untuk memperoleh kombinasi model dengan performa dan efisiensi komputasi terbaik.

Pemilihan pendekatan *edge computing* dalam penelitian ini didasarkan pada kebutuhan sistem pengomposan rumah tangga yang memerlukan respons lokal secara cepat dan berkelanjutan. Pemrosesan model secara langsung pada perangkat tertanam, seperti Raspberry Pi, memungkinkan sistem tetap beroperasi tanpa bergantung pada konektivitas internet yang tidak selalu stabil di lingkungan rumah tangga. Selain itu, model berbasis pohon keputusan (*CatBoost, Random Forest*, dan *XGBoost*) tidak memerlukan konversi format khusus seperti TensorFlow Lite, sehingga dapat diimplementasikan langsung pada perangkat edge menggunakan pustaka Python standar. Hal ini memungkinkan tercapainya latensi inferensi yang rendah tanpa kebutuhan transmisi data ke cloud. Penelitian ini juga bertujuan untuk mengevaluasi kinerja model dalam hal efisiensi komputasi pada perangkat edge dengan sumber daya terbatas, tanpa penurunan akurasi yang signifikan.

Penelitian ini relevan karena mengintegrasikan beberapa aspek yang belum banyak dikaji secara simultan dalam penelitian terdahulu, yaitu integrasi IoT dan machine learning, perbandingan langsung dua arsitektur prediksi (single-stage dan two-stage soft sensor), evaluasi sistematis kombinasi augmentasi data dan metode hyperparameter optimization, serta pengujian implementasi pada perangkat edge computing. Oleh karena itu, penelitian berjudul "Klasifikasi Kematangan Kompos Organik Menggunakan Metode Decission Tree Berbasis Edge Computing pada Sistem Smart Composter" diharapkan dapat memberikan kontribusi dalam pengembangan sistem pengomposan cerdas yang lebih efisien, objektif, dan aplikatif.

1.2. Perumusan Masalah

Berdasarkan latar belakang, permasalahan yang diangkat dalam penelitian ini dapat dirumuskan sebagai berikut.

1. Bagaimana merancang dan membandingkan Arsitektur A (*Single-Stage*) dan Arsitektur B (*Two-Stage Soft-Sensor*) menggunakan algoritma tree-based (*CatBoost, Random Forest, XGBoost*) untuk memprediksi tingkat

kematangan kompos (*Germination Index*) berbasis data sensor proses pengomposan?

2. Bagaimana mengimplementasikan dan menguji model prediksi GI terbaik pada sistem smart composter berbasis Raspberry Pi sehingga dapat memberikan informasi tingkat kematangan kompos secara mendekati real-time kepada pengguna?
3. Bagaimana menganalisis dan mengevaluasi pengaruh arsitektur, algoritma, skema augmentasi data, dan metode hyperparameter optimization terhadap akurasi prediksi kematangan kompos pada berbagai kategori GI (Fitotoksik, Rendah, Sedang, Matang)?

1.3. Batasan Masalah

Agar penelitian lebih fokus dan terarah, ditetapkan batasan masalah sebagai berikut.

1. Penelitian ini dibatasi pada perancangan dan perbandingan arsitektur A (*Single-Stage*) dan arsitektur B (*Two-Stage Soft-Sensor*) menggunakan tiga algoritma *tree-based* (CatBoost, Random Forest, dan XGBoost), lima skema augmentasi data (baseline, C-Mixup, GMM, CTGAN, dan SMOTE), serta tiga metode hyperparameter optimization (Bayesian Optimization, Random Search, dan Grid Search). Seluruh kombinasi dievaluasi menggunakan skema *nested cross-validation* berbasis *batch* per waktu pengomposan. Penelitian ini tidak mencakup eksplorasi algoritma maupun arsitektur *machine learning* di luar pendekatan *tree-based* yang digunakan.
2. Penelitian ini dibatasi pada pemantauan parameter proses dan implementasi model prediksi tingkat kematangan kompos pada perangkat keras berupa Raspberry Pi dan modul sensor, dengan rancangan mekanik dan elektronik yang mengikuti sistem kelompok yang telah ditetapkan. Penelitian tidak mencakup perancangan sistem kontrol otomatis, seperti motor pengaduk, motor pencacah, dan pompa air, maupun pengembangan aspek mekanik dan elektronik perangkat. Selain itu, penelitian ini juga tidak membahas degradasi performa sensor, *sensor drift* akibat paparan lingkungan, maupun prosedur rekalibrasi berkala. Seluruh evaluasi model dilakukan dengan asumsi bahwa

sensor berada dalam kondisi terkalibrasi dan mampu menghasilkan pembacaan yang akurat selama proses pengujian.

3. Penelitian ini dibatasi pada evaluasi akurasi model menggunakan data sekunder dari *Compost Maturity and Emission Monitoring Dataset* (Mullick, 2023), yang terdiri atas 452 baris data asli dari 46 *batch* unik tanpa menyertakan data sintetis SMOGN bawaan dataset. Evaluasi dilakukan menggunakan nilai *Germination Index* (GI) yang telah tersedia sebagai acuan penentuan kategori kematangan kompos (Fitotoksik, Rendah, Sedang, dan Matang). Penelitian ini tidak mencakup pengumpulan data primer melalui eksperimen pengomposan mandiri maupun pengujian laboratorium tambahan untuk memvalidasi tingkat kematangan kompos secara independen.

1.4. Tujuan dan Manfaat

Tujuan penelitian ini adalah sebagai berikut:

1. Merancang dan mengimplementasikan model two-stage machine learning untuk mengklasifikasi tingkat kematangan kompos (GI) berbasis data sensor.
2. Menganalisis pengaruh penggunaan beberapa algoritma pembelajaran mesin (CatBoost, Random Forest, XGBoost) serta strategi augmentasi data terhadap akurasi klasifikasi GI pada dataset kompos publik.
3. Mengimplementasikan model klasifikasi GI terbaik pada prototipe *smart composter* berbasis Raspberry Pi dan mengukur kinerja inferensi model (akurasi dan latensi) untuk memastikan kelayakan deployment real-time.

Manfaat Penelitian adalah sebagai berikut:

1. Manfaat praktis
   a) Menyediakan solusi teknologi untuk meningkatkan efektivitas pengelolaan sampah organik melalui pemantauan kondisi kompos dan klasifikasi kematangan yang terukur dan berbasis data.
   b) Membantu pengguna *smart composter* dalam mengambil keputusan yang lebih objektif terkait waktu panen kompos tanpa harus sepenuhnya bergantung pada pengalaman subjektif atau uji laboratorium.

2. Manfaat akademis
   a) Memberikan kontribusi pada pengembangan ilmu pengetahuan di bidang Internet of Things dan pembelajaran mesin, khususnya penerapan two-stage machine learning dan ensemble pada kasus klasifikasi kematangan kompos.
   b) Menjadi referensi bagi penelitian lanjutan terkait smart composting dengan strategi augmentasi data tabular, dan analisis kinerja berbagai algoritma machine learning untuk klasifikasi GI.
3. Manfaat bagi masyarakat
   a) Mendukung upaya pengurangan volume sampah yang dikirim ke TPA melalui pengolahan sampah organik di tingkat sumber dengan bantuan teknologi pemantauan dan klasifikasi yang lebih cerdas.
   b) Meningkatkan kesadaran mengenai pentingnya pemantauan dan klasifikasi proses dalam menghasilkan kompos berkualitas sehingga dapat dimanfaatkan secara aman dan berkelanjutan.

1.5. Sistematika Penulisan

Laporan Tugas Akhir ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Berisi latar belakang penelitian, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan laporan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan landasan teori dan penelitian terdahulu yang relevan dengan topik pengomposan, indikator kematangan kompos (termasuk GI dan T-Value), algoritma machine learning berbasis pohon (CatBoost, Random Forest, XGBoost), metode hyperparameter optimization (Grid Search, Random Search, Bayesian Optimization) dan konsep soft sensor, teknik augmentasi data tabular, serta gap penelitian yang mendasari penggunaan model two-stage machine learning pada penelitian ini.

3. BAB III METODE PENELITIAN

Menjelaskan rancangan penelitian, perancangan dua arsitektur prediksi (*Single-Stage* dan *Two-Stage Soft-Sensor*) pada smart composter berbasis IoT, rekayasa fitur, skema augmentasi data, metode hyperparameter optimization, skema evaluasi nested cross-validation, pendekatan ensemble, serta metode pengujian dan analisis kinerja model (termasuk uji deployment pada Raspberry Pi).

4. BAB IV HASIL DAN PEMBAHASAN

Memaparkan hasil analisis kebutuhan, perancangan, dan implementasi model two-stage machine learning, meliputi hasil eksperimen 270 kombinasi (arsitektur, algoritma, augmentasi, dan metode HPO), hasil optimasi ensemble, serta konversi model untuk deployment. Bab ini juga menyajikan hasil pengujian model (evaluasi R^2, MAE, RMSE secara global dan per kategori GI, serta analisis SHAP), hasil pengujian latensi inferensi pada Raspberry Pi, serta analisis kompromi (trade-off) antara akurasi model dan performa komputasi edge dari beberapa skenario yang diuji.

5. BAB V PENUTUP

Berisi simpulan yang menjawab rumusan masalah dan tujuan penelitian berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, serta saran bagi pengembangan penelitian selanjutnya, termasuk pengumpulan dataset mandiri, pengujian lapangan, dan penyempurnaan model.

BAB II

TINJAUAN PUSTAKA

2.1. Pengomposan dan Kematangan Kompos

2.1.1. Definisi dan Proses Pengomposan

Kompos merupakan produk akhir yang stabil dan aman dari proses dekomposisi bahan organik oleh mikroorganisme, yang dihasilkan melalui pengomposan (composting) sebagai metode pengolahan sampah organik. Proses ini mengubah bahan mentah menjadi produk yang tidak lagi bersifat toksik dan siap dimanfaatkan kembali, misalnya sebagai pupuk (Li et al., 2023). Pengomposan berlangsung melalui beberapa fase yang ditandai oleh perubahan suhu secara khas, yaitu fase mesofilik pada tahap awal, fase termofilik saat aktivitas mikroba mencapai puncaknya, fase pendinginan (cooling), serta fase pematangan (maturation) ketika suhu kembali stabil mendekati suhu lingkungan.

Kualitas dan kecepatan proses pengomposan sangat dipengaruhi oleh parameter yang dipantau, terutama suhu, kelembapan, dan pH. Rais dan Rokhmah (2025), dalam pengembangan Smart Composting Bin berbasis IoT untuk sampah organik rumah tangga, menunjukkan bahwa sistem yang mampu menjaga suhu pada rentang fase termofilik (50–60°C) dan kelembapan pada kisaran ideal (55–65%), serta melakukan penyesuaian otomatis seperti aktivasi pompa air ketika kelembapan turun di bawah 40%, terbukti dapat mempercepat proses dekomposisi dan menghasilkan kompos matang lebih cepat dibandingkan metode konvensional tanpa pemantauan. Selain itu, parameter lain seperti pH, electrical conductivity (EC), rasio C/N, kandungan bahan organik (organic matter/OM), serta total nitrogen (TN) juga berperan dalam memengaruhi laju degradasi bahan organik dan perlu dikendalikan untuk mengoptimalkan proses pengomposan (Li et al., 2023).

2.1.2. Indikator Kematangan Kompos

Kematangan kompos merupakan kondisi ketika bahan organik telah terdekomposisi secara sempurna dan stabil sehingga aman digunakan tanpa menimbulkan efek negatif pada tanaman. Kompos yang belum matang dapat bersifat fitotoksik, menghambat perkecambahan benih, serta masih mengandung senyawa yang belum terurai sempurna sehingga berpotensi merugikan apabila diaplikasikan secara langsung (Shi et al., 2025).

Kematangan kompos dapat dievaluasi melalui indikator fisik, kimia, dan biologis. Indikator fisik meliputi warna cokelat kehitaman, tekstur remah, tidak berbau busuk, serta suhu yang stabil mendekati suhu lingkungan, meskipun penilaian ini bersifat subjektif dan bergantung pada pengalaman pengamat (Putri et al., 2025). Indikator kimia mencakup rasio C/N di bawah 20:1 atau mengalami penurunan signifikan dari kondisi awal, pH pada rentang netral hingga sedikit basa, electrical conductivity (EC) yang mencerminkan konsentrasi garam terlarut dan potensi fitotoksisitas, serta kandungan amonia yang menurun seiring proses pematangan. Sebagai contoh, Putri et al. (2025) melaporkan bahwa kompos jerami dengan rasio C/N sebesar 18,13 telah memenuhi standar SNI 19-7030-2004 (10–20), yang mengindikasikan kematangan kompos.

Indikator biologis dianggap paling akurat karena secara langsung mengukur respons organisme hidup terhadap kompos (Shi et al., 2025). Salah satu metode yang umum digunakan adalah Germination Index (GI), yang pertama kali dikembangkan oleh Zucconi et al. (1981) sebagai bioassay sederhana untuk mengevaluasi fitotoksisitas bahan organik. Metode ini menggunakan biji cress (Lepidium sativum L.) yang diinkubasi selama 24 jam pada suhu 27°C dalam kondisi gelap. Nilai GI dihitung berdasarkan perkalian persentase perkecambahan dan pertumbuhan akar relatif terhadap kontrol, sehingga sensitif terhadap berbagai tingkat toksisitas.

Zucconi et al. (1981) juga menunjukkan bahwa efek ekstrak kompos berubah seiring tahap dekomposisi, yaitu bersifat menghambat pada kompos belum matang dan menjadi stimulatif pada kompos matang, bahkan dapat melampaui kontrol. Oleh karena itu, nilai GI di atas 100% diinterpretasikan sebagai efek stimulatif. Secara umum, kompos dinyatakan matang dan tidak

fitotoksik jika nilai GI > 80%, nilai 50–80% menunjukkan fitotoksisitas sedang, sedangkan nilai < 50% menunjukkan kompos belum matang (Shi et al., 2025). Selain GI, indikator lain seperti T-value juga digunakan untuk mengevaluasi kematangan kompos secara lebih komprehensif melalui kombinasi parameter kimia, seperti rasio C/N dan kandungan amonia (Li et al., 2023).

2.2. Penelitian Terdahulu

Beberapa penelitian telah dilakukan terkait penerapan machine learning untuk memprediksi kematangan kompos maupun penerapan teknologi IoT pada sistem pengomposan. Perbandingan penelitian-penelitian utama yang menjadi rujukan dan pembanding langsung bagi penelitian ini ditampilkan pada Tabel

Tabel 2.1 *Penelitian Terkait*

No	Peneliti (Tahun)	Fokus Penelitian	Metode	Hasil Utama
1	**Li et al. (2023)**	Prediksi kematangan (GI dan T-value) kompos limbah hijau (green waste)	Perbandingan 4 model: Extra Trees, Gradient Boosting, MLP, KNN; analisis SHAP	Extra Trees terbaik: R^2 = 0,928 (GI), R^2 = 0,957 (T-value)
2	**Shi et al. (2025)**	Prediksi GI pada kompos kotoran ternak (manure)	Perbandingan 6 model: RF, ET, XGBoost, GBDT, BPNN, MLP; analisis SHAP; validasi eksperimen aktual	RF terbaik: R^2 = 0,937 (RMSE = 7,261); ET: R^2 = 0,904; MRE validasi = 32,43%
3	**Rais & Rokhmah (2025)**	Smart Composting Bin berbasis IoT untuk sampah organik rumah tangga	Sensor suhu, kelembapan udara, kelembapan kompos, gas; platform Blynk; kontrol otomatis pompa air	Suhu termofilik terjaga 50–60°C, kelembapan ideal 55–65%; dekomposisi lebih cepat dibanding metode konvensional
4	**Putri et al. (2025)**	Monitoring suhu, kelembapan, dan pH pada kompos jerami padi	ESP32 + sensor DHT22, DS18B20, soil moisture, soil pH; visualisasi via platform Antares	Suhu puncak 56,7°C (hari ke-7); C/N akhir = 18,13, sesuai SNI 19-7030-2004

5	**Mujiyanti et al. (2022)**	Monitoring parameter pengomposan berbasis mikrokontroler	ESP32 + DHT11, ph sensor, aktuator (motor DC, hair dryer)	Parameter proses termonitor secara real-time
6	**Mullick et al. (2025)**	Prediksi maturity Score pada dataset publik yang sama dengan penelitian ini	Augmentasi SMOTE (452→1.314 baris); RF, XGBoost, Gradient Boosting, KNN, Linear Regression; weighted ensemble; SHAP	R^2 = 0,998–0,999, MAE = 0,13–0,79; T-Value dan GI(%) sebagai prediktor paling berpengaruh (SHAP +11,78 dan +3,48)
7	**Khandakar et al. (2025)**	Prediksi kematangan kompos dan monitoring emisi gas berbasis sensor	Sensor gas (CO_2, CH_4, SO_2, NO_2) + suhu/kelembapan; standardisasi fitur; feature importance XGBoost; 6 model ML; LIME dan SHAP	XGBoost dan CatBoost terbaik: R^2 = 0,9912; MAE = 1,1845 (XGBoost); validasi eksternal 36 sampel
8	**Shi et al. (2026)**	Optimasi multi-objektif pengomposan kotoran ternak (kehilangan nutrien, emisi GRK, degradasi antibiotik, bioavailabilitas logam berat)	Neural network multilayer (EcoCompost); analisis SHAP dan partial dependence; validasi eksperimen independen	R^2 = 0,87-0,93; rasio C/N sebagai prediktor dominan kehilangan karbon/nitrogen (hubungan non-linier); durasi pengomposan dominan untuk emisi GRK dan degradasi antibiotik
9	**Zou et al. (2026)**	Evaluasi kematangan kompos in-situ berbasis hyperspectral imaging (HSI) dan machine learning	HSI + PCA + Integrated Maturity Score (IMS); GridSearchCV (grid search lebih fleksibel dari random search); ANN, XGBoost, Random Forest, SVM	Klasifikasi: recall 96,30% (RF) dan 98,72% (XGBoost); Regresi maturity: R^2 = 0,917 (XGBoost) dan 0,886 (RF)

10	**Liu et al. (2026)**	Prediksi emisi amonia kumulatif pada pengomposan lumpur limbah (sewage sludge) menggunakan machine learning multi-tahap	Model bertahap sesuai fase pengomposan; Bayesian Optimization + GroupKFold CV; SVR, MLP, Extra Trees, CatBoost, Gaussian Process Regression; analisis SHAP dan partial dependence	R^2 = 0,85-0,91 pada test set; durasi dan aerasi sebagai faktor dominan emisi amonia pada fase mesofilik-termofilik; CatBoost dan Extra Trees rentan overfitting akibat Bayesian Optimization

2.2.1. Gap Penelitian

Berdasarkan tinjauan penelitian terdahulu, teridentifikasi beberapa gap penelitian sebagai berikut.

1. Gap integrasi ML dan IoT real-time. Penelitian machine learning untuk prediksi kematangan kompos (Li et al., 2023; Shi et al., 2025; Shi et al., 2026; Mullick et al., 2025) menunjukkan akurasi tinggi ($R^2 > 0,9$), namun umumnya belum terintegrasi dengan sistem IoT real-time. Model masih digunakan secara offline setelah proses pengumpulan data, sehingga belum mampu memberikan prediksi secara kontinu selama proses pengomposan berlangsung.
2. Gap jenis bahan organik. Studi sebelumnya umumnya berfokus pada jenis bahan spesifik, seperti green waste (Li et al., 2023) dan manure (Shi et al., 2025; Shi et al., 2026), sementara sampah organik rumah tangga yang bersifat heterogen belum banyak diteliti. Oleh karena itu, validasi model menggunakan dataset publik yang merepresentasikan keragaman bahan organik menjadi langkah awal penting sebelum implementasi pada kondisi nyata di lingkungan rumah tangga.
3. Gap kemampuan prediksi pada sistem IoT. Penelitian smart composter berbasis IoT di Indonesia (Rais & Rokhmah, 2025; Putri et al., 2025; Mujiyanti et al., 2022) telah mampu melakukan pemantauan parameter secara real-time, namun belum mengintegrasikan model ML untuk prediksi kematangan. Sistem yang dikembangkan masih terbatas pada fungsi monitoring tanpa kemampuan prediktif.

4. Gap ketergantungan pada fitur non-sensor. Mullick et al. (2025), meskipun mencapai $R^2 > 0,99$, menggunakan parameter seperti T-value dan GI sebagai bagian dari fitur input. Hal ini menunjukkan bahwa performa model lebih merefleksikan konsistensi antar-indikator kematangan yang telah diukur, bukan kemampuan prediksi berbasis data sensor lapangan. Hingga saat ini, masih terbatas penelitian yang memprediksi GI hanya dari parameter sensor fisik dasar, seperti suhu, kelembapan, pH, dan amonia, tanpa melibatkan parameter laboratorium pada tahap inferensi.
5. Gap integrasi end-to-end. Belum banyak penelitian yang mengintegrasikan secara komprehensif sistem IoT real-time dengan model ML dalam satu sistem utuh untuk prediksi kematangan kompos pada sampah organik rumah tangga. Integrasi ini penting untuk menghasilkan sistem yang tidak hanya memantau parameter, tetapi juga mampu memberikan klasifikasi kematangan secara otomatis dan berkelanjutan.

Berdasarkan gap tersebut, penelitian ini bertujuan merancang smart composter yang mengintegrasikan sistem IoT real-time dengan model klasifikasi berbasis machine learning. Sistem yang dikembangkan tidak hanya memantau parameter proses (suhu, kelembapan, pH, dan amonia) secara kontinu melalui sensor, tetapi juga memberikan klasifikasi tingkat kematangan kompos menggunakan pendekatan two-stage machine learning yang dilatih pada dataset publik, tanpa memerlukan parameter laboratorium saat proses inferensi. Dengan demikian, sistem diharapkan mampu meningkatkan efisiensi, objektivitas, dan keterukuran dalam pengelolaan kompos.

2.3. Algoritma Machine Learning Berbasis Pohon

Hubungan antara parameter proses pengomposan dan tingkat kematangannya bersifat nonlinier serta sulit dimodelkan menggunakan persamaan fisika konvensional. Machine learning menawarkan pendekatan alternatif dengan kemampuan memodelkan hubungan tersebut secara langsung dari data tanpa memerlukan formulasi matematis eksplisit. Mengacu pada temuan dalam Subbab 2.2, di mana algoritma berbasis pohon keputusan (*Extra Trees*, *Random Forest*, *XGBoost*, dan *Gradient Boosting*) secara konsisten menunjukkan kinerja terbaik

dalam tugas regresi kematangan kompos, penelitian ini berfokus pada perbandingan tiga algoritma *tree-based*, yaitu *CatBoost*, *Random Forest*, dan *XGBoost*. Ketiga algoritma tersebut dipilih karena kemampuannya dalam menangani data tabular berukuran kecil hingga menengah dengan performa tinggi, serta mendukung pembentukan model ensemble pada tahap analisis lanjutan.

2.3.1. Random Forest

Random Forest bekerja dengan logika ensemble berbasis *decision tree*, yaitu membangun sejumlah pohon secara acak menggunakan subset fitur dan data yang berbeda melalui mekanisme bagging, kemudian menggabungkan hasil prediksinya untuk meningkatkan stabilitas dan akurasi (Breiman, 2001). Pada prediksi GI untuk kompos manure, Shi et al. (2025) melaporkan bahwa Random Forest menunjukkan performa terbaik dengan R^2 = 0,937 dan RMSE = 7,261, berdasarkan analisis feature importance durasi pengomposan, total nitrogen (TN), dan electrical conductivity (EC) sebagai parameter paling berpengaruh terhadap kematangan kompos. Mullick et al. (2025) juga menguji Random Forest sebagai salah satu dari enam kandidat model awal. Namun, model tersebut tidak termasuk tiga model dengan performa paling konsisten, sehingga tidak diikutsertakan dalam ensemble final. Hal ini menunjukkan bahwa performa tinggi Random Forest pada kompos manure tidak selalu dapat digeneralisasi ke seluruh dataset dan skema pembagian data.

2.3.2. XGBoost

XGBoost (Extreme Gradient Boosting) merupakan pengembangan dari algoritma Gradient Boosting yang dioptimalkan dari sisi kecepatan komputasi dan regularisasi untuk mencegah overfitting (Chen & Guestrin, 2016). Model dibangun secara bertahap, di mana setiap pohon baru berfungsi memperbaiki residual error dari pohon sebelumnya melalui pendekatan gradient descent pada fungsi loss. Pada dataset yang sama dengan penelitian ini, Mullick et al. (2025) menempatkan XGBoost sebagai salah satu dari tiga model paling konsisten, bersama Gradient Boosting dan Linear Regression, dengan bobot kontribusi pada

ensemble akhir sebesar 0,153 yang dioptimasi berdasarkan inverse-RMSE pada performa validasi. Performa tinggi XGBoost pada pemodelan kematangan kompos juga didukung oleh Shi et al. (2025), yang menempatkannya sejajar dengan Random Forest dan Extra Trees sebagai kandidat model berbasis pohon dengan performa yang baik.

2.3.3. CatBoost

CatBoost (Categorical Boosting) adalah algoritma gradient boosting yang dikembangkan Dorogush et al. (2018) dengan dua inovasi utama dibanding algoritma gradient boosting konvensional. Pertama, CatBoost menerapkan ordered boosting, yaitu skema pelatihan yang menghitung residual setiap sampel menggunakan model yang dilatih dari sampel-sampel lain saja, untuk mengurangi prediction shift atau bias target leakage yang muncul pada gradient boosting standar akibat penggunaan target statistik dari sampel yang sama pada proses pelatihan dan evaluasi. Kedua, CatBoost menyediakan penanganan fitur kategorikal secara native melalui teknik ordered target statistics, sehingga tidak memerlukan encoding manual seperti one-hot atau label encoding yang umum diperlukan pada algoritma gradient boosting lain.

Selain dua inovasi tersebut, CatBoost menggunakan oblivious decision tree (pohon simetris) sebagai base learner, yaitu struktur pohon di mana seluruh node pada level kedalaman yang sama menggunakan kriteria split fitur dan threshold yang identik. Struktur ini berbeda dari pohon keputusan konvensional yang memungkinkan setiap node memiliki kriteria split berbeda, sehingga oblivious tree pada CatBoost mempercepat proses inferensi karena strukturnya dapat direpresentasikan sebagai tabel lookup, sekaligus berperan sebagai bentuk regularisasi implisit yang mengurangi risiko overfitting dibanding pohon keputusan asimetris pada Random Forest maupun XGBoost.

Dari sisi penerapan pada domain pengomposan, Liu et al. (2026) melaporkan bahwa CatBoost merupakan salah satu model dengan performa terbaik (R^2 lebih dari 0,93 pada set pelatihan) dalam memprediksi emisi amonia kumulatif selama pengomposan lumpur limbah, sejajar dengan Extra Trees, ketika hyperparameter kedua model tersebut dioptimasi menggunakan Bayesian Optimization dengan

skema validasi GroupKFold. Studi tersebut turut mencatat bahwa performa tinggi CatBoost berpotensi disertai risiko overfitting akibat proses tuning yang mendorong pembentukan kedalaman pohon lebih besar, sehingga penerapan CatBoost pada penelitian ini tetap memerlukan strategi mitigasi seperti pembatasan kedalaman pohon dan validasi silang yang ketat.

Perlu ditegaskan bahwa dalam penelitian ini, istilah ‘klasifikasi kematangan kompos’ merujuk pada proses dua langkah:

1. model bekerja sebagai regressor untuk memprediksi nilai GI secara numerik menggunakan metrik R^2, MAE, dan RMSE, kemudian
2. nilai GI hasil prediksi dikonversi menjadi label kategori kematangan (Fitotoksik, Rendah, Sedang, Matang) melalui operasi thresholding berdasarkan ambang batas yang didefinisikan pada Sub-bab 2.1.2.

Pendekatan ini memungkinkan evaluasi model dilakukan baik secara kontinu (metrik regresi) maupun diskrit (akurasi klasifikasi), sehingga memberikan gambaran kinerja yang lebih komprehensif.

2.4. Hyperparameter Optimization (HPO)

Performa algoritma machine learning berbasis pohon sangat dipengaruhi oleh nilai hyperparameter yang digunakan, seperti kedalaman pohon, learning rate, dan jumlah estimator. Oleh karena itu, pencarian kombinasi hyperparameter terbaik menjadi bagian penting dalam pipeline pemodelan. Penelitian ini membandingkan tiga metode *hyperparameter optimization* (HPO), yaitu *Grid Search*, *Random Search*, dan *Bayesian Optimization*.

2.4.1. Bayesian Optimization

Bayesian Optimization merupakan strategi pencarian hyperparameter yang bekerja dengan membangun model surrogate probabilistik dari hasil evaluasi trial sebelumnya, kemudian menggunakan model tersebut untuk mengarahkan pemilihan kombinasi hyperparameter pada trial berikutnya. Pada penelitian ini, pendekatan tersebut diimplementasikan melalui pustaka Optuna dengan algoritma Tree-structured Parzen Estimator (TPE), yang memungkinkan pencarian

hyperparameter berlangsung lebih terarah dibandingkan Grid Search maupun Random Search (Akiba et al., 2019). Berbeda dengan Grid Search yang mengevaluasi seluruh kombinasi pada grid tetap atau Random Search yang mengambil sampel secara acak tanpa memanfaatkan informasi dari evaluasi sebelumnya, TPE memodelkan distribusi probabilitas kombinasi hyperparameter yang menghasilkan performa baik maupun buruk, lalu memfokuskan trial selanjutnya pada area ruang pencarian yang diperkirakan lebih menjanjikan berdasarkan distribusi tersebut. Konsekuensinya, Bayesian Optimization cenderung mencapai konvergensi menuju wilayah optimal dengan jumlah trial yang relatif lebih sedikit, meskipun biaya komputasi per trial dapat meningkat akibat proses pembaruan model surrogate pada setiap iterasi.

Penerapan Bayesian Optimization pada domain pengomposan telah dilaporkan pula oleh Liu et al. (2026), yang menggabungkan pendekatan ini dengan skema validasi GroupKFold cross-validation untuk menuning hyperparameter beberapa model, yaitu SVR, MLP, Extra Trees, CatBoost, dan GPR, pada tugas prediksi emisi amonia kumulatif selama pengomposan lumpur limbah. Penelitian tersebut melaporkan bahwa optimasi Bayesian berkontribusi signifikan terhadap peningkatan performa model berbasis ensemble seperti Extra Trees dan CatBoost, dengan capaian R^2 di atas 0,93 pada set pelatihan. Namun demikian, Liu et al. (2026) juga mengidentifikasi risiko overfitting sebagai konsekuensi dari proses tuning yang intensif, yang cenderung mendorong pembentukan kedalaman pohon (tree depth) yang lebih besar pada model tersebut.

Temuan tersebut memperkuat relevansi penggunaan Bayesian Optimization pada penelitian ini, sekaligus menjadi pertimbangan penting untuk menerapkan strategi mitigasi overfitting secara konsisten selama proses tuning model CatBoost, antara lain melalui regularisasi, pembatasan kedalaman pohon, penerapan validasi silang yang ketat, serta pemantauan kemampuan generalisasi model secara berkelanjutan.

2.4.2. Grid Search

Grid Search menelusuri seluruh kombinasi parameter secara *exhaustive*, yakni mencoba setiap kemungkinan nilai hyperparameter yang telah ditentukan sebelumnya pada suatu rentang tertentu. Pendekatan ini memberikan cakupan penuh terhadap ruang pencarian yang didefinisikan, sehingga kombinasi terbaik di dalam ruang tersebut dapat ditemukan secara pasti. Namun, Grid Search memiliki skalabilitas yang kurang baik ketika jumlah hyperparameter maupun jumlah kandidat nilai pada tiap hyperparameter meningkat, karena total kombinasi yang harus dievaluasi bertambah secara eksponensial. Akibatnya, biaya komputasi menjadi tinggi pada ruang pencarian yang besar.

Sebagai contoh pada domain kompos, Khandakar et al. (2025) menggunakan GridSearchCV untuk melakukan tuning hyperparameter pada enam model machine learning, termasuk XGBoost dan CatBoost, dalam tugas prediksi kematangan kompos berbasis sensor, dan melaporkan bahwa hasil optimasi tersebut berkontribusi pada pencapaian R^2 sebesar 0,9912 untuk kedua model. Selain itu, Zou et al. (2026) membandingkan Grid Search dan Random Search pada tugas klasifikasi dan regresi kematangan kompos berbasis citra hyperspectral, serta menyimpulkan bahwa Grid Search menunjukkan performa yang lebih baik dalam konteks tersebut, sehingga mereka memilih GridSearchCV untuk optimasi hyperparameter pada empat model yang diuji, yaitu ANN, XGBoost, Random Forest, dan SVM.

2.4.3. Random Search

Random Search memilih sejumlah kombinasi hyperparameter secara acak dari ruang pencarian, alih-alih menelusuri seluruh grid secara exhaustive. Bergstra dan Bengio (2012) menunjukkan bahwa *Random Search* secara empiris lebih efisien dibanding Grid Search ketika pengaruh antar-hyperparameter terhadap performa model tidak merata, karena *Grid Search* cenderung menghabiskan banyak evaluasi pada dimensi yang kurang berpengaruh, sedangkan *Random Search* dapat mengeksplorasi lebih banyak nilai pada dimensi yang lebih penting dengan jumlah percobaan yang sama. Dengan anggaran komputasi yang setara, Random

Search sering kali mampu menemukan kombinasi hyperparameter yang setidaknya sebanding, bahkan lebih baik, daripada *Grid Search* dalam banyak kasus praktis.

Namun, keunggulan ini tidak selalu konsisten pada domain kompos. Zou et al. (2026) justru melaporkan hasil sebaliknya, yaitu *Grid Search* yang lebih fleksibel dibanding *Random Search* pada tugas prediksi kematangan kompos berbasis data hyperspectral. Perbedaan hasil tersebut kemungkinan dipengaruhi oleh ukuran ruang pencarian hyperparameter serta karakteristik dataset yang digunakan. Oleh karena itu, penelitian ini tetap membandingkan ketiga metode HPO, yaitu *Grid Search*, *Random Search*, dan *Bayesian Optimization*, secara empiris, tanpa mengasumsikan bahwa salah satunya unggul secara umum berdasarkan literatur.

2.5. Konsep Soft-Sensor

Konsep soft sensor mengacu pada model matematis atau berbasis data yang digunakan untuk mengestimasi nilai suatu variabel proses yang sulit maupun mahal untuk diukur secara langsung, dengan memanfaatkan variabel lain yang lebih mudah diukur sebagai input (Pural, 2023). Pendekatan ini telah banyak diadopsi pada industri pemrosesan mineral dan sektor proses lainnya, khususnya untuk mengestimasi parameter kualitas penting tanpa harus bergantung pada pengukuran laboratorium secara langsung di lapangan.

Pural (2023) mengembangkan soft sensor untuk memprediksi kadar pengotor silikat pada konsentrat flotasi bijih besi dengan membandingkan tiga algoritma, yaitu Ridge Regression, Multi-Layer Perceptron (MLP), dan Random Forest. Hasil pengujian menunjukkan bahwa *Random Forest* memberikan performa terbaik (R^2 = 0,965; MAE = 0,089), jauh mengungguli MLP (R^2 = 0,886; MAE = 0,243) maupun Ridge Regression (R^2 = 0,153; MAE = 1,213) dalam mengestimasi variabel proses dari data sensor tidak langsung. Temuan tersebut memberikan indikasi empiris bahwa model berbasis pohon lebih unggul dibanding jaringan saraf sederhana untuk tugas soft-sensing pada data tabular berukuran kecil hingga menengah.

Berdasarkan pertimbangan tersebut, penelitian ini tidak menetapkan MLP secara tunggal sebagai model Stage 1 pada Arsitektur B (Two-Stage Soft-Sensor).

Sebagai gantinya, model Stage 1 dipilih secara empiris dari kandidat algoritma berbasis pohon yang sama dengan Stage 2, yaitu CatBoost, Random Forest, dan XGBoost.

2.6. Feature Engineering pada Data Sensor Tabular

Rekayasa fitur (*feature engineering*) merupakan proses pembentukan variabel baru yang lebih informatif dari fitur mentah yang tersedia, dengan tujuan meningkatkan kemampuan model machine learning dalam menangkap pola yang tidak secara langsung tampak dari nilai sensor mentah. Pada data sensor deret waktu multi-batch seperti proses pengomposan, langkah ini menjadi krusial mengingat hubungan antara parameter proses (suhu, kelembapan, pH, dan amonia) dengan tingkat kematangan kompos umumnya bersifat non-linier serta melibatkan interaksi antar-parameter, bukan sekadar fungsi linier dari nilai sensor pada satu titik waktu tertentu.

2.6.1. Fitur Non-linear dan Interaksi Antar-Sensor

Transformasi non-linear seperti kuadrat dan logaritma umum diaplikasikan dalam rekayasa fitur ketika hubungan antara fitur sensor dan target prediksi diketahui tidak bersifat proporsional secara linier, mengingat model yang hanya menerima fitur mentah berpotensi kehilangan sebagian informasi asosiasi tersebut, bahkan pada algoritma berbasis pohon keputusan. Transformasi logaritma khususnya relevan untuk variabel konsentrasi kimia yang bersifat toksik pada level tinggi namun tidak proporsional toksik pada level rendah.

Dalam mengembangkan model EcoCompost berbasis jaringan saraf tiruan multilayer untuk optimasi multi-objektif pengomposan kotoran ternak, Shi et al. (2026) menegaskan bahwa analisis korelasi Pearson secara inheren hanya mampu menangkap dependensi statistik yang bersifat linier, sehingga hanya memberikan wawasan awal terhadap interaksi kompleks antar-parameter proses; model machine learning yang lebih komprehensif diperlukan untuk mengurai secara akurat pengaruh spesifik tiap variabel terhadap kehilangan nutrien dan konversi

zat berbahaya. Melalui analisis feature importance, penelitian tersebut mendapati bahwa rasio C/N merupakan prediktor paling dominan terhadap kehilangan karbon (importance 12,60%) maupun nitrogen (importance 12,12%). Namun, hubungan antara rasio C/N dan kehilangan nitrogen bersifat non-monoton — rasio C/N yang terlalu rendah (10 – 20:1) menyebabkan kehilangan nitrogen tertinggi akibat porositas timbunan yang berkurang dan volatilisasi amonia yang meningkat, sementara rasio C/N yang terlalu tinggi (35 – 50:1) menciptakan lingkungan defisit nitrogen yang memperlambat biodegradasi dan memperpanjang durasi proses, dengan retensi nitrogen optimal justru tercapai pada rasio seimbang di kisaran 25 – 30:1. Pola ini membentuk kurva menyerupai huruf U yang tidak dapat direpresentasikan secara memadai oleh satu fitur linier saja.

Temuan tersebut memperkuat justifikasi penggunaan transformasi non-linear (seperti Temp_sq dan Day_sq) maupun fitur ambang eksplisit (seperti Acidity_Deficit) pada rekayasa fitur penelitian ini, alih-alih hanya mengandalkan nilai sensor mentah yang berasumsi berhubungan linier terhadap target.

Selain transformasi tunggal, fitur interaksi (hasil kali dua fitur sensor) juga lazim digunakan untuk menangkap efek gabungan yang tidak dapat direpresentasikan hanya dengan menjumlahkan kontribusi masing-masing fitur secara terpisah, sebagaimana umum diterapkan pada rekayasa fitur berbasis pengetahuan domain (domain-driven feature engineering) di berbagai bidang pemrosesan data sensor.

2.6.2. Seleksi dan Konstruksi Fitur pada Soft Sensor

Ji, Ma, Wang, dan Sun (2023) mengembangkan metode seleksi fitur berbasis conditional entropy untuk soft sensor pada proses kimia. Penelitian tersebut menunjukkan bahwa performa model soft sensor cenderung menurun pada industri proses dengan jumlah variabel yang besar, terutama apabila terdapat variabel yang redundan atau tidak relevan terhadap variabel target. Pemilihan dan konstruksi fitur yang tepat dengan demikian menjadi tahap krusial untuk menekan redundansi informasi antar-fitur sekaligus mempertahankan, bahkan meningkatkan, akurasi model. Temuan ini menjadi landasan bagi penelitian ini, mengingat fitur turunan (engineered, delta, dan temporal) yang dikonstruksi dari

empat sensor fisik dasar perlu dirancang secara cermat agar mampu menangkap informasi yang relevan terhadap GI tanpa menimbulkan redundansi berlebih antar-fitur.

Analisis feature importance merupakan komponen yang tidak terpisahkan dari proses rekayasa fitur, karena berfungsi mengevaluasi kontribusi masing-masing fitur turunan terhadap hasil prediksi akhir. Khandakar et al. (2025) menerapkan analisis ini melalui model XGBoost yang dikombinasikan dengan uji korelasi Pearson antar-fitur guna mengidentifikasi serta mengeliminasi fitur dengan kontribusi rendah pada model prediksi kematangan kompos berbasis sensor gas dan parameter fisikokimia. Pendekatan tersebut terbukti mampu meningkatkan efisiensi komputasi model tanpa mengorbankan akurasi prediksi secara signifikan, dengan capaian R^2 sebesar 0,9912 baik pada model XGBoost maupun CatBoost.

Selanjutnya, Shi et al. (2026) memperkenalkan pendekatan evaluasi kontribusi fitur yang lebih komprehensif pada model EcoCompost, yaitu melalui kombinasi SHAP (SHapley Additive exPlanations) dan partial dependence plot. Kombinasi ini memungkinkan pengungkapan tidak hanya besaran (magnitude) kontribusi fitur, tetapi juga bentuk hubungannya (arah dan non-linearitas) terhadap variabel target, sehingga melengkapi keterbatasan metode feature importance berbasis model semata yang umumnya hanya melaporkan besaran kontribusi tanpa menjelaskan pola hubungannya. Pendekatan gabungan tersebut relevan untuk diadopsi pada tahap analisis feature importance dalam penelitian ini (Sub-bab 4.3.3), guna menginterpretasikan kontribusi fitur turunan pada model CatBoost terbaik secara lebih komprehensif dibandingkan penggunaan feature importance native semata.

2.6.3. Feature Scaling sebagai Bagian dari Preprocessing

Tahap penyekalaan (scaling) lazim dilakukan setelah rekayasa fitur pada data sensor, dengan tujuan menyetarakan rentang nilai antar-fitur yang memiliki satuan dan skala magnitudo berbeda (misalnya suhu dalam °C dan konsentrasi amonia dalam mg/kg). Tanpa penyekalaan, fitur dengan rentang nilai yang lebih besar berpotensi mendominasi proses pelatihan model secara tidak proporsional, semata-mata akibat perbedaan skala dan bukan karena tingkat relevansinya

terhadap variabel target. Khandakar et al. (2025) menerapkan standardisasi berbasis rata-rata dan deviasi standar pada data sensor kompos sebelum tahap pelatihan model, serta menegaskan bahwa langkah tersebut diperlukan agar setiap fitur berkontribusi secara proporsional dalam proses pelatihan, mengingat perbedaan rentang nilai antar-atribut sensor dapat menimbulkan bias apabila tidak diseragamkan terlebih dahulu.

Meskipun demikian, penyekalaan berbasis rata-rata dan deviasi standar (Z-score) memiliki kelemahan berupa sensitivitas terhadap nilai ekstrem (outlier). Pada data dengan distribusi tidak simetris atau yang mengandung outlier, pendekatan penyekalaan berbasis median dan interquartile range dinilai lebih tahan (robust) terhadap pengaruh nilai ekstrem dibandingkan Z-score maupun Min-Max Scaling. Atas dasar pertimbangan tersebut, pendekatan ini diadopsi pada tahap preprocessing dalam penelitian ini.

2.7. Augmentasi Data Tabular

Augmentasi data merupakan teknik untuk memperluas dataset pelatihan secara artifisial guna meningkatkan kemampuan generalisasi model, khususnya pada kondisi jumlah data asli yang terbatas atau distribusi kelas yang tidak seimbang. Dalam penelitian ini, ketidakseimbangan distribusi kelas GI merupakan tantangan yang perlu mendapat perhatian, sebagaimana tercermin dari komposisi 452 baris data asli berikut: kelas Fitotoksik (GI ≤ 50%) sebanyak 128 baris (28,3%), kelas Rendah (50% < GI ≤ 80%) sebanyak 136 baris (30,1%), kelas Sedang (80% < GI ≤ 100%) sebanyak 145 baris (32,1%), sementara kelas Matang (GI > 100%) hanya sebanyak 43 baris (9,5%) dari keseluruhan dataset. Guna mengatasi ketidakseimbangan tersebut, lima metode augmentasi data tabular diujicobakan dan dibandingkan secara sistematis, yaitu Category-Mixup (C-Mixup), Gaussian Mixture Model oversampling (GMM), dan CTGAN yang diterapkan per kategori GI. Keempat metode augmentasi tersebut akan diterapkan dan dibandingkan secara terkontrol.

2.7.1. Categorical Mixup (C-Mixup)

C-Mixup merupakan pengembangan dari metode Mixup standar yang melakukan interpolasi linier antara pasangan sampel dalam kelompok kategori label yang sama. Berbeda dengan Mixup konvensional yang memungkinkan interpolasi lintas kelas, C-Mixup membatasi proses interpolasi agar setiap sampel sintetis yang dihasilkan tetap berada dalam manifold distribusi kelas aslinya, sehingga konsistensi fisikokimia data dapat terjaga (Yao et al., 2022). Dengan pendekatan ini, titik data baru yang dihasilkan secara matematis valid karena merupakan kombinasi konveks dari kondisi yang telah teramati sebelumnya, sehingga model yang dilatih berpotensi menghasilkan pemetaan yang lebih smooth antara fitur sensor dan nilai GI. Secara teoretis, karakteristik tersebut mendukung kemampuan generalisasi yang lebih baik, karena model tidak dituntut untuk mempelajari sampel sintetis yang terlalu menyimpang dari distribusi data yang sesungguhnya (Yao et al., 2022).

2.7.2. Gaussian Mixture Model Oversampling (GMM)

GMM Oversampling memanfaatkan model probabilistik campuran Gaussian untuk mengestimasi distribusi data pada setiap kelas, kemudian membangkitkan sampel baru dari distribusi hasil estimasi tersebut. Yang dan Cha (2023), dalam mengembangkan metode Gaussian-based Minority Oversampling Technique (GMOTE), mengemukakan bahwa metode oversampling konvensional seperti SMOTE membentuk instance baru melalui interpolasi linear, sehingga ruang data sintetis yang dihasilkan cenderung berbentuk poligonal dan berisiko memunculkan outlier pada kelas minoritas. Sebagai alternatif, GMOTE mengestimasi distribusi kelas minoritas menggunakan Gaussian Mixture Model yang dilatih melalui algoritma Expectation-Maximization (EM), dengan jumlah komponen campuran ditentukan secara otomatis berdasarkan Bayesian Information Criterion (BIC). Selanjutnya, tail probability tiap instance diadaptasi melalui jarak Mahalanobis terhadap pusat masing-masing komponen GMM untuk mendeteksi dan menangani outlier lokal, sebelum sampel sintetis baru dibangkitkan dari distribusi GMM yang telah dibersihkan dari outlier tersebut.

Metode berbasis GMM secara eksplisit menyasar kelas minoritas dengan menambahkan sampel sintetis berdasarkan karakteristik distribusi kelas yang bersangkutan (Yang dan Cha, 2023). Meskipun demikian, metode ini bersifat parametrik dan mengasumsikan bahwa distribusi tiap kelas dapat direpresentasikan sebagai superposisi komponen Gaussian, sehingga kualitas sampel yang dihasilkan sangat bergantung pada jumlah data asli yang menjadi basis estimasi. Mengingat basis estimasi pada penelitian ini hanya terdiri atas 128 baris untuk kelas Fitotoksik dan 43 baris untuk kelas Matang, estimasi kovariansi berisiko tidak stabil terutama pada kelas dengan jumlah sampel paling sedikit. Farou, Wang, dan Horváth (2024), dalam mengevaluasi berbagai metode oversampling berbasis distribusi terhadap permasalahan small disjunct (subset kelas minoritas yang berukuran kecil dan tersebar), secara eksplisit melaporkan bahwa hasil eksperimen menunjukkan efektivitas GMOTE yang terbatas (limited effectiveness) pada subset kelas minoritas yang kecil dan terpisah, serta berpotensi menimbulkan kesalahan klasifikasi (misclassification) pada instance kelas mayoritas di sekitarnya.

2.7.3. CTGAN (Conditional Tabular GAN)

CTGAN adalah metode generatif berbasis Generative Adversarial Network (GAN) yang dirancang khusus untuk data tabular dengan kondisi label tertentu. Model ini melatih dua jaringan secara adversarial: generator yang berusaha menghasilkan sampel sintetis yang menyerupai data asli, dan diskriminator yang berusaha membedakan antara data asli dan sintetis (Xu et al., 2019). CTGAN mampu menangkap distribusi data yang kompleks dan non-Gaussian, serta dapat dikondisikan pada kategori GI tertentu sehingga secara teori cocok untuk menambah kelas minoritas. Review terbaru tentang synthetic tabular data generation menegaskan bahwa efektivitas CTGAN dan model generatif sejenis sangat bergantung pada ketersediaan data asli yang cukup untuk mempelajari distribusi antar fitur; pada kasus data kecil dan sangat timpang, model generatif berisiko mempelajari representasi yang terlalu sempit dan menghasilkan sampel sintetis yang kurang variatif (Fonseca & Bacao, 2023). Oleh karena itu, pada skenario dengan jumlah sampel minoritas yang sangat terbatas, data sintetis yang

dihasilkan berisiko terlalu homogen dan kurang merepresentasikan variasi nyata di lapangan (Fonseca & Bacao, 2023).

2.7.4. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE merupakan salah satu teknik oversampling paling awal dan paling banyak digunakan untuk menangani ketidakseimbangan kelas. Prinsip kerjanya adalah, untuk setiap instance pada kelas minoritas, dipilih salah satu k tetangga terdekatnya (k-nearest neighbors) dalam ruang fitur yang sama, kemudian dibentuk sampel baru melalui interpolasi linear di antara keduanya (Chawla et al., 2002). Dengan pendekatan ini, SMOTE tidak sekadar menggandakan data seperti oversampling sederhana, melainkan menghasilkan titik data baru yang berada di antara dua sampel asli.

Sebagaimana telah dibahas pada GMOTE (Yang dan Cha, 2023), ketergantungan SMOTE pada interpolasi linear dapat menyebabkan sampel sintetis terbentuk di lokasi yang kurang representatif, seperti di area perbatasan antar kelas atau di sekitar data yang mengandung noise, sehingga berpotensi menimbulkan outlier. Selain itu, SMOTE tidak mempertimbangkan distribusi probabilistik global seperti pada GMM, maupun kemampuan untuk memodelkan distribusi kompleks seperti CTGAN. Akibatnya, kualitas data sintetis yang dihasilkan sangat dipengaruhi oleh kepadatan lokal di sekitar instance yang digunakan dalam proses interpolasi.

Di sisi lain, SMOTE memiliki keunggulan dalam hal kesederhanaan komputasi. Prosesnya hanya melibatkan pencarian tetangga terdekat dan interpolasi linear, tanpa memerlukan pelatihan iteratif seperti pada GMM maupun CTGAN. Oleh karena itu, SMOTE tetap relevan digunakan sebagai baseline untuk mengevaluasi apakah pendekatan yang lebih kompleks benar-benar memberikan peningkatan kinerja yang signifikan dibanding metode yang lebih sederhana ini.

Dalam penelitian terkait, Mullick et al. (2025) juga menerapkan SMOTE pada dataset sensor yang sama, yaitu 452 sampel kompos hasil pemantauan (Khandakar et al., 2025). Namun, mereka mengaplikasikan SMOTE pada seluruh dataset sebelum proses pembagian batch dan pemisahan data latih serta uji, sehingga jumlah sampel meningkat menjadi 1.314. Validasi dilakukan menggunakan 100 sampel asli yang disisihkan, dengan hasil performa yang tinggi dan konsisten ($R^2 > 0,95$) pada berbagai algoritma, seperti Linear Regression, Random Forest, XGBoost, Gradient Boosting, dan KNN.

BAB III

METODE PENELITIAN

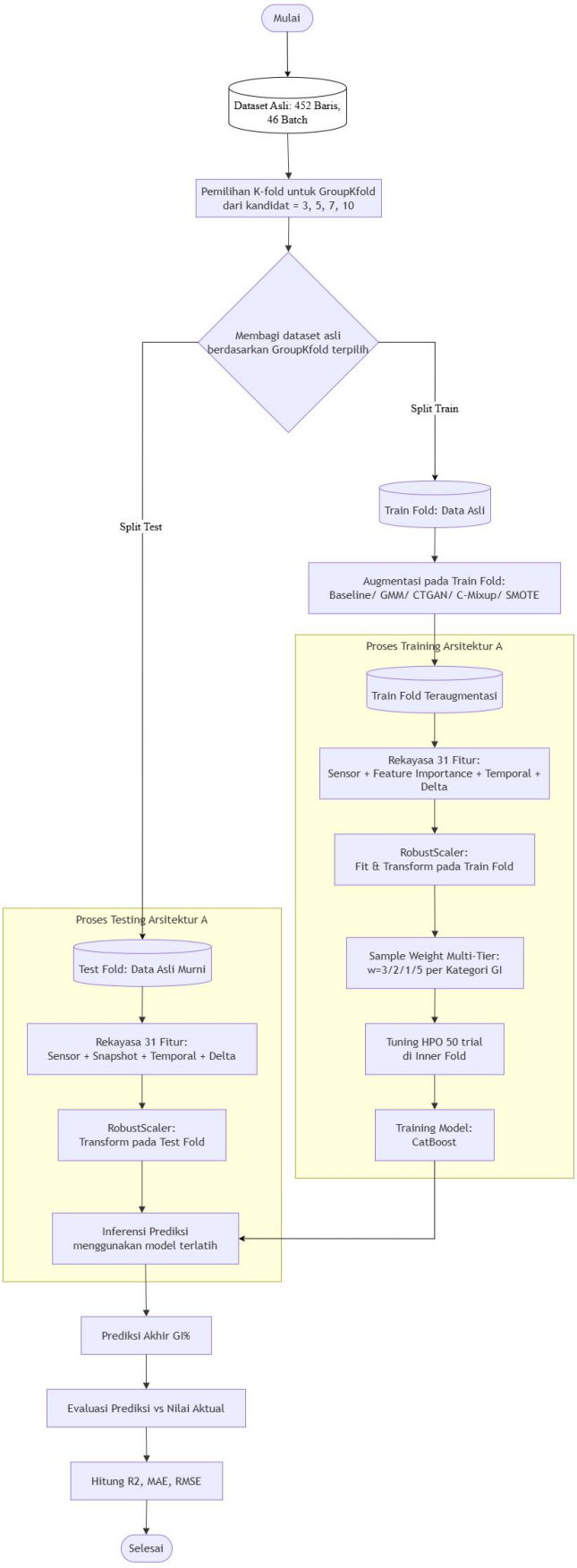
3.1. Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental untuk mengembangkan serta mengevaluasi model machine learning dalam memprediksi tingkat kematangan kompos berdasarkan nilai Germination Index (GI) dari data sensor pengomposan. Pendekatan kuantitatif dipilih karena fokus penelitian terletak pada pengukuran kinerja model secara objektif menggunakan metrik terstandar, seperti RMSE, MAE, dan R^2, serta pada perbandingan terkontrol antara empat metode augmentasi data, dua arsitektur prediksi, tiga algoritma pembelajaran mesin, dan tiga metode optimasi hyperparameter (HPO).

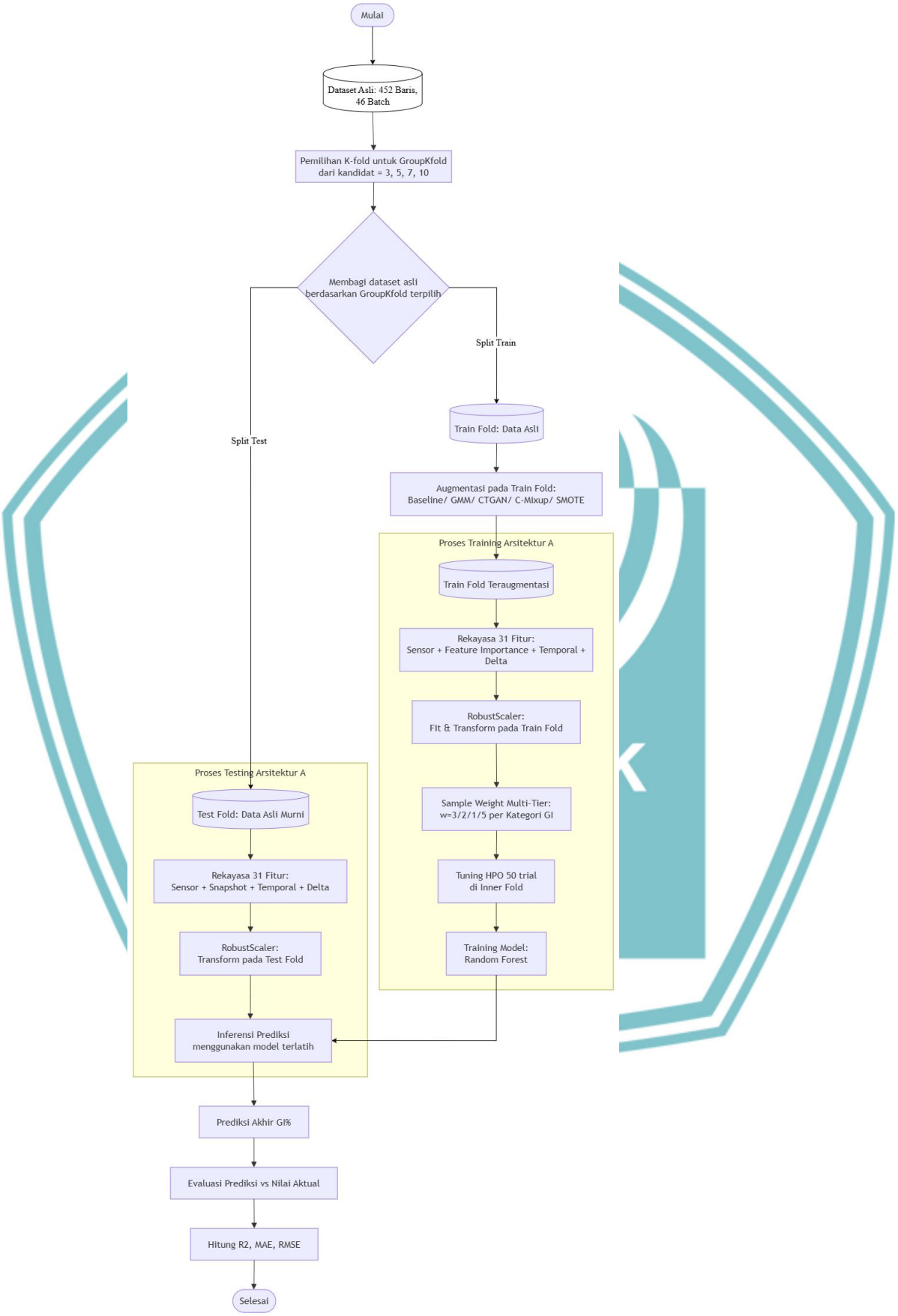
Desain penelitian menggunakan skema nested cross-validation berbasis grup (batch) untuk mencegah terjadinya kebocoran data (data leakage) antar sampel dalam batch yang sama, sekaligus memberikan estimasi performa yang lebih representatif dan tidak bergantung pada satu skenario pembagian data. Secara keseluruhan, penelitian ini dirancang sebagai suatu rekayasa sistem yang mengintegrasikan dua eksperimen utama.

1. melakukan pengembangan dan evaluasi model secara komputasional melalui 270 kombinasi eksperimen (2 arsitektur × 3 model × 4 metode augmentasi ditambah baseline (total jadi 5) × 3 metode HPO × 3 fold) pada dataset publik. Kemudian,
2. melakukan implementasi dan pengujian model terbaik pada perangkat Raspberry Pi 4B sebagai bagian dari sistem smart composter berbasis edge computing.

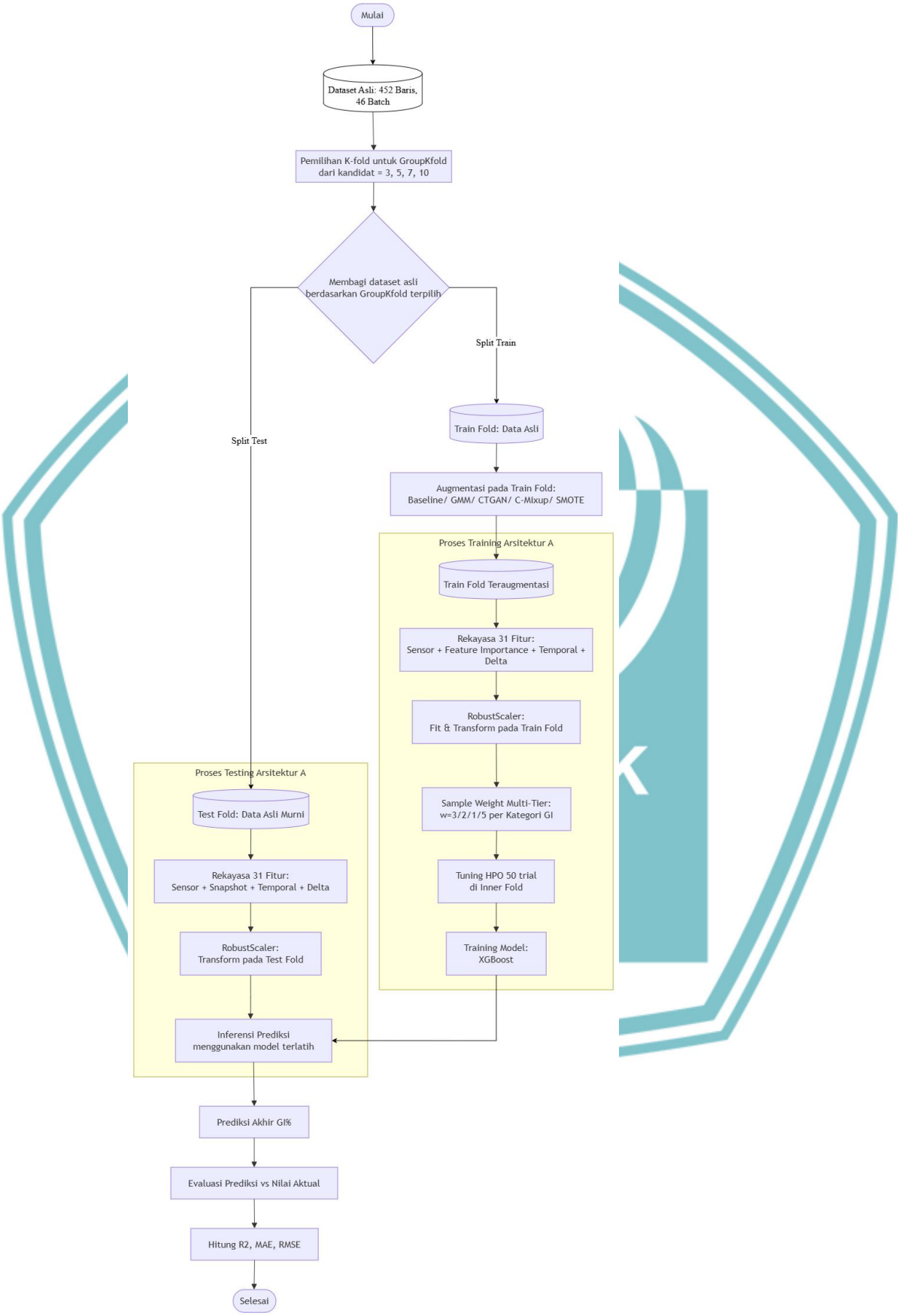
Penelitian ini mengembangkan dua arsitektur prediksi untuk dibandingkan secara langsung dalam kondisi eksperimen yang identik, mencakup pembagian fold, metode augmentasi, pembobotan sampel, serta metrik evaluasi. Arsitektur A (Single-Stage) memetakan fitur sensor secara langsung ke nilai GI (%), sedangkan Arsitektur B (Two-Stage Soft-Sensor) terlebih dahulu mengestimasi parameter laboratorium sebagai representasi antara sebelum menghasilkan prediksi GI. Diagram alur (*Flowchart*) dari masing-masing arsitektur disajikan pada Gambar berikut:



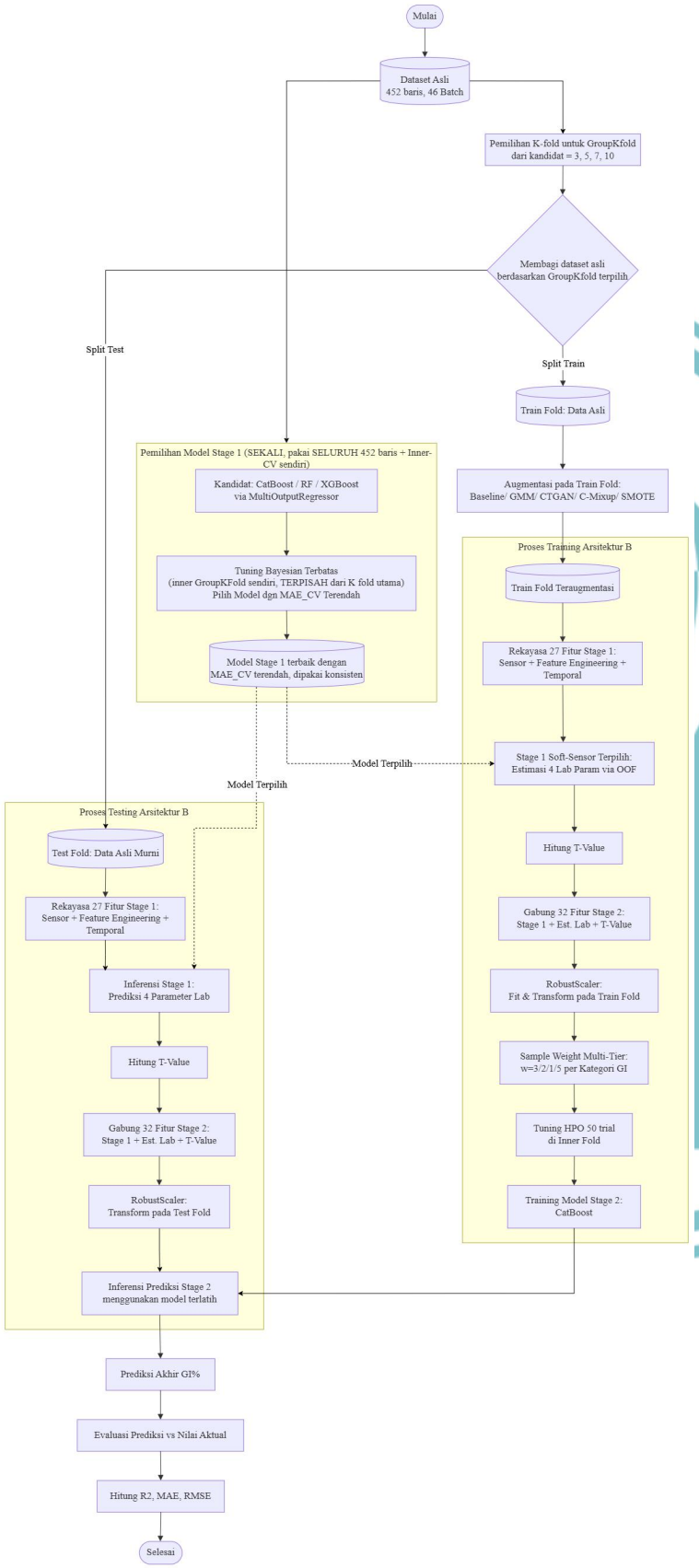
Gambar 3.1 *Diagram Alur (Flowchart) Arsitektur A (Single-Stage) untuk Catboost*



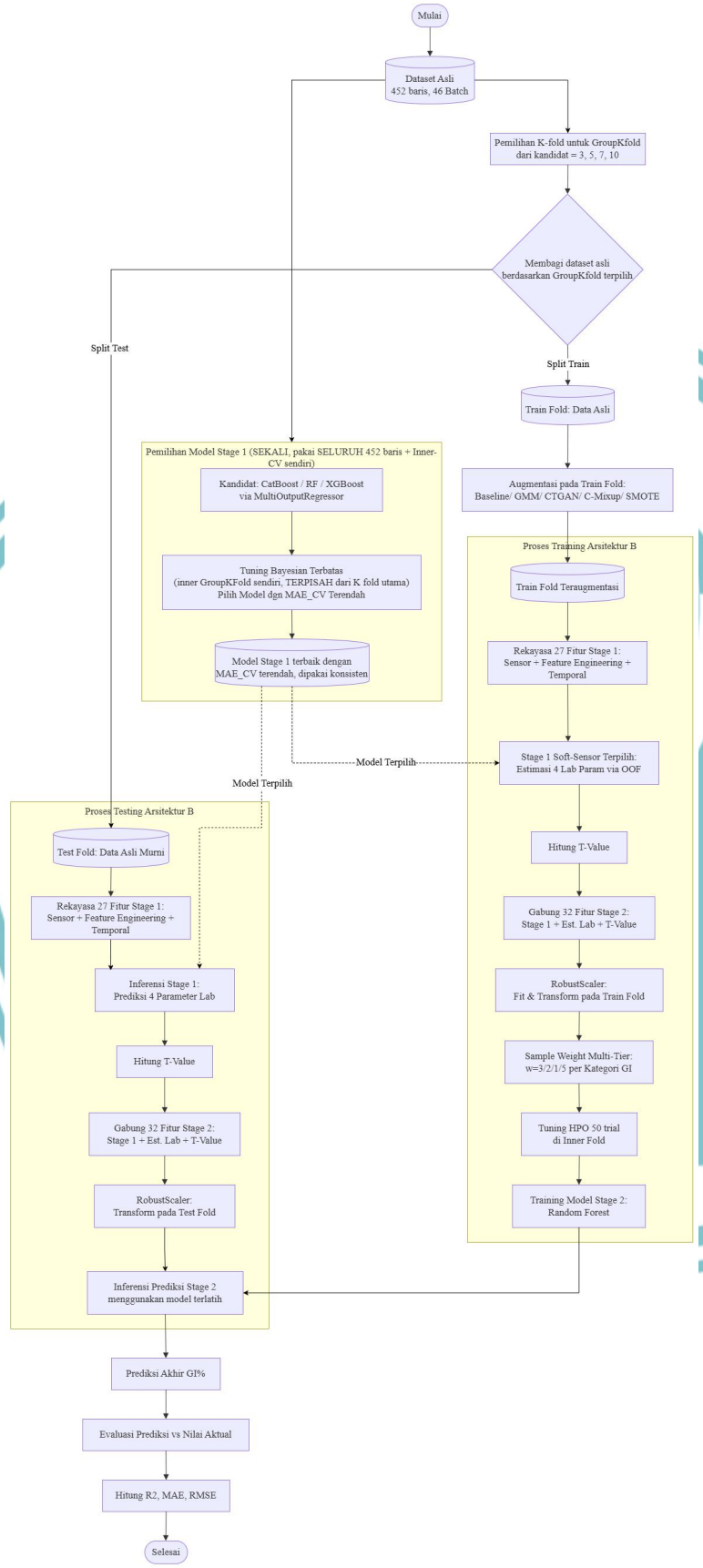
Gambar 3.2 *Diagram Alur (Flowchart) Arsitektur A (Single-Stage) untuk Random Forest*



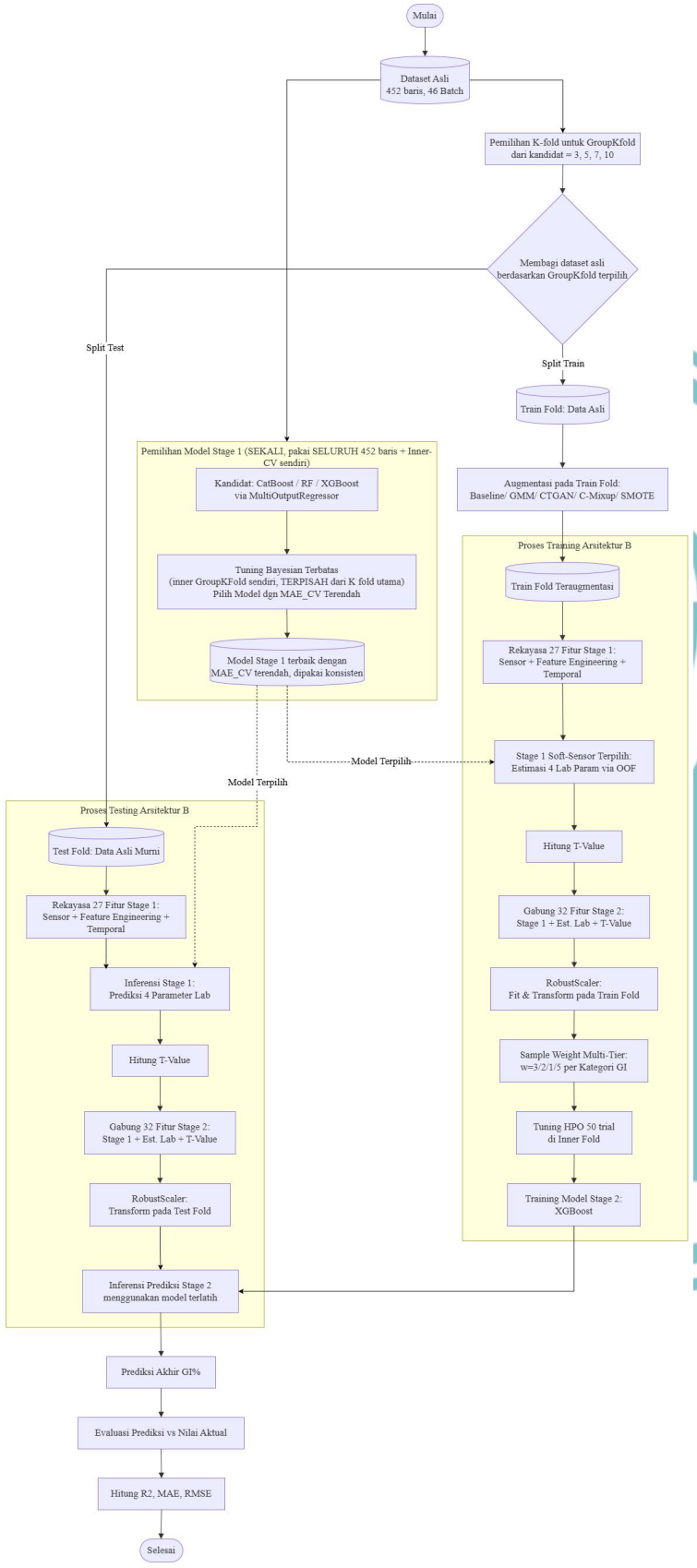
Gambar 3.3 *Diagram Alur (Flowchart) Arsitektur A (Single-Stage) untuk 3 XGBoost*



Gambar 3.4 *Diagram Alur (Flowchart) Arsitektur B (Two-Stage Soft-Sensor) untuk Catboost*



Gambar 3.5 *Diagram Alur (Flowchart) Arsitektur B (Two-Stage Soft-Sensor) untuk Random Forest*



Gambar 3.6 *Diagram Alur (Flowchart) Arsitektur B (Two-Stage Soft-Sensor) untuk XGBoost*

Seluruh eksperimen dilakukan dengan kondisi yang identik: pembagian fold (nested cross-validation berbasis Batch), skema pembobotan sampel, dan metrik evaluasi sama untuk seluruh 270 kombinasi. Dengan demikian, perbandingan kinerja antar skenario augmentasi, arsitektur, model, dan metode HPO dapat dilakukan secara terkontrol dan objektif.

3.2. Tahapan Penelitian

Penelitian dilaksanakan melalui delapan tahapan berurutan sebagai berikut.

3.2.1. Studi Literatur

Studi literatur dilakukan untuk memahami landasan teori pengomposan dan indikator kematangan kompos, algoritma pembelajaran mesin berbasis pohon (CatBoost, Random Forest, XGBoost) untuk regresi tabular, teknik augmentasi data tabular (C-Mixup, GMM, CTGAN, SMOTE), metode hyperparameter optimization (Bayesian Optimization, Random Search, Grid Search), skema validasi nested cross-validation, serta praktik deployment model pada perangkat edge computing berbasis Raspberry Pi.

3.2.2. Akuisisi dan Eksplorasi Dataset

Dataset "Compost Maturity and Emission Monitoring Dataset" diunduh dari Kaggle (Mullick, 2023). Berkas dataset berisi 1.314 baris, terdiri atas 452 baris data asli (hasil pembacaan sensor Arduino/ESP32, ditandai kolom Synthetic = 0) dan 862 baris data sintetis bawaan dari preprocessing SMOGN yang telah dilakukan oleh penyedia dataset di Kaggle. Penelitian ini hanya menggunakan 452 baris data asli (46 batch unik) sebagai data training, dan tidak menggunakan data sintetis SMOGN bawaan tersebut. Augmentasi data dilakukan secara mandiri dan terkontrol melalui empat skema augmentasi yang diterapkan hanya pada data training di dalam skema nested cross-validation, sehingga kontribusi augmentasi terhadap performa dapat diukur dan dibandingkan secara adil antar skema.

Tahap eksplorasi awal dilakukan melalui analisis statistik deskriptif, evaluasi ukuran data pada setiap batch, serta pemeriksaan distribusi nilai GI. Hasil analisis

ukuran batch menunjukkan adanya ketidakseimbangan jumlah data antarbatch. Setiap batch memiliki median sebanyak 7 baris data, dengan jumlah minimum 6 baris, sedangkan Batch 27 teridentifikasi sebagai outlier karena memiliki 72 baris data. Ketimpangan tersebut kemudian menjadi pertimbangan dalam perancangan fungsi *balanced_group_kfold()* untuk proses pembagian data.

Selain itu, analisis distribusi nilai GI menunjukkan bahwa terdapat 43 observasi (9,5% dari total data) dengan nilai GI(%) melebihi 100. Sebagian besar nilai tersebut ditemukan pada sampel kompos dengan usia (Day) yang lebih tinggi. Nilai-nilai tersebut tetap dipertahankan dalam proses pemodelan karena dianggap merepresentasikan fenomena biologis yang valid berupa efek stimulasi pertumbuhan, sehingga tidak dikategorikan sebagai kesalahan pengukuran maupun *noise* pada data.

3.2.3. Preprocessing dan Rekayasa Fitur

Penelitian ini menggunakan *RobustScaler* sebagai metode normalisasi data. Proses *fitting* dilakukan secara eksklusif pada data latih, kemudian parameter yang dihasilkan diterapkan *transform* pada data uji di setiap *fold nested cross-validation* secara independen. Pendekatan ini diterapkan untuk mencegah terjadinya *data leakage*, yaitu masuknya informasi statistik dari data uji, seperti median dan rentang interkuartil (*interquartile range* atau IQR), ke dalam proses pelatihan model. Pemilihan *RobustScaler* didasarkan pada kemampuannya yang lebih baik dalam menangani keberadaan *outlier* dibandingkan metode normalisasi lain, seperti *Min-Max Scaling* maupun standardisasi berbasis *Z-score*. Berbeda dengan kedua metode tersebut yang bergantung pada nilai minimum-maksimum atau rata-rata dan simpangan baku, *RobustScaler* memanfaatkan median dan IQR sehingga lebih tahan terhadap pengaruh data ekstrem. Pemilihan metode ini dinilai sesuai dengan karakteristik dataset yang menunjukkan adanya ketimpangan ukuran batch serta sejumlah nilai GI yang berada di luar rentang umum hasil observasi. Proses rekayasa fitur dirancang secara bertingkat melalui sebuah fungsi tunggal yang diterapkan secara konsisten, baik pada data asli maupun pada setiap sampel hasil augmentasi. Pendekatan ini bertujuan untuk memastikan kesesuaian antara fitur turunan yang dihasilkan dan nilai sensor yang mendasarinya.

Berdasarkan empat variabel sensor mentah, yaitu Temperatur, *Moisture Content* (%), pH, dan Ammonia (mg/kg), serta variabel usia kompos (Day), rekayasa fitur menghasilkan tiga kelompok fitur turunan. Kelompok pertama adalah fitur engineered, yang merepresentasikan kondisi sistem pada suatu titik pengamatan tertentu. Kelompok kedua adalah fitur delta, yang menggambarkan perubahan nilai parameter terhadap kondisi awal pada masing-masing batch, dan kelompok ketiga merupakan fitur temporal yang digunakan untuk menangkap pola perubahan antarobservasi yang berurutan sepanjang proses pengomposan. Definisi konseptual, rumus matematis, serta keterangan notasi untuk setiap fitur dijelaskan pada bagian berikut.

1. Fitur Engineered (9 Fitur)

a) Log_Amm (Logaritma Natural Ammonia)

Fitur yang menangkap transformasi logaritma natural yang diterapkan pada konsentrasi amonia untuk mengurangi pengaruh nilai ekstrem (outlier) serta membantu membentuk hubungan yang lebih linear antara konsentrasi amonia dan nilai target GI. Penerapan transformasi ini didasarkan pada karakteristik biologis amonia yang menunjukkan pola pengaruh nonlinier terhadap proses perkecambahan. Pada konsentrasi tinggi, amonia bersifat toksik sehingga peningkatan efek penghambatan tidak terjadi secara proporsional terhadap kenaikan konsentrasi, melainkan meningkat secara signifikan setelah melewati ambang tertentu. Dirumuskan sebagai:

$$Log\_Amm = ln(1 + Ammonia)$$

Ammonia = konsentrasi amonia terukur sensor MQ-135 (mg/kg)

konstanta 1 ditambahkan sebelum logaritma agar terhindar dari ln(0) yang tidak terdefinisi ketika Ammonia = 0.

b) Day_sq (Kuadrat Usia Kompos)

Fitur yang menjadi representasi nonlinier dari usia kompos yang digunakan untuk memodelkan perubahan laju pematangan selama proses pengomposan. Fitur ini memungkinkan model menangkap dinamika dekomposisi bahan organik yang tidak berlangsung secara konstan, di mana proses penguraian umumnya terjadi lebih cepat pada fase awal akibat tingginya aktivitas mikroorganisme dan

ketersediaan substrat yang mudah terdegradasi, kemudian melambat secara bertahap seiring mendekatinya kondisi kematangan kompos. Dirumuskan sebagai:

$$Day\_sq = Day^2$$

Day = usia kompos dalam satuan hari yangdihitung sejak awal proses (Day = 0 pada hari pertama batch dimulai).

c) Temp_sq (Kuadrat Suhu)

fitur yang digunakan untuk merepresentasikan hubungan nonlinier antara suhu dan aktivitas mikroorganisme pengurai selama proses pengomposan. Transformasi ini memungkinkan model menangkap karakteristik biologis mikroorganisme yang memiliki rentang suhu optimum pada fase termofilik, umumnya sekitar 45–65°C. Aktivitas mikroba cenderung meningkat seiring kenaikan suhu hingga mencapai rentang optimum tersebut, kemudian menurun ketika suhu berada di bawah maupun di atas batas tersebut. Oleh karena itu, hubungan antara suhu dan aktivitas mikroba tidak bersifat linear, melainkan membentuk pola kurvilinear atau parabolik. Dirumuskan sebagai:

$$Temp\_sq = Temperature^2$$

Temperature = suhu tumpukan kompos terukur sensor DS18B20 (°C)

d) Temp_x_Amm (Interaksi Suhu × Amonia)

fitur interaksi yang digunakan untuk merepresentasikan pengaruh gabungan antara suhu dan konsentrasi amonia terhadap proses pematangan kompos. Pembentukan fitur ini didasarkan pada keterkaitan biologis antara kedua variabel, di mana suhu memengaruhi laju dekomposisi senyawa nitrogen organik sekaligus meningkatkan volatilisasi amonia pada kondisi temperatur yang lebih tinggi. Akibatnya, dampak suhu dan amonia terhadap tingkat kematangan kompos tidak bersifat independen dan tidak dapat dijelaskan hanya melalui penjumlahan kontribusi masing-masing variabel secara terpisah. Dengan memasukkan fitur interaksi ini, model diharapkan mampu menangkap hubungan yang lebih kompleks antara kondisi termal dan dinamika senyawa nitrogen selama proses pengomposan. Dirumuskan sebagai:

$$Temp\_x\_Amm = Temperature \times Ammonia$$

e) pH_x_MC (Interaksi pH × Kadar Air)

fitur interaksi yang digunakan untuk merepresentasikan pengaruh gabungan antara tingkat keasaman dan kadar air terhadap aktivitas mikroorganisme selama proses pengomposan. Pembentukan fitur ini didasarkan pada fakta bahwa efektivitas dekomposisi tidak hanya dipengaruhi oleh masing-masing faktor secara terpisah, tetapi juga oleh kombinasi keduanya. Kondisi kompos yang terlalu kering maupun terlalu asam dapat menghambat pertumbuhan dan aktivitas mikroba pengurai. Selain itu, pengaruh kedua faktor tersebut dapat saling memperkuat sehingga menghasilkan efek yang bersifat nonaditif. Dengan memasukkan fitur interaksi ini, model diharapkan mampu menangkap hubungan yang lebih kompleks antara pH, kadar air, dan tingkat kematangan kompos. Dirumuskan sebagai:

$$pH\_x\_MC = pH \times MC(\%)$$

pH = tingkat keasaman kompos terukur sensor pH analog (skala 0–14)
MC(%) = kadar air (moisture content) kompos terukur sensor capacitive soil moisture (%).

f) Amm_per_Day (Amonia per Hari)

Fitur yang merepresentasikan laju akumulasi amonia relatif terhadap usia kompos, sehingga dapat menggambarkan kecepatan proses mineralisasi nitrogen selama pengomposan. Proses mineralisasi tersebut berkaitan erat dengan tingkat kematangan kompos karena mencerminkan perubahan senyawa organik menjadi bentuk yang lebih sederhana dan stabil. Dirumuskan sebagai:

$$Amm\_per\_Day = Ammonia / (Day + 1)$$

konstanta 1 ditambahkan pada penyebut agar terhindar dari pembagian dengan nol pada Day = 0

g) Temp_x_pH (Interaksi Suhu × pH)

fitur interaksi yang digunakan untuk merepresentasikan pengaruh gabungan antara suhu dan tingkat keasaman terhadap aktivitas mikroorganisme selama proses pengomposan. Pembentukan fitur ini didasarkan pada karakteristik mikroba pengurai yang responsnya tidak hanya ditentukan oleh masing-masing faktor secara terpisah, tetapi juga oleh kombinasi keduanya. Meskipun suhu

berada dalam rentang optimum bagi aktivitas mikroba termofilik, kondisi pH yang terlalu rendah dapat menghambat pertumbuhan dan metabolisme beberapa jenis mikroorganisme. Dirumuskan sebagai:

$$Temp\_x\_pH = Temperature \times pH$$

h) pH_x_Day (Interaksi pH × Usia Kompos)

fitur interaksi yang digunakan untuk merepresentasikan perubahan pengaruh pH terhadap tingkat kematangan kompos seiring bertambahnya usia pengomposan. Selama proses dekomposisi, nilai pH umumnya mengalami peningkatan menuju kondisi yang lebih basa akibat pelepasan amonia dan transformasi senyawa nitrogen. Oleh karena itu, interaksi antara pH dan usia kompos tidak hanya menggambarkan nilai pH pada suatu waktu tertentu, tetapi juga memberikan informasi mengenai arah dan laju perubahan kondisi kimia kompos sepanjang proses pematangan.

Penambahan fitur ini didasarkan pada hasil error analysis terhadap Batch 37, yang menunjukkan karakteristik pH awal sangat asam (5,76). Kondisi tersebut diduga menghambat aktivitas mikroorganisme pengurai, meskipun batch tersebut memiliki profil suhu dan usia pengomposan yang serupa dengan batch lain yang mencapai kematangan lebih cepat. Temuan ini mengindikasikan bahwa fitur interaksi linear yang telah digunakan sebelumnya, seperti Temp_x_pH, belum mampu sepenuhnya menangkap keterlambatan proses pematangan yang disebabkan oleh kondisi keasaman awal. Dengan memasukkan interaksi antara pH dan usia kompos, model diharapkan dapat merepresentasikan dinamika pengaruh pH secara lebih komprehensif sepanjang proses pengomposan. Dirumuskan sebagai:

$$pH\_x\_Day = pH \times Day$$

i) Acidity_Deficit (Defisit Keasaman)

fitur ambang (threshold feature) yang digunakan untuk mengukur sejauh mana nilai pH berada di bawah nilai referensi mendekati netral, yaitu 6,5, yang dianggap mendukung aktivitas mikroorganisme pengurai. Fitur ini bernilai nol ketika pH telah mencapai kondisi netral atau cukup basa, sedangkan nilai positif

menunjukkan tingkat keasaman kompos, dengan nilai yang semakin besar merepresentasikan kondisi yang semakin asam.

Penambahan fitur ini bertujuan untuk menangkap pola hubungan nonlinier antara pH dan tingkat kematangan kompos. Hasil error analysis pada beberapa batch dengan pH awal yang sangat rendah menunjukkan bahwa hambatan terhadap proses pematangan tidak meningkat secara proporsional di seluruh rentang pH, melainkan menjadi signifikan ketika nilai pH berada di bawah ambang tertentu. Oleh karena itu, fitur ambang ini dirancang untuk merepresentasikan efek kritis dari kondisi asam yang tidak dapat sepenuhnya dijelaskan oleh fitur pH linear maupun interaksi sederhana. Dirumuskan sebagai:

$$Acidity\_Deficit = max(6{,}5 - pH\ ,\ 0)$$

6,5 = ambang referensi pH mendekati netral yang umum dijadikan acuan kondisi optimal aktivitas mikroba pengurai pada literatur pengomposan

fungsi max(·, 0) memastikan nilai fitur tidak pernah negatif, sehingga fitur ini hanya aktif (bernilai > 0) ketika kompos dalam kondisi asam (pH < 6,5)

2. Fitur Delta (Khusus arsitektur A)

fitur yang digunakan untuk mengukur perubahan nilai sensor pada waktu pengamatan (t) relatif terhadap kondisi awal setiap batch (Day = 0), sehingga tidak merepresentasikan nilai absolut sensor pada suatu waktu tertentu. Melalui pendekatan ini, model memperoleh informasi mengenai sejauh mana proses pengomposan telah berkembang dibandingkan dengan kondisi awal bahan.

Selain menggambarkan dinamika perubahan selama proses dekomposisi, kelompok fitur ini juga berperan sebagai padanan sederhana dari konsep T-Value pada Arsitektur B. Perbedaannya terletak pada metode pembentukannya, di mana fitur delta dihasilkan secara langsung melalui rekayasa fitur terhadap data sensor mentah tanpa melibatkan model estimasi tahap pertama (Stage 1). Dengan demikian, informasi mengenai perkembangan proses pengomposan tetap dapat dimanfaatkan oleh model tanpa menambah kompleksitas arsitektur. Dirumuskan sebagai:

$$Delta\_X(t) = X(t) - X(Day{=}0),\quad X \in \{Temperature,\ MC(\%),\ pH,\ Ammonia\}$$

X(t) = nilai sensor pada observasi hari ke-t

X(Day=0) = nilai sensor pada hari pertama (awal) batch yang sama dengan observasi t.

Notasi ini menghasilkan empat fitur turunan, yaitu Delta_Temperature, Delta_MC, Delta_pH, dan Delta_Ammonia.

3. Fitur Temporal (13 Fitur. Dipakai kedua arsitektur)

Kelompok fitur ini dirancang untuk menangkap pola perubahan jangka pendek antarobservasi yang berurutan dalam setiap batch pengomposan. Fitur temporal tersebut diterapkan pada kedua arsitektur yang diusulkan agar proses perbandingan antara Arsitektur A dan Arsitektur B tetap berlangsung secara adil. Dengan demikian, perbedaan kinerja yang diperoleh diharapkan terutama mencerminkan perbedaan pada konsep dan rancangan utama masing-masing arsitektur, bukan akibat adanya ketidakseimbangan informasi yang diberikan kepada model.

a) **Lag1_X (Nilai Observasi Sebelumnya (4 fitur dari setiap sensor))**

Fitur dari nilai sensor pada observasi sebelumnya dalam batch yang sama, bukan pada hari kalender sebelumnya, mengingat interval pengukuran pada dataset ini tidak selalu seragam. Fitur ini memberikan informasi kontekstual mengenai kondisi sistem tepat sebelum waktu pengamatan saat ini, sehingga model dapat mengenali pola transisi dan mendeteksi perubahan yang terjadi secara mendadak selama proses pengomposan. Dirumuskan sebagai:

$$Lag1\_X(t) = X(t-1)$$

X(t−1) = nilai fitur X pada observasi urutan sebelumnya dalam batch yang sama, setelah data diurutkan berdasarkan Day.

Untuk observasi pertama pada tiap batch (tidak memiliki observasi sebelumnya), nilai Lag1_X diisi sama dengan X(t) itu sendiri (asumsi tidak ada perubahan) untuk menghindari nilai kosong (NaN). Notasi ini menghasilkan empat fitur turunan: Lag1_Temperature, Lag1_MC, Lag1_pH, dan Lag1_Ammonia.

b) **RateOfChange_X (Laju Perubahan per Hari (4 fitur dari setiap sensor))**

fitur yang merepresentasikan laju perubahan nilai sensor per hari, sehingga mampu menggambarkan kecepatan perubahan kondisi selama proses pengomposan, bukan hanya besarnya perubahan yang terjadi. Fitur ini relevan karena perbedaan laju dekomposisi dapat memberikan indikasi yang berbeda

mengenai tahap kematangan kompos, meskipun dua sampel memiliki nilai sensor absolut yang serupa pada waktu pengamatan yang sama. Dengan demikian, informasi mengenai dinamika perubahan yang ditangkap oleh fitur ini dapat membantu model dalam membedakan kondisi kompos yang berada pada fase pematangan yang berbeda. Dirumuskan sebagai:

$$RateOfChange\_X(t) = [X(t) - X(t-1)] / [Day(t) - Day(t-1)]$$

Pada observasi pertama tiap batch, nilai RateOfChange_X diisi nol. Notasi ini menghasilkan empat fitur turunan: RateOfChange_Temperature, RateOfChange_MC, RateOfChange_pH, dan RateOfChange_Ammonia.

c) **Rolling_mean_X (Rata-Rata Bergerak (4 fitur dari setiap sensor))**

rata-rata bergerak (moving average) yang dihitung dari tiga observasi terakhir dalam satu batch pengomposan. Fitur ini digunakan untuk mengurangi pengaruh fluktuasi jangka pendek dan noise pengukuran yang dapat muncul akibat keterbatasan presisi maupun proses kalibrasi sensor. Dengan melakukan perataan terhadap beberapa pengamatan sebelumnya, fitur ini memungkinkan model untuk lebih berfokus pada tren perubahan yang stabil dibandingkan nilai sensor mentah pada satu waktu pengamatan tertentu. Dirumuskan sebagai:

$$Rolling\_mean\_X(t) = (1/3) \times \Sigma \text{ (dari } i = t-2 \text{ sampai } t) \; X(i)$$

X(i) = nilai fitur X pada observasi ke-i

penjumlahan dan pembagian dihitung pada window bergerak sebanyak 3 observasi terakhir, dengan window yang menyesuaikan (lebih pendek) pada observasi awal batch yang belum memiliki 3 riwayat data. Perhitungan dilakukan per batch (tidak melewati batas antar batch yang berbeda). Notasi ini menghasilkan empat fitur turunan: Rolling_mean_Temperature, Rolling_mean_MC, Rolling_mean_pH, dan Rolling_mean_Ammonia.

d) **CumDegreeDay (Akumulasi Derajat-Hari (1 fitur))**

fitur yang merepresentasikan akumulasi paparan panas (thermal exposure) secara kumulatif selama proses pengomposan. Konsep ini diadaptasi dari metode growing degree days yang umum digunakan dalam bidang agronomi untuk menggambarkan akumulasi energi panas sepanjang periode pertumbuhan.

Berbeda dengan fitur Temperature maupun Temp_sq yang hanya mencerminkan kondisi suhu pada suatu waktu pengamatan tertentu, fitur ini menggambarkan total energi panas yang telah terakumulasi selama proses pengomposan. Akumulasi tersebut berkaitan erat dengan total aktivitas mikroorganisme termofilik, sehingga dapat memberikan informasi tambahan mengenai perkembangan dan tingkat kematangan kompos. Dirumuskan sebagai:

$$CumDegreeDay(t) = \ln(1 + \Sigma \text{ (dari } i=0 \text{ sampai } t) [\ Temperature(i) \times \Delta Day(i)\ ])$$

Temperature(i) = suhu pada observasi ke-i; ΔDay(i) = selisih usia kompos (hari) antara observasi ke-i dan observasi sebelumnya (interval waktu antar-pengukuran).

penjumlahan dilakukan kumulatif sejak Day = 0 hingga observasi saat ini (t) dalam batch yang sama. Transformasi logaritmik ln(1+x) diterapkan untuk menekan rentang nilai mentah (dapat mencapai ribuan, berkisar 0–2384 pada dataset ini) menjadi skala yang jauh lebih sepadan dengan fitur lain (~0–7,8), tanpa mengubah urutan monotonisitasnya. Model berbasis decission tree (CatBoost, Random Forest, XGBoost) bersifat invariant terhadap transformasi monoton semacam ini sehingga tidak kehilangan informasi peringkat (ranking) data.

Secara keseluruhan, proses rekayasa fitur menghasilkan 31 fitur masukan pada Arsitektur A (Single-Stage), yang terdiri atas empat fitur sensor mentah, satu fitur usia kompos (Day), sembilan fitur engineered, empat fitur delta, dan tiga belas fitur temporal. Sementara itu, Stage 1 pada Arsitektur B menggunakan total 27 fitur, dengan komposisi yang identik kecuali tanpa empat fitur delta.

Peran fitur delta pada Arsitektur B digantikan oleh keluaran estimasi Stage 1 dan konsep T-Value. Dengan demikian, perbedaan jumlah fitur antara kedua arsitektur tidak berasal dari variasi informasi sensor yang digunakan, melainkan dari perbedaan pendekatan dalam merepresentasikan perkembangan proses pengomposan.

3.2.4. Rancangan Dua Arsitektur Prediksi

Penelitian ini merancang dan membandingkan dua arsitektur prediksi GI secara head-to-head, keduanya dapat diisi oleh algoritma pembelajaran mesin yang sama untuk perbandingan yang adil:

1. Arsitektur A (Single-Stage): memetakan 31 fitur sensor (Sub-bab 3.2.3) langsung ke nilai GI(%), tanpa representasi antara. Direpresentasikan pada Gambar 3.1.
2. Arsitektur B (Two-Stage Soft-Sensor): meniru cara laboran menilai kematangan kompos secara bertahap dengan mengestimasi empat parameter laboratorium (C/N Ratio, EC, TN, Nitrate) terlebih dahulu dari 27 fitur Stage 1, kemudian menghitung T-Value dari estimasi C/N Ratio:

$$T\text{-}Value(t) = \hat{C}/N(t) \,/\, \hat{C}/N(Day=0)$$

di mana $\hat{C}/N(t)$ adalah estimasi C/N Ratio hasil Stage 1 pada hari ke-t, dan $\hat{C}/N(Day=0)$ adalah estimasi pada hari pertama batch yang sama. Estimasi empat parameter laboratorium beserta T-Value kemudian digabungkan dengan 27 fitur Stage 1 untuk membentuk 32 fitur Stage 2, yang selanjutnya digunakan model akhir untuk memprediksi GI(%). Diagram alur lengkap disajikan pada Gambar 3.2.

Model yang digunakan pada Stage 1 tidak ditentukan secara apriori, melainkan dipilih berdasarkan hasil evaluasi empiris terhadap tiga algoritma kandidat yang juga digunakan pada Stage 2, yaitu CatBoost, Random Forest, dan XGBoost. Ketiga algoritma tersebut dibungkus menggunakan MultiOutputRegressor untuk memungkinkan estimasi empat parameter laboratorium secara simultan. Setiap kandidat menjalani proses optimasi hiperparameter secara terbatas menggunakan Bayesian Optimization dan dievaluasi berdasarkan nilai mean absolute error (MAE) rata-rata pada seluruh cross-validation fold. Model dengan nilai MAE_mean_CV terendah selanjutnya ditetapkan sebagai model terbaik untuk Stage 1 pada arsitektur B dan digunakan secara konsisten pada seluruh proses out-of-fold (OOF) Stage 1.

Strategi pemilihan model ini diterapkan untuk menghindari asumsi bahwa satu jenis arsitektur tertentu, seperti Multi-Layer Perceptron (MLP), secara otomatis menjadi pilihan terbaik untuk tugas soft-sensor pada dataset yang digunakan. Selain itu, pendekatan ini juga menjaga konsistensi keluarga algoritma berbasis pohon (tree-based) pada kedua tahap pemodelan.

Keputusan untuk tidak memasukkan MLP sebagai kandidat Stage 1 turut didukung oleh temuan empiris pada penelitian sebelumnya. Pural (2023), dalam pengembangan soft sensor untuk memprediksi kadar silikat pada konsentrat flotasi bijih besi, melaporkan bahwa Random Forest secara konsisten memberikan kinerja yang lebih baik dibandingkan MLP, dengan koefisien determinasi (R^2) mencapai 96,5%. Hasil tersebut menunjukkan bahwa model berbasis pohon memiliki kemampuan yang lebih baik dalam mempelajari hubungan antara variabel proses dan data sensor tidak langsung dibandingkan jaringan saraf sederhana, khususnya pada data tabular berukuran kecil hingga menengah. Temuan tersebut sejalan dengan karakteristik dataset yang digunakan dalam penelitian ini dan mendukung pemilihan pendekatan tree-based untuk tugas soft-sensor.

3.2.5. Eksperimen Lima Metode Augmentasi

Empat metode augmentasi dieksperimenkan secara terpisah dengan kondisi identik, diterapkan hanya pada data fold latih untuk mencegah kebocoran data:

1. Baseline: tanpa augmentasi tambahan. data fold latih digunakan apa adanya.
2. C-Mixup: interpolasi linear antar baris dengan nilai GI berdekatan, dalam kategori kematangan yang sama (metode Zucconi: Fitotoksik<50, Rendah 50–80, Sedang 80–100, Matang>100), dengan bobot pemilihan pasangan berbanding terbalik terhadap selisih GI:

$$X' = \lambda X_i + (1-\lambda)X_j,\ \lambda \sim \text{Beta}(0{,}4\ ;\ 0{,}4),\ w_{ij} \propto 1 / (|GI_i - GI_j| + \varepsilon)$$

Setiap baris basis (i) dipasangkan dengan maksimal 2 baris tetangga (n_per=2) yang dipilih berdasarkan bobot invers-jarak GI di atas. Baris hasil mixup mewarisi label Batch dari baris basis (i) agar tetap valid digunakan pada skema GroupKFold.

3. GMM (Gaussian Mixture Model): oversampling generatif per kategori GI menggunakan model Gaussian Mixture (1–3 komponen, adaptif menurut ukuran kategori:

$$n\_komponen = \max(1, \min(3, n\_tersedia \,//\, 3))$$

di mana n_tersedia adalah jumlah baris data asli pada kategori GI tersebut (dalam fold latih), dan // menyatakan pembagian bulat (hasil dibulatkan ke bawah) yang di-fit secara eksklusif pada data fold latih, kemudian di-sampling hingga mencapai target populasi 60 sampel per kategori.

Kategori dengan jumlah anggota kurang dari 3 baris dilewati karena data tidak mencukupi untuk membentuk model campuran yang stabil. Sampel hasil sampling yang jatuh di luar rentang kategori GI aslinya (akibat variasi generatif) dibuang, sehingga hanya sampel yang konsisten dengan kategori target yang dipertahankan. Baris hasil sampling diberi label Batch sintetis unik (id negatif) karena tidak berasal dari satu batch tertentu.

4. CTGAN (Conditional Tabular GAN): pembangkitan data sintetis kondisional per kategori GI menggunakan Conditional Generative Adversarial Network (epoch=100), diterapkan hanya pada kategori Zucconi yang memiliki jumlah data mencukupi (minimal 10 baris); kategori dengan data kurang dari itu dilewati. Ukuran batch pelatihan disesuaikan secara adaptif terhadap jumlah data per kategori (kelipatan 10, antara 10–50) agar proses pelatihan tetap stabil pada kategori dengan data terbatas. Untuk mencegah overfitting terhadap data yang sedikit, dimensi generator dan diskriminator diperkecil dari nilai default (256,256) menjadi (64,64). Nilai hasil generasi kemudian dibatasi (clipping) sesuai rentang fisik yang masuk akal untuk tiap sensor (GI% ≤ nilai maksimum data asli; pH: 3,5–9,5; Suhu: 15–75°C; Kadar Air: 10–85%; Amonia: 0–15.000 mg/kg; Hari: 0–150), guna mencegah nilai sintetis yang secara fisik tidak realistis. Kualitas data sintetis divalidasi dengan membandingkan rata-rata, standar deviasi, dan kemiripan struktur korelasi antar fitur terhadap data asli per kategori, yang hasilnya dicatat sebagai laporan validasi. Sama seperti GMM, baris hasil CTGAN diberi label Batch sintetis unik.

5. SMOTE (Synthetic Minority Oversampling Technique): oversampling sintetis intra-kategori GI menggunakan interpolasi k-nearest neighbors (k-NN). Untuk setiap kategori kematangan Zucconi yang populasinya di bawah target (dan memiliki minimal 3 anggota), jarak k-NN dihitung antar baris dalam kategori yang sama menggunakan seluruh fitur numerik dalam skala mentah (belum di-scaling), agar sampel sintetis tetap merepresentasikan struktur multivariat data asli. Sampel baru dibangkitkan dengan interpolasi antara satu baris data asli dan salah satu dari k tetangga terdekatnya, dengan k = min(5, n−1). k diset 5 kecuali jumlah anggota kategori (n) kurang dari 6, sehingga k disesuaikan menjadi n−1 untuk menghindari galat. Proses diulang hingga populasi tiap kategori mencapai target 60 sampel. Sampel hasil interpolasi yang GI-nya bergeser keluar dari kategori aslinya turut dibuang. Berbeda dengan C-Mixup, baris hasil SMOTE diberi label Batch sintetis unik sama seperti skema pada GMM dan CTGAN karena baris sintetis merupakan hasil interpolasi generik dalam kategori, bukan pewarisan langsung dari satu baris induk tertentu.

Untuk seluruh metode generatif (GMM, CTGAN, SMOTE), proses fitting dan pembangkitan sampel dilakukan eksklusif pada data fold latih di setiap fold nested cross-validation, konsisten dengan prinsip pencegahan data leakage yang diterapkan di seluruh skenario augmentasi. Tabel 3.1 merangkum keempat metode augmentasi (selain baseline) beserta parameter utamanya.

Tabel 3.1 *Ringkasan Parameter Metode Augmentasi Data*

Metode	Jenis	Parameter Utama
C-Mixup	Interpolasi intra-kategori	λ~Beta(0,4;0,4), bobot $\propto$ 1/selisih GI
GMM	Generatif (model campuran)	1–3 komponen, target 60 sampel/kategori
CTGAN	Generatif (GAN kondisional)	100 epoch, ≥10 baris/kategori
SMOTE	Interpolasi k-NN (minoritas)	k=min(5,n-1), target 60 sampel/kategori

3.2.6. Nested Cross-Validation dan Hyperparameter Optimization

penelitian ini menerapkan skema nested cross-validation penuh. Jumlah fold terluar (K) ditentukan melalui analisis stabilitas empiris terhadap kandidat K = {3, 5, 7, 10} menggunakan model proxy Random Forest, dipilih berdasarkan standar deviasi MAE antar-fold terkecil (bukan sekadar rata-rata terendah). Lalu tiga algoritma pembelajaran mesin dibandingkan pada tiap arsitektur: CatBoost,

Random Forest, dan XGBoost. Seluruhnya model berbasis pohon keputusan (tree-based) yang secara empiris lebih sesuai untuk data tabular berukuran menengah (452 baris asli). Ketiga model dituning menggunakan tiga metode hyperparameter optimization (HPO) dengan anggaran pencarian yang dibuat sepadan (n_trials = 50 untuk Bayesian dan Random Search):

1. Bayesian Optimization (Optuna, algoritma TPE): mengarahkan trial berikutnya berdasarkan hasil trial sebelumnya melalui model surrogate probabilistik.
2. Random Search: mengambil n_iter = 50 sampel acak dari ruang pencarian.
3. Grid Search: menjelajahi seluruh kombinasi grid parameter secara exhaustive.

Tabel 3.2 merangkum ruang pencarian hyperparameter untuk masing-masing model.

Tabel 3.2 *Ruang Pencarian Hyperparameter per Model*

Model	Hyperparameter	Rentang Pencarian
CatBoost	iterations, depth, learning_rate, l2_leaf_reg	50–200 (step 25); 4–10; 0,01–0,3 (log-scale); 1,0–10,0
Random Forest	n_estimators, max_depth, min_samples_leaf, max_features	50–200 (step 25); 3–20; 1–8; {sqrt, log2, None}
XGBoost	n_estimators, max_depth, learning_rate, subsample, colsample_bytree	50–200 (step 25); 3–10; 0,01–0,3 (log-scale); 0,6–1,0; 0,6–1,0

Pembobotan sampel (sample weight) multi-tier diterapkan pada seluruh proses pelatihan untuk menangani ketidakseimbangan distribusi kategori GI, mengikuti metode kategorisasi Zucconi:

$$w(GI) = 3{,}0 \text{ jika } GI \leq 50 \text{ (Fitotoksik)};\ 2{,}0 \text{ jika } 50 < GI \leq 80 \text{ (Rendah)};\ 1{,}0 \text{ jika } 80 < GI \leq 100 \text{ (Sedang)};\ 5{,}0 \text{ jika } GI > 100 \text{ (Matang)}$$

Bobot ekstra diberikan pada kategori Fitotoksik (bukan hanya kategori Matang) karena kesalahan klasifikasi kompos fitotoksik sebagai "cukup matang" berisiko merusak tanaman apabila kompos benar-benar diaplikasikan, sama pentingnya dengan mendeteksi kompos yang benar-benar matang secara akurat.

Evaluasi model pada setiap kombinasi (Arsitektur × Model × Augmentasi × Metode HPO × Fold) menggunakan tiga metrik yang dihitung identik di seluruh kombinasi:

$$RMSE = \sqrt{[\ (1/n)\ \Sigma_i\ (y_i - \hat{y}_i)^2\ ]}$$

$$MAE = (1/n)\ \Sigma_i\ |y_i - \hat{y}_i|$$

$$R^2 = 1 - [\ \Sigma_i (y_i - \hat{y}_i)^2 / \Sigma_i (y_i - \bar{y})^2\ ]$$

Rancangan pengujian penuh menghasilkan 270 kombinasi eksperimen (2 arsitektur × 3 model × 5 augmentasi + baseline × 3 metode HPO × 3 fold), dengan mekanisme checkpoint-based resume yang memungkinkan proses dilanjutkan tanpa mengulang kombinasi yang sudah selesai apabila proses komputasi terhenti.

3.2.7. Perbandingan Statistik, Ablation Study, dan Ensemble

Untuk menjawab rumusan masalah mengenai pengaruh arsitektur dan algoritma secara rigor, dilakukan tiga analisis tambahan pada hasil nested cross-validation:

1. Uji statistik berpasangan (paired) antara RMSE Arsitektur A dan B pada seluruh kombinasi Model × Augmentasi × HPO_Method × Fold yang sepadan, menggunakan paired t-test:

   $$t = \bar{d} / (s\_d / \sqrt{n})$$

   $\bar{d}$ = rata-rata selisih berpasangan

   s_d = standar deviasi selisih

   dan uji non-parametrik Wilcoxon signed-rank sebagai pelengkap yang tidak mengasumsikan distribusi normal pada selisih data.

2. Ablation study yang mengisolasi kontribusi augmentasi data dan arsitektur secara terpisah, dengan merata-ratakan RMSE lintas dimensi lain yang tidak diuji.
3. Pendekatan ensemble yang menggabungkan prediksi tiga model (CatBoost, Random Forest, XGBoost) dengan bobot non-negatif berjumlah satu, dioptimasi dari prediksi Out-of-Fold menggunakan metode SLSQP (Sequential Least Squares Programming) untuk meminimalkan MAE tertimbang:

   $$\text{minimalkan } MAE(\Sigma_m\ w_m\ \hat{y}_m) \ \text{ terhadap } \ \Sigma_m\ w_m = 1,\ \ w_m \geq 0$$

Interval kepercayaan 95% (CI95%) untuk RMSE dan R^2 dihitung pada setiap kombinasi menggunakan aproksimasi distribusi normal (z = 1,96) dari 3 nilai fold, alih-alih distribusi-t yang idealnya lebih tepat untuk ukuran sampel sekecil ini (n=3 fold); pendekatan z ini dipilih demi kesederhanaan komputasi dan konsistensi dengan literatur nested cross-validation sejenis, dengan konsekuensi interval yang dihasilkan sedikit lebih sempit (kurang konservatif) dibandingkan bila memakai t(0,975 ; df=2) = 4,303:

$$CI95\% = \bar{x} \pm z \times (s / \sqrt{n}),\ dengan\ z = 1,96$$

Sebagai pengujian tambahan di luar skema nested cross-validation, dilakukan pula validasi hold-out terhadap 15% batch (dipilih acak dengan RNG seed tetap, sebelum augmentasi dan pelatihan) yang genuinely tidak pernah dilihat model manapun selama pelatihan maupun tuning hyperparameter, untuk memverifikasi estimasi performa nested-CV terhadap data yang benar-benar independen.

3.2.8. Konversi dan Deployment Model

Pipeline final (RobustScaler + model teroptimasi, direfit pada seluruh 452 baris data asli plus augmentasi terbaik) disimpan sebagai artefak siap-deploy menggunakan pustaka joblib. Berbeda dari rancangan sebelumnya yang memerlukan konversi model CNN ke format TensorFlow Lite (TFLite), penelitian ini menggunakan model berbasis pohon (CatBoost/Random Forest/XGBoost) yang tidak memerlukan konversi format khusus. Model dapat langsung dimuat pada perangkat edge computing (Raspberry Pi 4B) menggunakan pustaka Python standar (joblib, scikit-learn, catboost/xgboost).

Hasil prediksi tingkat kematangan kompos ditampilkan kepada pengguna melalui dua kanal output: layar LCD yang terpasang pada perangkat smart composter, dan aplikasi smartphone yang menerima data melalui koneksi internet. Seluruh komputasi inferensi model, mulai dari preprocessing sensor hingga prediksi GI, dilakukan secara lokal di Raspberry Pi 4B tanpa ketergantungan terhadap koneksi internet. Koneksi internet hanya digunakan untuk pengiriman hasil prediksi ke aplikasi smartphone, sehingga sistem tetap dapat memberikan

hasil prediksi kematangan kompos meskipun koneksi internet tidak tersedia. Dalam kondisi deployment nyata, fitur Day pada pipeline inferensi diperoleh dari pencacah hari otomatis yang dikelola sistem operasi Raspberry Pi 4B menggunakan real-time clock (RTC) terintegrasi, dihitung sebagai selisih antara tanggal pengukuran saat ini dan tanggal mulai batch pengomposan yang sedang berjalan. Pada Arsitektur B, nilai referensi $\hat{C}/N(Day=0)$ yang dibutuhkan untuk menghitung T-Value disimpan secara persisten dalam berkas konfigurasi lokal di Raspberry Pi 4B dan diperbarui otomatis setiap kali batch pengomposan baru dimulai. Pendekatan ini memungkinkan seluruh pipeline inferensi berjalan mandiri hanya dengan empat masukan dari sensor fisik (suhu, kelembapan, pH, amonia), tanpa memerlukan pengukuran laboratorium tambahan di lapangan.

3.2.9. Pengujian Latensi Inferensi

Pengujian latensi dilakukan di Raspberry Pi 4B menggunakan script Python yang menjalankan pipeline inferensi penuh sebanyak 200 kali setelah warm-up 10 kali. Pengukuran waktu menggunakan fungsi time.perf_counter() dengan resolusi sub-milidetik. Kriteria keberhasilan ditetapkan rata-rata latensi kurang dari 50 milidetik per inferensi agar sistem dapat memberikan prediksi secara mendekati real-time. Mengingat model final berbasis pohon keputusan. Namun tetap perlu diverifikasi secara empiris pada Bab Hasil dan Pembahasan.

3.3. Objek Penelitian

3.3.1. Dataset

Objek utama penelitian adalah dataset "Compost Maturity and Emission Monitoring Dataset" yang diperoleh dari platform Kaggle (Mullick, 2023). Dataset ini berisi data pengukuran sensor dan parameter laboratorium dari proses pengomposan selama 0–32 hari. Spesifikasi dataset disajikan pada Tabel 3.3.

Tabel 3.3 *Spesifikasi Dataset "Compost Maturity and Emission Monitoring Dataset"*

Atribut	Keterangan
Sumber	Kaggle Dataset Repository (Mullick, 2023)
Total baris berkas mentah	1.314 baris
Data digunakan penelitian ini	452 baris data asli (Synthetic=0), 46 batch unik
Data tidak digunakan	862 baris sintetis SMOGN bawaan Kaggle
Periode	0–120 hari proses pengomposan
Fitur sensor mentah	4 (Temperature, MC(%), pH, Ammonia(mg/kg))
Total fitur input (setelah rekayasa)	31 (Arsitektur A) / 27+4+1 (Arsitektur B)
Target	Germination Index (GI(%))
Rentang GI	0,582 – 167,365

Dataset mencakup fitur yang terukur langsung oleh sensor (Temperature, MC(%), pH, Ammonia, Day) dan fitur laboratorium (C/N Ratio, EC, TN, Nitrate) yang pada Arsitektur B diestimasi oleh Stage 1, bukan diukur langsung. Dalam implementasi sistem, hanya empat fitur yang diukur sensor fisik (DS18B20 untuk suhu, *Capacitive Soil Moisture Sensor* untuk kelembapan, Analog pH Sensor, dan MQ-135 untuk amonia).

3.3.2. Teknik Pengumpulan Data

Data primer penelitian diperoleh dari dua sumber. Pertama, log hasil training, validasi, dan *Out-of-Fold* dari seluruh 270 kombinasi eksperimen yang dieksekusi secara lokal. Kedua, data uji latensi inferensi model GI pada Raspberry Pi 4B sebanyak 200 kali run yang dicatat menggunakan *time.perf_counter()* dari modul time Python. Data sekunder berupa dataset kompos diunduh dari Kaggle sesuai prosedur yang telah dijelaskan pada Sub-bab 3.3.1.

3.3.3. Teknik Analisis Data

Evaluasi model dalam penelitian ini menggunakan pendekatan regresi murni: model dilatih dan dievaluasi sebagai regressor yang memprediksi nilai Germination Index (GI) secara kontinu menggunakan metrik RMSE, MAE, dan R^2 (Sub-bab 3.2.6), dihitung secara konsisten pada seluruh 270 kombinasi eksperimen agar perbandingan antar skenario adil. Interval kepercayaan 95% dihitung pada tingkat kombinasi (Model × Augmentasi × Arsitektur ×

HPO_Method) dari 3 nilai fold nested cross-validation. Signifikansi statistik perbandingan Arsitektur A dan B diuji menggunakan *paired t-test* dan *Wilcoxon signed-rank test* pada $\alpha = 0{,}05$. Kontribusi augmentasi, arsitektur, metode HPO, dan pendekatan *ensemble* diisolasi melalui *ablation study* dan perbandingan *apple-to-apple*. Selanjutnya, *feature importance* dari model terbaik dianalisis untuk mengidentifikasi kontribusi tiap fitur terhadap prediksi GI. Kualitas dan potensi pengembangan model dievaluasi melalui diagnostik residual (uji normalitas *Shapiro-Wilk* dan analisis bias) serta l*earning curve*, sedangkan latensi inferensi dianalisis menggunakan statistik deskriptif dan dibandingkan dengan ambang 50 ms.

BAB IV

HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penelitian berbasis rancangan nested cross-validation yang membandingkan dua arsitektur (Single-Stage dan Two-Stage Soft-Sensor), tiga algoritma pembelajaran mesin (CatBoost, Random Forest, XGBoost), empat skema augmentasi data, dan tiga metode hyperparameter optimization yang dievaluasi secara sistematis pada dataset kompos dengan 452 baris data asli (46 batch). Seluruh 270 kombinasi eksperimen (2 arsitektur × 3 model × 5 augmentasi + baseline × 3 metode HPO × 3 fold) berhasil dievaluasi tanpa kegagalan, sehingga memberikan estimasi performa yang komprehensif dan robust.

4.1. Analisis Kebutuhan

4.1.1. Analisis Dataset dan Distribusi Nilai GI

Dataset mentah berjumlah 1.314 baris. Setelah memfilter data sintetis pengujian sebelumnya (kolom Synthetic = 0), diperoleh 452 baris data asli yang berasal dari 46 batch pengomposan berbeda. Tabel 4.1 merangkum statistik deskriptif keempat fitur sensor mentah, fitur usia kompos (Day), dan target GI(%) pada 452 baris data asli tersebut.

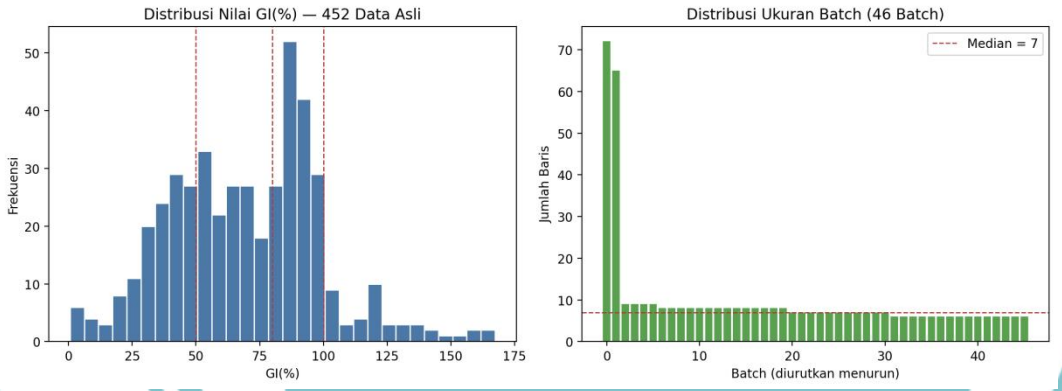
Tabel 4.1 *Statistik Deskriptif Dataset Asli (n = 452)*

Variabel	Min	Mean	Std	Max
Day (hari)	0,00	14,03	12,15	120,00
Temperature (°C)	9,52	34,17	14,16	68,80
MC(%)	11,01	49,11	12,28	79,22
pH	4,28	7,39	0,85	9,37
Ammonia (mg/kg)	3,70	1.268,81	1.884,64	13.407,00
GI(%)	0,58	69,50	29,70	167,36

Tidak ditemukan nilai kosong (missing value) pada seluruh 452 baris data asli untuk 17 kolom dataset, sehingga tidak diperlukan proses imputasi. Konsentrasi Ammonia menunjukkan sebaran yang sangat lebar (3,70 hingga 13.407 mg/kg,

standar deviasi 1.884,64) dan menjadi salah satu motivasi digunakannya transformasi logaritmik (Log_Amm, Sub-bab 3.2.3) untuk menekan pengaruh nilai ekstrem tersebut terhadap proses pelatihan model.

Diagnostik ukuran batch menunjukkan distribusi yang sangat timpang: median 7 baris per batch (kuartil 1 dan 3 masing-masing 6 dan 8 baris), dengan dua batch outlier yang jauh melebihi batch lainnya yaitu Batch 27 (72 baris, rentang Day 0–120) dan Batch 18 (65 baris), sementara batch terkecil hanya berisi 6 baris. Gambar 4.0 (kanan) memvisualisasikan ketimpangan ini secara eksplisit: dua batch pertama menyumbang 137 dari 452 baris (30,3% dari total data) hanya dari 2 dari 46 batch. Ketimpangan ini berisiko menyebabkan GroupKFold konvensional menempatkan batch besar sendirian pada satu fold uji, sehingga skor antar-fold menjadi tidak stabil menjadi motivasi utama perancangan fungsi balanced_group_kfold().



Gambar 4.1 *Distribusi Nilai GI(%) (kiri) dan Ukuran Batch (kanan) pada 452 Data Asli*

Berdasarkan kategorisasi Zucconi, distribusi 452 baris data asli ke dalam empat kategori kematangan disajikan pada Tabel 4.2. Distribusi relatif seimbang antar tiga kategori pertama (masing-masing 28–32%), namun kategori Matang (GI > 100%) jauh lebih sedikit (9,5%). ketidakseimbangan inilah yang mendasari penerapan skema pembobotan sampel multi-tier yang memberi bobot lebih tinggi pada kategori Fitotoksik dan Matang.

Tabel 4.2 *Distribusi Kategori Kematangan (Zucconi) pada 452 Data Asli*

Kategori	Rentang GI(%)	Jumlah Baris	Persentase
Fitotoksik	≤ 50	128	28,3%
Rendah	50 – 80	136	30,1%
Sedang	80 – 100	145	32,1%
Matang	> 100	43	9,5%

Diagnostik lebih lanjut menemukan bahwa 43 baris dengan nilai GI di atas 100 memiliki rata-rata usia kompos (Day) = 26,09, jauh lebih tinggi dibanding rata-rata Day = 12,76 pada baris dengan GI ≤ 100. Mengindikasikan nilai GI di atas 100% terkonsentrasi pada kompos yang lebih matang (Day lebih tinggi), konsisten dengan efek stimulatory yang dilaporkan pada literatur (ekstrak kompos sangat matang kadang mempercepat perkecambahan biji dibanding kontrol air murni). Nilai-nilai ini dipertahankan apa adanya dalam pemodelan (bukan noise yang dibuang atau di-cap), karena mencerminkan fenomena biologis yang sah. Analisis korelasi linear (Pearson) sederhana antara tiap fitur sensor mentah dan target GI. sebelum rekayasa fitur diterapkan untuk memberi gambaran awal arah hubungan tiap fitur terhadap target.

Tabel 4.3 *Korelasi Pearson Fitur Sensor Mentah terhadap GI(%)*

Fitur	Korelasi terhadap GI(%)	Arah Hubungan
Day	0,603	Positif (kuat)
MC(%)	−0,540	Negatif (kuat)
pH	0,273	Positif (lemah)
Temperature	−0,255	Negatif (lemah)
Ammonia(mg/kg)	−0,221	Negatif (lemah)

Usia kompos (Day) menunjukkan korelasi positif terkuat terhadap GI(%) (0,603). semakin lama proses pengomposan berlangsung, semakin tinggi nilai GI, sesuai ekspektasi teoritis kematangan kompos. Kadar air (MC(%)) menunjukkan korelasi negatif terkuat (−0,540). kompos yang terlalu basah cenderung berasosiasi dengan GI lebih rendah, konsisten dengan literatur bahwa kadar air berlebih menghambat aerasi dan aktivitas mikroba aerobik. Perlu dicatat bahwa korelasi Pearson hanya menangkap hubungan linear. Korelasi yang relatif lemah pada Temperature, pH, dan Ammonia tidak berarti fitur-fitur tersebut tidak penting.

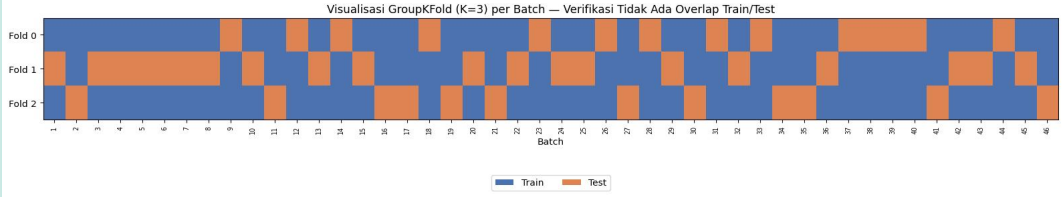
4.1.2. Skema Validasi: Nested Cross-Validation dan Pemilihan K

Penelitian ini menggunakan skema nested cross-validation berbasis grup (Batch) untuk mencegah kebocoran data antar baris dalam batch yang sama serta memberikan estimasi performa yang lebih jujur dan tidak bergantung pada satu pembagian data tertentu. Jumlah fold terluar (K) ditentukan melalui analisis stabilitas empiris menggunakan model proxy Random Forest pada Arsitektur A dengan augmentasi baseline, terhadap kandidat K = {3, 5, 7, 10}, sebagaimana disajikan pada Tabel 4.4.

Tabel 4.4 *Analisis Stabilitas Pemilihan K Terbaik*

K	Mean MAE	Std MAE	Jumlah Fold Score
10	12,936	3,646	30
3	13,696	0,955	9
7	13,511	3,269	21
5	13,616	3,814	15

K = 3 dipilih sebagai jumlah fold terluar karena menghasilkan standar deviasi MAE antar-fold paling kecil (0,955). meskipun K = 10 memiliki mean MAE sedikit lebih rendah, standar deviasinya jauh lebih besar (3,646), mengindikasikan hasil yang kurang stabil/konsisten antar fold akibat ukuran subset data yang lebih kecil per fold. Stabilitas dipilih sebagai kriteria utama dibanding nilai rata-rata semata, karena tujuan penelitian ini adalah memperoleh estimasi performa yang dapat dipercaya, bukan sekadar nilai rata-rata terendah yang mungkin kebetulan pada pembagian fold tertentu. Untuk mengatasi ketimpangan ukuran batch, digunakan fungsi balanced_group_kfold() yang menyeimbangkan total baris per fold, menghasilkan ukuran fold uji yang jauh lebih seimbang: 154, 150, dan 148 baris untuk fold 0, 1, dan 2. Gambar 4.1 memvisualisasikan pembagian Batch ke setiap fold, memverifikasi tidak ada overlap Batch antara data latih dan data uji pada fold manapun.



Gambar 4.2 *Visualisasi Pembagian GroupKFold (K=3) per Batch*

4.2. Perancangan Model

4.2.1. Dua Arsitektur yang Dibandingkan

penelitian ini merancang dan membandingkan dua arsitektur prediksi GI secara langsung:

1. Arsitektur A (Single-Stage): memetakan 31 fitur sensor (termasuk fitur rekayasa Delta, Lag1, RateOfChange, Rolling_mean, dan CumDegreeDay) langsung ke nilai GI(%), tanpa representasi antara.
2. Arsitektur B (Two-Stage Soft-Sensor): mengestimasi empat parameter laboratorium (C/N Ratio, EC, TN, Nitrate) terlebih dahulu dari fitur sensor menggunakan skema Out-of-Fold (OOF) untuk mencegah kebocoran data, kemudian menggunakan estimasi tersebut (bersama T Value hasil komputasi) sebagai fitur tambahan (total 32 fitur) untuk memprediksi GI(%). Pendekatan ini meniru cara laboran menilai kematangan kompos secara bertahap: mengukur parameter kimia terlebih dahulu, baru menyimpulkan kematangan.

4.2.2. Tiga Algoritma Pembelajaran Mesin

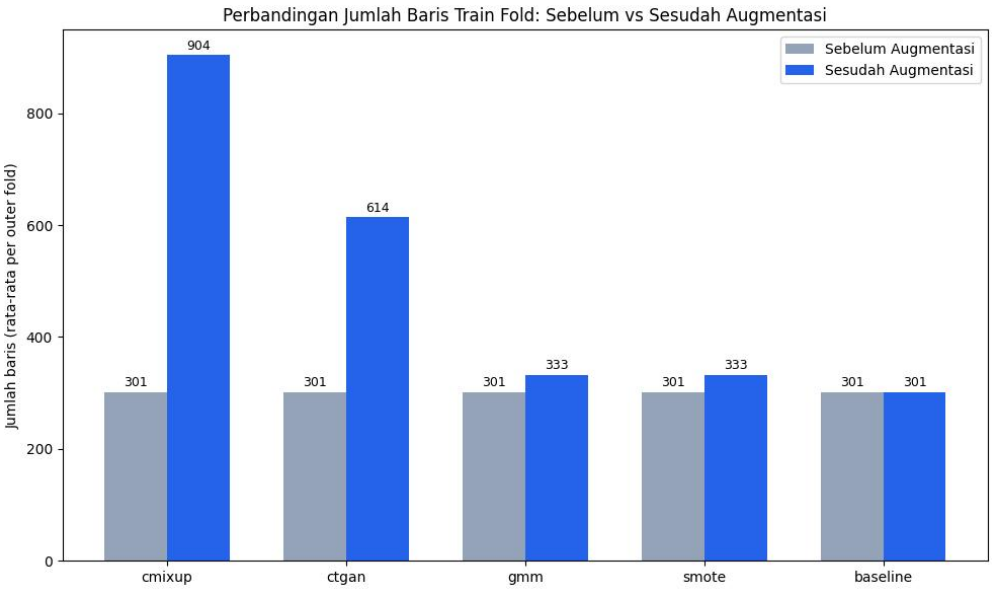
Algoritma yang dibandingkan pada penelitian ini adalah CatBoost, Random Forest, dan XGBoost seluruhnya model berbasis pohon keputusan (tree-based) yang secara empiris lebih sesuai untuk data tabular berukuran menengah (452 baris).

4.2.3. Rancangan Augmentasi Data (Lima Skema)

Lima skema augmentasi data tabular dibandingkan: Baseline (tanpa augmentasi), Gaussian Mixture Model oversampling (GMM), Category-Mixup (C-Mixup), CTGAN (Conditional Tabular GAN), dan Synthetic Minority Oversampling Technique (SMOTE). Tabel 4.5 dan Gambar 4.3 menunjukkan perbandingan jumlah baris data latih sebelum dan sesudah augmentasi (dirata-rata lintas 3 fold luar).

Tabel 4.5 *Jumlah Baris Data Latih Sebelum vs Sesudah Augmentasi*

Augmentasi	N Sebelum	N Sesudah	N Ditambah	Tambahan (%)
cmixup	301	904	603	200
ctgan	301	614	313	103,8
gmm	301	332	31	10,4
smote	301	332	31	10,4
baseline	301	301	0	0



Gambar 4.3 *Perbandingan Jumlah Baris Data Latih Sebelum vs Sesudah Augmentasi*

4.2.4. Tiga Metode Hyperparameter Optimization

Penelitian ini membandingkan tiga metode optimasi hiperparameter dengan anggaran pencarian yang disetarakan, yaitu *Bayesian Optimization*, *Random Search*, dan *Grid Search*. Pada *Bayesian Optimization* dan *Random Search*, jumlah percobaan ditetapkan sebanyak 50 trial, sedangkan ruang pencarian Grid Search dirancang sedemikian rupa sehingga menghasilkan tepat 50 kombinasi untuk setiap model. Sebagai contoh, pada *CatBoost* digunakan 5 nilai iterations, 5 nilai depth, 2 nilai learning_rate, dan 1 nilai tetap untuk l2_leaf_reg, sehingga diperoleh total 50 kombinasi hiperparameter.

Grid Search mengevaluasi seluruh kombinasi hiperparameter yang telah ditentukan secara menyeluruh (*exhaustive search*). Meskipun pendekatan ini menjamin cakupan penuh terhadap ruang pencarian yang didefinisikan,

skalabilitasnya menurun seiring bertambahnya jumlah hiperparameter. Oleh karena itu, satu hiperparameter yang dianggap memiliki pengaruh relatif lebih kecil pada setiap model difiksasi pada satu nilai tertentu agar jumlah kombinasi tetap sebanding dengan metode optimasi lainnya. Kondisi ini berbeda dengan *Bayesian Optimization* dan *Random Search*, yang tetap mengeksplorasi seluruh hiperparameter secara simultan.

Sementara itu, *Random Search* memilih konfigurasi hiperparameter secara acak sebanyak 50 iterasi (n_iter = 50). Pendekatan ini dilaporkan lebih efisien pada ruang pencarian yang kontribusi masing-masing hiperparameternya tidak merata, karena tidak semua hiperparameter memberikan pengaruh yang sama terhadap performa model (Bergstra & Bengio, 2012).

Berbeda dengan kedua metode tersebut, *Bayesian Optimization* menentukan konfigurasi hiperparameter berikutnya berdasarkan hasil evaluasi sebelumnya melalui model surrogate. Dengan memanfaatkan informasi dari percobaan terdahulu, metode ini dapat mengarahkan proses pencarian menuju wilayah yang lebih menjanjikan dan, secara empiris, berpotensi mencapai solusi optimal dengan jumlah percobaan yang lebih sedikit.

4.3. Implementasi dan Eksperimen

4.3.1. Rancangan Eksperimen Nested Cross-Validation

Rancangan eksperimen pada penelitian ini terdiri atas kombinasi 2 arsitektur, 3 algoritma pemodelan, 4 skema augmentasi data ditambah 1 data *baseline*, 3 metode *Hyperparameter Optimization*, dan 3 *outer fold*, sehingga menghasilkan total 270 konfigurasi pengujian. Seluruh kombinasi tersebut berhasil dievaluasi tanpa mengalami kegagalan proses. Keberhasilan pelaksanaan seluruh eksperimen didukung oleh mekanisme *checkpoint-based resume*, yang memungkinkan proses pelatihan dilanjutkan dari titik terakhir apabila terjadi penghentian sistem atau gangguan selama komputasi. Dengan demikian, tidak ada *trial* yang perlu dijalankan ulang dari awal. Selain itu, konfigurasi hiperparameter terbaik untuk setiap kombinasi eksperimen dan setiap *fold* disimpan secara terpisah dalam

berkas lampiran guna menjamin keterlacakan (*traceability*) dan reproduksibilitas hasil penelitian.

4.3.2. Hasil Eksperimen Keseluruhan

Tabel 4.6 menyajikan sepuluh konfigurasi terbaik dari total 270 kombinasi eksperimen berdasarkan nilai *RMSE_mean* terendah yang diperoleh dari rata-rata hasil evaluasi pada tiga *fold* dalam skema *nested cross-validation*. Selain metrik RMSE, tabel tersebut juga menampilkan interval kepercayaan 95% (*95% confidence interval* atau CI95%) untuk nilai RMSE dan R^2, sehingga tidak hanya menggambarkan performa rata-rata model, tetapi juga tingkat ketidakpastian dan konsistensi hasil pengujian antar-*fold*.

Tabel 4.6 *10 Kombinasi Terbaik dengan Confidence Interval 95%*

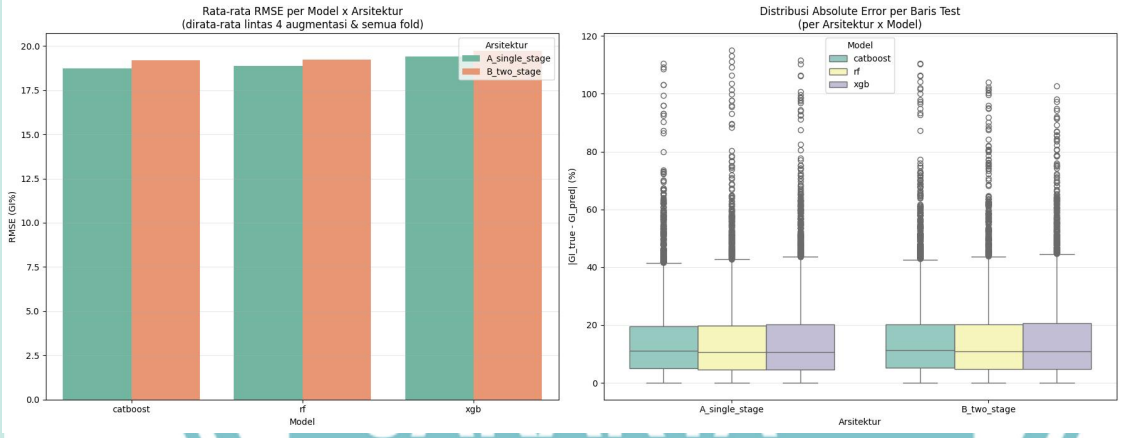
Model	Augmentasi	Arsitektur	HPO	RMSE	Rentang CI95% RMSE	R^2
CatBoost	gmm	A (Single)	random_search	17,729	[16,030 , 19,429]	0,628
CatBoost	baseline	A (Single)	bayesian	17,852	[16,371 , 19,332]	0,627
CatBoost	smote	A (Single)	bayesian	17,852	[16,633 , 19,071]	0,624
CatBoost	baseline	A (Single)	grid_search	18,093	[15,583 , 20,602]	0,614
CatBoost	cmixup	A (Single)	bayesian	18,111	[16,123 , 20,100]	0,618
CatBoost	smote	B (Two-Stage)	bayesian	18,126	[16,923 , 19,329]	0,617
CatBoost	cmixup	A (Single)	random_search	18,268	[17,392 , 19,143]	0,609
RF	cmixup	A (Single)	bayesian	18,304	[18,034 , 18,574]	0,605
CatBoost	baseline	B (Two-Stage)	bayesian	18,353	[17,770 , 18,936]	0,604
RF	ctgan	A (Single)	bayesian	18,388	[18,238 , 18,539]	0,602

Kombinasi dengan kinerja terbaik secara keseluruhan diperoleh oleh CatBoost dengan augmentasi GMM pada Arsitektur A (Single-Stage) yang dioptimalkan menggunakan Random Search. Kombinasi ini menghasilkan nilai RMSE sebesar 17,729 ± 1,502 (CI95%: [16,030; 19,429]), MAE sebesar 13,342 ± 0,930, dan koefisien determinasi (R^2) sebesar 0,628 ± 0,083.

Meskipun demikian, kombinasi yang menempati peringkat kedua dan ketiga, yaitu CatBoost tanpa augmentasi (baseline) dengan RMSE sebesar 17,852 dan

CatBoost dengan augmentasi SMOTE yang juga memperoleh RMSE sebesar 17,852, menunjukkan performa yang secara statistik hampir tidak dapat dibedakan dari kombinasi terbaik. Selisih RMSE sebesar 0,123 jauh lebih kecil dibandingkan simpangan baku RMSE masing-masing konfigurasi yang melebihi 1,0. Temuan ini mengindikasikan bahwa penggunaan augmentasi GMM, tanpa augmentasi, maupun augmentasi SMOTE memberikan tingkat performa yang relatif sebanding pada kombinasi CatBoost dengan Arsitektur A.

Perbandingan yang lebih luas disajikan pada Gambar 4.4, yang menampilkan rata-rata RMSE untuk setiap kombinasi model dan arsitektur setelah dirata-ratakan terhadap seluruh skema augmentasi dan fold. Hasil tersebut menunjukkan bahwa CatBoost secara konsisten menghasilkan performa terbaik pada kedua arsitektur yang diuji. Selain itu, Arsitektur A (Single-Stage) secara umum memberikan nilai RMSE yang lebih rendah dibandingkan Arsitektur B (Two-Stage) pada seluruh model yang dievaluasi.



Gambar 4.4 *Rata-rata RMSE per Model × Arsitektur dan Distribusi Absolute Error*

4.3.3. Analisis Hyperparameter dan Konvergensi Optimasi

Sub-bab ini menganalisis lebih lanjut hyperparameter yang benar-benar terpilih setelah proses optimasi, pola konvergensi Bayesian Optimization selama proses pencarian, serta kontribusi relatif tiap hyperparameter terhadap performa model. Analisis difokuskan pada kombinasi Model × Baseline × Arsitektur A × Bayesian Optimization untuk ketiga algoritma (CatBoost, Random Forest, XGBoost), karena kombinasi ini memiliki catatan hyperparameter yang lengkap

pada seluruh fold dan riwayat trial Optuna yang utuh, sehingga dapat dianalisis secara mendalam tanpa terkendala data yang hilang.

a. Konvergensi Bayesian Optimization, Random Search, dan Grid Search

Sebelum membandingkan ketiga metode *hyperparameter optimization* (HPO) secara langsung, subbab ini terlebih dahulu menyajikan hasil pencarian masing-masing metode secara terpisah. Analisis difokuskan pada rentang nilai hyperparameter yang dieksplorasi, konfigurasi terbaik yang diperoleh, serta rentang nilai yang muncul pada 10% *trial* terbaik. Seluruh analisis dilakukan pada kombinasi augmentasi *baseline* dan Arsitektur A, dengan total 150 *trial* (3 *fold* × 50 *trial*) untuk setiap model. Ringkasan hasil disajikan pada Tabel 4.7, Tabel 4.8, dan Tabel 4.9 untuk *Bayesian Optimization*, *Random Search*, dan *Grid Search* secara berurutan. Perlu diperhatikan bahwa makna 150 *trial* berbeda pada ketiga metode tersebut.

Pada *Bayesian Optimization*, setiap *fold* menjalankan proses optimasi secara independen dan adaptif berdasarkan hasil evaluasi *inner cross-validation*. Oleh karena itu, gabungan 150 *trial* benar-benar merepresentasikan 150 kombinasi hyperparameter yang berbeda. Sebaliknya, pada *Random Search* dan *Grid Search*, hasil pemeriksaan terhadap `cv_results_` menunjukkan bahwa ketiga *fold* mengevaluasi 50 kombinasi hyperparameter yang sama, karena *RandomizedSearchCV* maupun *GridSearchCV* dijalankan dengan *random_state* tetap. Dengan demikian, yang berubah antar-*fold* hanyalah nilai MAE hasil evaluasi, sedangkan kombinasi hyperparameter yang diuji tetap identik.

Tabel 4.7 *Rincian Trial per Hyperparameter untuk Bayesian Optimization (150 Trial per Model)*

Model	Hyperparameter	Rentang Dicoba	Nilai Terbaik	Rentang 10% Terbaik	MAE Terbaik
CatBoost	iterations	50 - 200	200	125 - 200	13,051
CatBoost	depth	4 - 10	7	6 - 9	13,051
CatBoost	learning_rate	0,0107 - 0,2979	0,0712	0,0687 - 0,2938	13,051
CatBoost	l2_leaf_reg	1,001 - 9,964	1,419	1,036 - 7,825	13,051
Random Forest	n_estimators	50 - 200	125	50 - 175	13,066
Random Forest	max_depth	3 - 20	13	10 - 15	13,066
Random Forest	min_samples_leaf	1 - 8	1	1	13,066

Model	Hyperparameter	Rentang Dicoba	Nilai Terbaik	Rentang 10% Terbaik	MAE Terbaik
Random Forest	max_features	log2, sqrt	log2	log2, sqrt	13,066
XGBoost	n_estimators	50 - 200	175	75 - 200	13,756
XGBoost	max_depth	3 - 10	6	5 - 10	13,756
XGBoost	learning_rate	0,0112 - 0,2967	0,0773	0,0381 - 0,2519	13,756
XGBoost	subsample	0,6017 - 0,9994	0,7957	0,7252 - 0,9405	13,756
XGBoost	colsample_bytree	0,6002 - 0,9984	0,6336	0,6009 - 0,7165	13,756

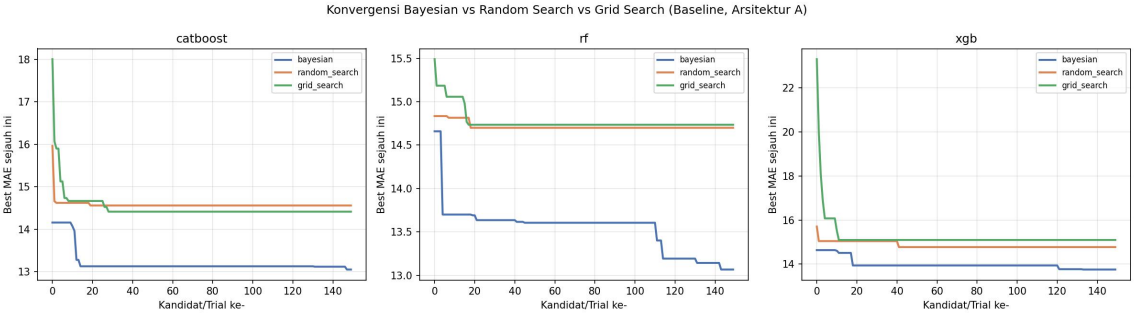
Tabel 4.8 *Rincian Trial per Hyperparameter untuk Random Search (50 Kombinasi × 3 Fold per Model)*

Model	Hyperparameter	Rentang Dicoba	Nilai Terbaik	Rentang 10% Terbaik	MAE Terbaik
CatBoost	iterations	52 - 199	90	52 - 197	14,559
CatBoost	depth	4 - 9	7	4 - 8	14,559
CatBoost	learning_rate	0,0235 - 0,2977	0,0579	0,0379 - 0,1988	14,559
CatBoost	l2_leaf_reg	1,007 - 9,745	3,666	1,063 - 9,337	14,559
Random Forest	n_estimators	50 - 192	64	64 - 192	14,699
Random Forest	max_depth	3 - 19	15	5 - 17	14,699
Random Forest	min_samples_leaf	1 - 7	5	3 - 7	14,699
Random Forest	max_features	log2, sqrt	sqrt	log2, sqrt	14,699
XGBoost	n_estimators	51 - 196	91	51 - 196	14,773
XGBoost	max_depth	3 - 9	4	3 - 9	14,773
XGBoost	learning_rate	0,0102 - 0,2993	0,2232	0,0148 - 0,2984	14,773
XGBoost	subsample	0,6028 - 0,9906	0,7032	0,6260 - 0,9387	14,773
XGBoost	colsample_bytree	0,6126 - 0,9985	0,8703	0,6126 - 0,9637	14,773

Tabel 4.9 *Rincian Trial per Hyperparameter untuk Grid Search (50 Kombinasi × 3 Fold per Model)*

Model	Hyperparameter	Rentang Dicoba	Nilai Terbaik	Rentang 10% Terbaik	MAE Terbaik
CatBoost	iterations	50 - 200	200	50 - 200	14,414
CatBoost	depth	4 - 10	6	4 - 10	14,414
CatBoost	learning_rate	0,03 - 0,2	0,03	0,03 - 0,2	14,414
CatBoost	l2_leaf_reg	3 - 3 (fiksasi)	3	3 - 3	14,414
Random Forest	n_estimators	50 - 200	125	87 - 200	14,734
Random Forest	max_depth	3 - 20	6	6 - 20	14,734
Random Forest	min_samples_leaf	1 - 4	4	4 - 4	14,734
Random Forest	max_features	sqrt (fiksasi)	sqrt	sqrt	14,734
XGBoost	n_estimators	50 - 200	87	50 - 200	15,096
XGBoost	max_depth	4 - 8	4	4 - 8	15,096
XGBoost	learning_rate	0,01 - 0,3	0,05	0,05 - 0,1	15,096
XGBoost	subsample	0,8 - 0,8 (fiksasi)	0,8	0,8 - 0,8	15,096
XGBoost	colsample_bytree	0,8 - 0,8 (fiksasi)	0,8	0,8 - 0,8	15,096

Perbandingan hasil ketiga metode HPO menunjukkan pola yang konsisten dengan performa akhirnya. *Bayesian Optimization* menghasilkan MAE terbaik pada ketiga model, yaitu 13,051 (*CatBoost*), 13,066 (*Random Forest*), dan 13,756 (*XGBoost*). Selain mencapai nilai MAE terendah, metode ini juga cenderung memilih nilai *iterations/n_estimators* yang tinggi, mendekati atau mencapai batas atas ruang pencarian pada *CatBoost* dan *XGBoost*. Sebaliknya, *Random Search* memperoleh konfigurasi terbaik pada nilai *iterations/n_estimators* yang jauh lebih rendah (90 dan 91), menunjukkan bahwa metode ini dapat menemukan kombinasi yang baik secara acak, namun tidak sebaik hasil yang diperoleh melalui pencarian terarah pada Bayesian Optimization. Sementara itu, *Grid Search* menunjukkan performa yang relatif lebih rendah. Hal ini diduga dipengaruhi oleh beberapa hyperparameter yang difiksasi pada satu nilai, yaitu *l2_leaf_reg* = 3 pada CatBoost, *max_features* = 'sqrt' pada *Random Forest*, serta *subsample* dan *colsample_bytree* = 0,8 pada *XGBoost*. Pembatasan tersebut mengurangi luas ruang pencarian dibandingkan Bayesian Optimization maupun Random Search yang mengeksplorasi seluruh kombinasi hyperparameter secara lebih fleksibel. Akibatnya, *Grid Search* menghasilkan MAE tertinggi pada *Random Forest* (14,734) dan XGBoost (15,096).



Gambar 4.5 *Konvergensi Bayesian Optimization vs Random Search vs Grid Search per Trial*

Gambar 4.5 memperlihatkan kurva konvergensi ketiga metode HPO (*best-so-far* per *trial* atau kandidat). Hasilnya menunjukkan bahwa *Bayesian Optimization* secara konsisten mencapai MAE akhir terendah pada ketiga model, dengan selisih sekitar 1,3 hingga 1,7 poin dibandingkan *Random Search* maupun *Grid Search*. Temuan ini dapat disimpulkan bahwa keunggulan *Bayesian Optimization* tidak hanya terlihat pada rata-rata RMSE seluruh kombinasi eksperimen, tetapi juga pada proses optimasi setiap model secara individual. Kurva konvergensi *Bayesian Optimization* pada *CatBoost* dan *Random Forest* masih menunjukkan sedikit penurunan hingga *trial* gabungan ke-150. Namun, pola ini mencerminkan perbedaan hasil optimasi antar-*fold*, bukan karena proses optimasi dalam satu *fold* belum konvergen, mengingat setiap *fold* menjalankan optimasi secara independen dengan anggaran 50 *trial*. Sebaliknya, *Random Search* dan *Grid Search* umumnya mencapai nilai yang mendekati optimum dalam 10 hingga 40 kandidat pertama pada setiap *fold*, sesuai dengan karakteristik kedua metode yang tidak memanfaatkan informasi dari *trial* sebelumnya.

b. Hyperparameter Importance: Bayesian, Random Search, dan Grid Search

Kontribusi relatif setiap hyperparameter terhadap variasi nilai objektif (MAE pada *inner cross-validation*) dianalisis menggunakan metode *functional Analysis of Variance* (fANOVA) yang tersedia pada *Optuna*. Metode ini mendekomposisi variasi nilai objektif menjadi kontribusi masing-masing hyperparameter beserta efek interaksinya. Analisis dilakukan pada gabungan hasil tiga *fold* (150 *trial* untuk setiap kombinasi model dan metode HPO) serta mencakup ketiga metode HPO. Untuk Random Search dan Grid Search, perhitungan menggunakan

mekanisme penyuntikan (*trial injection*) yang sama seperti dijelaskan pada Subbab 4.3.3.a. Ringkasan hasil analisis fANOVA disajikan pada Tabel 4.10.

Tabel 4.10 *Hyperparameter Importance (fANOVA) per Model (Bayesian vs Random Search vs Grid Search)*

Model	Hyperparameter	Bayesian	Random Search	Grid Search
CatBoost	learning_rate	0,386	0,418	0,116
CatBoost	l2_leaf_reg	0,449	0,284	0,000
CatBoost	iterations	0,111	0,203	0,538
CatBoost	depth	0,054	0,095	0,346
Random Forest	max_features	0,565	0,456	0,000
Random Forest	max_depth	0,182	0,193	0,503
Random Forest	min_samples_leaf	0,140	0,123	0,070
Random Forest	n_estimators	0,112	0,229	0,427
XGBoost	max_depth	0,304	0,070	0,088
XGBoost	colsample_bytree	0,291	0,208	0,000
XGBoost	learning_rate	0,181	0,392	0,680
XGBoost	subsample	0,160	0,220	0,000
XGBoost	n_estimators	0,063	0,111	0,231

Hasil pada Tabel 4.10 menunjukkan bahwa beberapa hyperparameter memiliki nilai *importance* tepat 0,000 pada Grid Search, yaitu *l2_leaf_reg* (*CatBoost*), *max_features* (*Random Forest*), serta subsample dan *colsample_bytree* (*XGBoost*). Nilai tersebut bukan merupakan kesalahan perhitungan, melainkan konsekuensi dari rancangan ruang pencarian pada Tabel 3.2, di mana keempat hyperparameter tersebut difiksasi pada satu nilai tetap untuk membatasi jumlah kombinasi *Grid Search menjadi 50, sehingga sebanding dengan anggaran trial pada Bayesian Optimization* dan Random Search (Subbab 4.2.4). Karena parameter tersebut tidak divariasikan, metode fANOVA tidak dapat mengaitkannya dengan variasi nilai MAE sehingga *importance*-nya secara matematis bernilai nol.

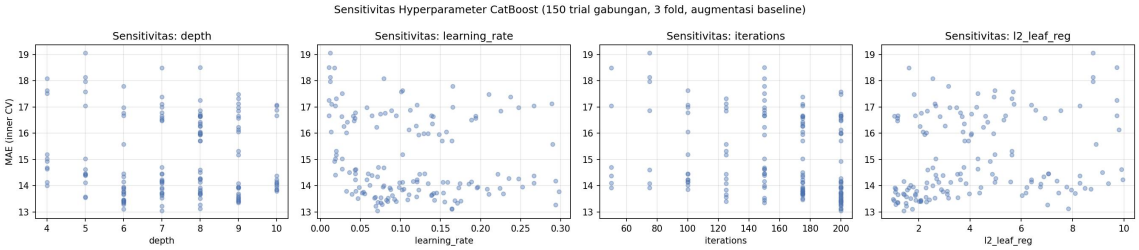
Pada *CatBoost*, *l2_leaf_reg* dan *learning_rate* menjadi hyperparameter yang paling berpengaruh pada *Bayesian Optimization* (0,449 dan 0,386) maupun *Random Search* (0,418 dan 0,284), sedangkan pada *Grid Search* peran terbesar ditunjukkan oleh iterations (0,538). Pada *Random Forest*, max_*features* menjadi

hyperparameter paling dominan pada *Bayesian Optimization* (0,565) dan *Random Search* (0,456), sejalan dengan prinsip *bagging* bahwa keragaman antar-pohon lebih berpengaruh dibandingkan jumlah pohon (*n_estimators*). Namun, karena *max_features* difiksasi pada *Grid Search*, *max_depth menjadi* parameter yang paling dominan (0,503). Sementara itu, pada *XGBoost*, *learning_rate* memiliki *importance* tertinggi pada *Random Search* (0,392) dan *Grid Search* (0,680), sedangkan pada *Bayesian Optimization* peran terbesar justru ditunjukkan oleh *max_depth* (0,304) dan colsample_*bytree* (0,291).

Perbedaan pola *importance* antar-metode HPO dipengaruhi oleh karakteristik proses pencarian masing-masing. *Bayesian Optimization* mengeksplorasi ruang hyperparameter secara adaptif berdasarkan hasil *trial* sebelumnya, sehingga pencarian cenderung terkonsentrasi pada wilayah yang diperkirakan optimal. Sebaliknya, *Grid Search* mengevaluasi seluruh kombinasi dalam ruang pencarian secara sistematis, sedangkan *Random Search* mengambil sampel secara acak tanpa mempertimbangkan hasil *trial* sebelumnya. Perbedaan strategi tersebut menyebabkan estimasi *importance* suatu hyperparameter dapat berbeda meskipun menggunakan model yang sama. Temuan ini menunjukkan bahwa interpretasi *hyperparameter importance* sebaiknya tidak hanya didasarkan pada satu metode HPO. Pola yang muncul secara konsisten pada lebih dari satu metode, seperti dominasi max_features pada Random Forest, memberikan indikasi yang lebih kuat mengenai hyperparameter yang benar-benar berpengaruh terhadap performa model.

c. Sensitivitas Hyperparameter

Untuk memverifikasi bahwa ruang pencarian hyperparameter pada Tabel 3.2 telah memadai dan tidak membatasi proses optimasi hingga nilai terbaik berada pada batas ruang pencarian, dilakukan visualisasi sebaran seluruh 150 trial (3 *fold* × 50 *trial*) CatBoost sebagai metode terpilih terhadap nilai objektif (MAE). Visualisasi ini bertujuan untuk menunjukkan distribusi hasil pencarian pada setiap hyperparameter sekaligus mengevaluasi apakah nilai-nilai optimal terkonsentrasi di dalam rentang pencarian atau justru berada pada batas bawah maupun batas atas ruang pencarian. Hasil visualisasi tersebut disajikan pada Gambar 4.xx.



Gambar 4.6 *Sensitivitas Hyperparameter CatBoost*

Hasil visualisasi menunjukkan bahwa nilai MAE terendah pada parameter depth terkonsentrasi pada rentang 6 hingga 8, bukan pada batas bawah (4) maupun batas atas (10) ruang pencarian. Hal ini mengindikasikan bahwa rentang depth yang digunakan telah mencakup nilai optimal. Pola serupa juga terlihat pada learning_rate, di mana *trial* dengan MAE terendah terkonsentrasi pada kisaran 0,05 hingga 0,20, bukan pada nilai ekstrem 0,01 maupun 0,30. Berbeda dengan kedua parameter tersebut, iterations menunjukkan sebaran yang lebih luas. Nilai MAE rendah diperoleh baik pada kisaran 125 hingga 150 maupun pada batas atas, yaitu 200. Temuan ini konsisten dengan hasil pada Tabel 4.5f yang menunjukkan bahwa iterations = 200 merupakan konfigurasi terbaik yang dipilih oleh Bayesian Optimization. Sementara itu, l2_leaf_reg tidak memperlihatkan hubungan yang jelas terhadap perubahan MAE, sejalan dengan hasil analisis fANOVA yang menunjukkan nilai *importance* relatif rendah (0,131). Secara keseluruhan, visualisasi tidak menunjukkan bahwa nilai-nilai terbaik terkonsentrasi pada batas ruang pencarian. Dengan demikian, ruang pencarian hyperparameter yang digunakan pada Tabel 3.2 dapat dianggap telah memadai, sehingga perluasan rentang pencarian diperkirakan tidak akan memberikan peningkatan performa yang signifikan.

d. Hyperparameter Kombinasi Terbaik Keseluruhan

Selanjutnya dilakukan analisis hyperparameter terpilih antara dua kombinasi terbaik, yaitu *CatBoost + GMM + Random Search* (kombinasi dengan RMSE terendah secara keseluruhan) dan *CatBoost + Baseline + Bayesian Optimization* (kombinasi referensi dengan riwayat trial paling lengkap), pada Tabel 4.11 berikut.

Tabel 4.11 *Perbandingan Hyperparameter: Kombinasi Terbaik Keseluruhan vs Nilai Terbaik CatBoost pada Baseline*

Hyperparameter	CatBoost + GMM + Random Search (RMSE=17,729)	CatBoost + Baseline + Bayesian, Nilai Terbaik (Tabel 4.5f, RMSE=17,852)
iterations	183	200
depth	5	7
learning_rate	0,175	0,0712
l2_leaf_reg	2,871	1,419

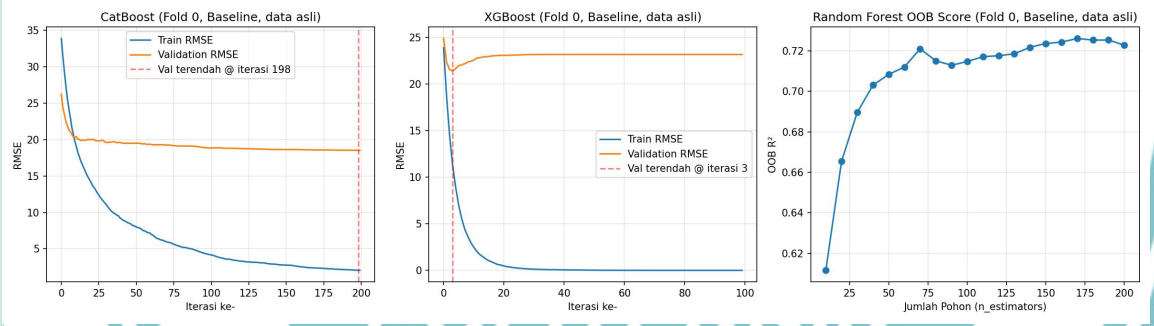
Menariknya, kombinasi terbaik *CatBoost + GMM + Random Search* memilih iterations yang lebih rendah (183 dibandingkan 200) dan depth yang lebih dangkal (5 dibandingkan 7) dibandingkan konfigurasi terbaik pada baseline. Hal ini mengindikasikan bahwa penambahan data sintetis melalui augmentasi GMM memungkinkan model mencapai kemampuan generalisasi yang baik tanpa memerlukan kompleksitas pohon yang terlalu tinggi.

Sebaliknya, *learning_rate* (0,175 dibandingkan 0,0712) dan *l2_leaf_reg* (2,871 dibandingkan 1,419) pada kombinasi terbaik justru lebih tinggi daripada konfigurasi baseline. Meskipun nilai l2_leaf_reg yang lebih besar tampak bertentangan dengan dugaan bahwa data tambahan dapat mengurangi kebutuhan regularisasi, kondisi ini dapat dijelaskan oleh interaksi antar-hyperparameter pada CatBoost. Penggunaan learning_rate yang lebih tinggi dengan jumlah iterations yang lebih sedikit membuat setiap pohon memberikan kontribusi yang lebih besar terhadap proses pembelajaran. Oleh karena itu, peningkatan *l2_leaf_reg* berperan sebagai mekanisme regularisasi untuk menyeimbangkan risiko overfitting akibat strategi pembelajaran yang lebih agresif tersebut.

Temuan ini menunjukkan bahwa hyperparameter pada model *gradient boosting* tidak bekerja secara independen, melainkan saling memengaruhi. Dengan demikian, interpretasi konfigurasi terbaik sebaiknya dilakukan sebagai satu kesatuan pengaturan (*hyperparameter configuration*), bukan berdasarkan masing-masing hyperparameter secara terpisah.

e. Analisis Training Loss per Iterasi (Data Asli)

Sub-bab ini membahas lebih lanjut pengaruh konfigurasi hyperparameter terbaik terhadap dinamika proses pelatihan model, khususnya untuk mengidentifikasi potensi overfitting yang tidak selalu tercermin dari metrik akhir seperti RMSE atau MAE. Sebagai pelengkap analisis konvergensi berdasarkan riwayat *trial Optuna*, dilakukan analisis kurva pembelajaran (*learning dynamics*) menggunakan hyperparameter terbaik hasil HPO pada Fold 0 (Baseline, Bayesian Optimization). Model kemudian dilatih ulang pada data asli sebanyak 452 baris menggunakan skema Balanced Group K-Fold dengan pembobotan sampel *multi-tier*. Gambar 4. menyajikan kurva train-RMSE dan validation-RMSE terhadap jumlah iterasi untuk CatBoost dan XGBoost, serta kurva *out-of-bag* (OOB) R^2 terhadap jumlah pohon pada Random Forest.



Gambar 4.7 *Training Loss per Iterasi (CatBoost, XGBoost) dan OOB Score (Random Forest)*

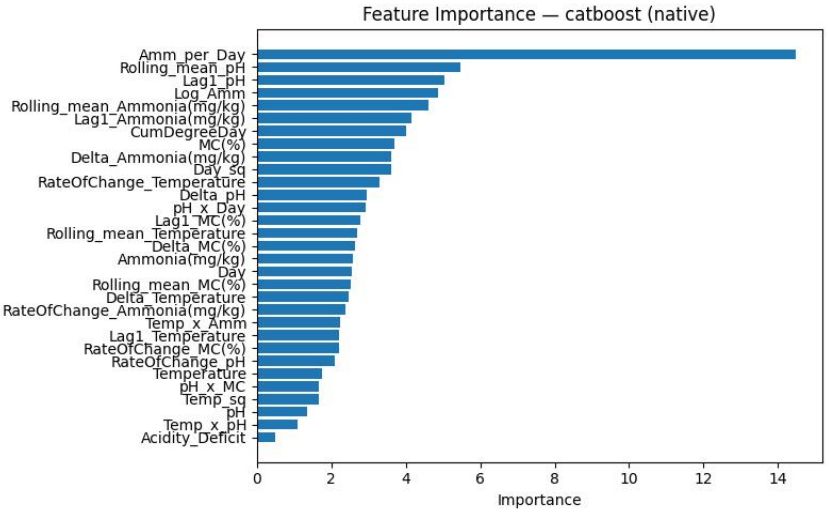
Hasil analisis menunjukkan bahwa model berbasis *boosting* memiliki kecenderungan *overfitting* ketika dievaluasi pada satu fold secara terpisah. *XGBoost* menghasilkan *train RMSE* yang hampir nol (0,001), sedangkan *validation RMSE* tetap tinggi (23,168), sehingga terbentuk selisih sebesar 23,167 yang mengindikasikan model sangat menyesuaikan diri terhadap data latih. *CatBoost* memperlihatkan pola serupa, meskipun dengan tingkat yang lebih rendah, yaitu *train RMSE* sebesar 2,056 dan *validation RMSE* sebesar 18,506 (selisih 16,450). Sebaliknya, Random Forest menunjukkan perilaku yang lebih stabil. Nilai OOB R^2 meningkat dari 0,612 pada 10 pohon menjadi 0,723 pada 200 pohon, kemudian mulai melandai sejak sekitar 140 pohon tanpa menunjukkan indikasi overfitting yang berarti. Pola ini sejalan dengan hasil analisis fANOVA

pada Sub-bab 4.3.3 bagian b yang menunjukkan bahwa *n_estimators* memiliki pengaruh relatif kecil terhadap performa model. Temuan tersebut mengindikasikan bahwa performa model pada satu fold tunggal dapat dipengaruhi oleh variasi komposisi data latih dan data uji, sehingga belum tentu mencerminkan kemampuan generalisasi model secara keseluruhan. Dengan demikian, penggunaan n*ested cross-validation* tiga fold pada penelitian ini menjadi penting karena menghasilkan estimasi performa yang lebih stabil dan representatif dibandingkan evaluasi yang hanya didasarkan pada satu pembagian data.

4.3.4. Feature Importance

Analisis feature importance bawaan (native feature importance) dari model CatBoost terbaik (dengan augmentasi GMM dan Arsitektur A (Single-Stage)) yang dilatih ulang menggunakan seluruh dataset, disajikan pada Gambar 4.8. Hasil analisis menunjukkan bahwa fitur Amm_per_Day (rasio konsentrasi amonia terhadap Day + 1) memiliki kontribusi paling dominan terhadap prediksi model, dengan tingkat pengaruh yang jauh lebih tinggi dibandingkan fitur lainnya. Fitur-fitur dengan kontribusi terbesar berikutnya adalah Rolling_mean_pH, Lag1_pH, Log_Amm, dan Rolling_mean_Ammonia (mg/kg).

Dominasi fitur-fitur yang berkaitan dengan amonia dan pH tersebut sejalan dengan peran biokimia kedua parameter dalam proses pengomposan. Konsentrasi amonia merefleksikan dinamika mineralisasi senyawa nitrogen, sedangkan pH menggambarkan perubahan kondisi kimia yang memengaruhi aktivitas mikroorganisme pengurai. Oleh karena itu, kedua parameter tersebut dapat dianggap sebagai indikator utama yang berkaitan erat dengan perkembangan dekomposisi dan tingkat kematangan kompos.



Gambar 4.8 *Feature Importance — CatBoost (native)*

4.3.5. Penyimpanan Pipeline untuk Deployment

Pipeline final yang terdiri atas RobustScaler dan model CatBoost teroptimasi dilatih ulang (refit) menggunakan seluruh 452 data asli beserta data hasil augmentasi GMM. Model yang telah dilatih kemudian disimpan sebagai artefak siap-deploy dalam berkas best_pipeline_catboost_gmm_A_single_stage.pkl. karakteristik CatBoost sebagai model berbasis *tree-based*, membuat model tidak memerlukan proses konversi ke format khusus sebelum diterapkan pada perangkat edge computing. Artefak yang dihasilkan dapat langsung dimuat dan dijalankan menggunakan pustaka joblib melalui lingkungan Python standar pada perangkat seperti Raspberry Pi.

4.4. Pengujian

4.4.1. Deskripsi dan Prosedur Pengujian

Pengujian pada penelitian ini terdiri atas lima bagian:

1. perbandingan statistik formal antara Arsitektur A dan B,
2. ablation study kontribusi augmentasi dan Stage 1,
3. perbandingan metode HPO dari sisi performa dan efisiensi komputasi,
4. analisis diagnostik error dan learning curve, dan

5. validasi hold-out pada batch yang genuinely tidak pernah dilihat model, ditambah perbandingan pendekatan ensemble multi-model terhadap model tunggal terbaik.

4.4.2. Perbandingan Arsitektur: Uji Statistik Formal

Perbandingan berpasangan (paired) dilakukan pada n = 135 pasangan antara RMSE Arsitektur A dan Arsitektur B (kombinasi Model × Augmentasi × HPO_Method × Fold).Tabel 4.12 merangkum hasil ablation dan uji statistik.

Tabel 4.12 *Perbandingan Arsitektur A vs B dan Uji Statistik*

Arsitektur	RMSE Mean	RMSE Std
A (Single-Stage)	19,008	1,057
B (Two-Stage Soft-Sensor)	19,383	1,040

Hasil pengujian menunjukkan bahwa selisih rata-rata performa antara Arsitektur A dan Arsitektur B (A - B) sebesar −0,375, yang mengindikasikan bahwa Arsitektur A (Single-Stage) menghasilkan kinerja sekitar 1,9% lebih baik dibandingkan Arsitektur B (Two-Stage Soft-Sensor). Uji paired t-test menghasilkan nilai t = -3,795 dengan p = 0,000222, sedangkan uji *Wilcoxon signed-rank* menghasilkan nilai W = 3153 dengan p = 0,001599. Kedua pengujian tersebut menunjukkan hasil yang signifikan pada tingkat signifikansi α = 0,05.

Berdasarkan hasil tersebut, keunggulan Arsitektur A terhadap Arsitektur B dapat dinyatakan signifikan secara statistik dan tidak semata-mata disebabkan oleh variasi acak pada data. Selain itu, penggunaan skema out-of-fold dalam proses estimasi parameter laboratorium pada Stage 1 menjamin bahwa representasi antara yang dihasilkan bebas dari data leakage. Dengan demikian, perbedaan performa yang diamati dapat diatribusikan pada perbedaan rancangan arsitektur, bukan pada kebocoran informasi selama proses pelatihan.

Temuan ini bertentangan dengan hipotesis awal yang mendasari pengembangan pendekatan two-stage pada penelitian sebelumnya. Alih-alih meningkatkan kemampuan prediksi, strategi dekomposisi prediksi GI melalui estimasi parameter kimia pada Stage 1 justru menghasilkan penurunan performa, meskipun dalam skala yang relatif kecil, pada dataset dan pipeline yang digunakan dalam penelitian ini.

4.4.3. Ablation Study: Kontribusi Augmentasi Data

Tabel 4.13 dan Gambar 4.9 (panel kiri) menyajikan hasil ablation terhadap kontribusi augmentasi data (dirata-rata dari seluruh model, arsitektur, HPO, dan fold untuk masing-masing skema augmentasi).

Tabel 4.13 *Ablation Study — Kontribusi Augmentasi Data*

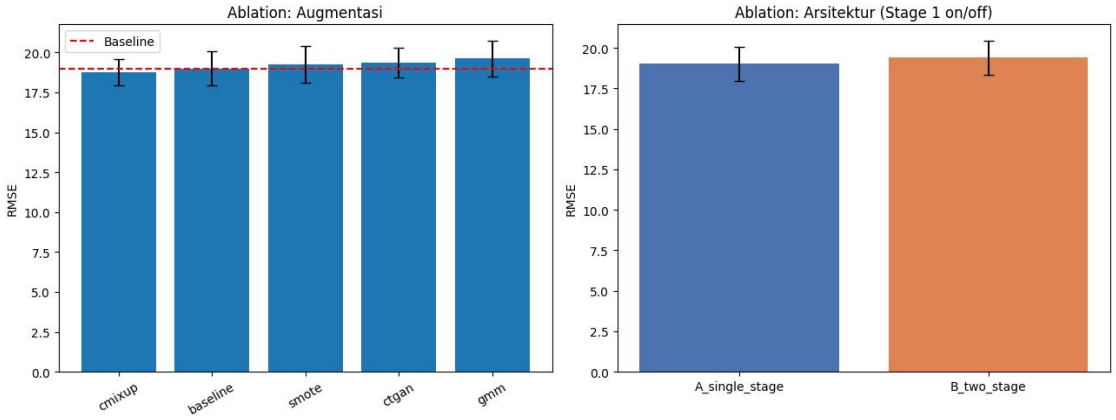
Augmentasi	RMSE Mean	RMSE Std
cmixup	18,777	0,831
baseline (tanpa augmentasi)	18,996	1,085
smote	19,247	1,158
ctgan	19,345	0,937
gmm	19,614	1,108

Jika dirata-ratakan terhadap seluruh kombinasi eksperimen, C-Mixup menghasilkan nilai RMSE terendah, yaitu sebesar 18,777, atau 0,219 lebih rendah dibandingkan skema baseline tanpa augmentasi yang memperoleh RMSE sebesar 18,996. Hasil tersebut menunjukkan bahwa penerapan augmentasi data secara umum memberikan kontribusi positif terhadap performa model.

Sementara itu, SMOTE menempati urutan ketiga dengan nilai RMSE sebesar 19,247, yang sedikit lebih tinggi dibandingkan baseline ketika dievaluasi pada seluruh kombinasi model dan arsitektur. Meskipun demikian, pada kombinasi CatBoost dengan Arsitektur A, performa SMOTE hampir identik dengan baseline sebagaimana ditunjukkan pada Tabel 4.8. Temuan ini mengindikasikan bahwa penambahan sekitar 31 sampel sintetis per fold (10,4% pada Tabel 4.3) melalui interpolasi k-nearest neighbors yang digunakan dalam SMOTE belum mampu memberikan peningkatan performa secara konsisten pada berbagai kombinasi model dan arsitektur, meskipun juga tidak menimbulkan penurunan performa pada konfigurasi terbaik.

Selain itu, hasil penelitian menunjukkan bahwa efektivitas metode augmentasi sangat bergantung pada kombinasi model dan arsitektur yang digunakan. Meskipun C-Mixup memberikan performa rata-rata terbaik secara keseluruhan, kombinasi dengan kinerja tertinggi pada penelitian ini, yaitu CatBoost dengan Arsitektur A, justru dicapai melalui augmentasi GMM. Dengan demikian, tidak terdapat satu metode augmentasi yang secara universal unggul untuk seluruh

skenario pemodelan. Pemilihan strategi augmentasi perlu disesuaikan dengan karakteristik model dan arsitektur yang digunakan.



Gambar 4.9 *Ablation Study: Kontribusi Augmentasi (kiri) dan Arsitektur (kanan)*

4.4.4. Perbandingan Metode HPO: Performa vs Efisiensi Komputasi

Tabel 4.14 dan Gambar 4.10 membandingkan ketiga metode HPO dari sisi performa (RMSE rata-rata total seluruh kombinasi) dan efisiensi komputasi (waktu rata-rata per run serta waktu total).

Tabel 4.14 *Perbandingan Metode HPO — Performa dan Waktu Komputasi*

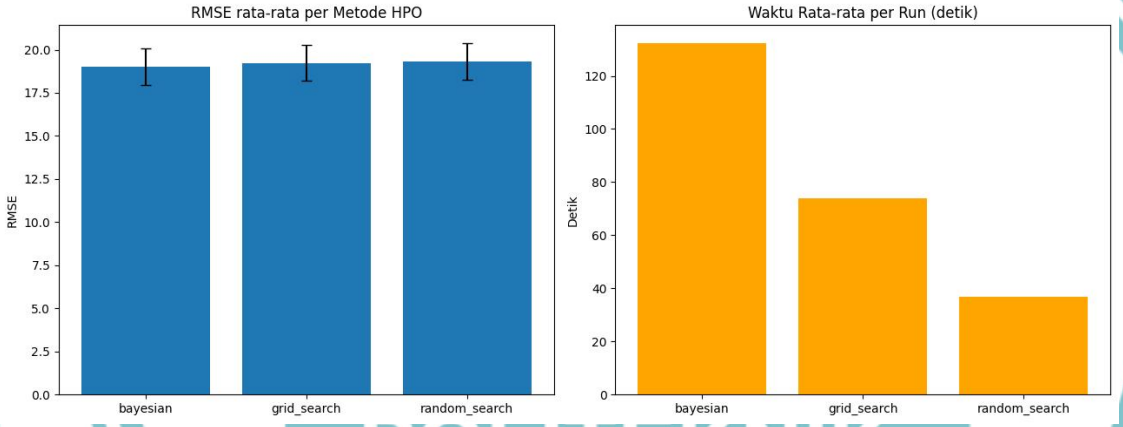
Metode HPO	RMSE Mean	RMSE Std	Waktu/Run (detik)	Waktu Total (menit)
Bayesian	19,034	1,058	132,48	198,72
Grid Search	19,232	1,038	73,80	110,70
Random Search	19,321	1,085	36,72	55,08

Berdasarkan rata-rata seluruh 270 kombinasi eksperimen, Bayesian Optimization menghasilkan performa terbaik dengan nilai RMSE rata-rata sebesar 19,034, diikuti oleh Grid Search sebesar 19,232 dan Random Search sebesar 19,321. Meskipun demikian, keunggulan tersebut diperoleh dengan konsekuensi waktu komputasi yang jauh lebih tinggi. Total waktu yang dibutuhkan oleh Bayesian Optimization mencapai 198,7 menit, sedangkan Grid Search memerlukan 110,7 menit dan Random Search hanya 55,1 menit, atau sekitar 3,6 kali lebih cepat dibandingkan Bayesian Optimization.

Menariknya, kombinasi terbaik secara keseluruhan, yaitu CatBoost dengan augmentasi GMM, justru diperoleh melalui Random Search, bukan melalui

Bayesian Optimization. Temuan ini menunjukkan bahwa keunggulan rata-rata suatu metode optimasi hiperparameter tidak selalu menjamin hasil terbaik pada setiap kombinasi model dan arsitektur. Perbedaan tersebut kemungkinan dipengaruhi oleh variasi performa antar-fold serta ruang pencarian hiperparameter yang masih relatif terbatas, mengingat jumlah percobaan yang digunakan hanya sebanyak 50 trial dari keseluruhan ruang hiperparameter yang jauh lebih luas.

Oleh karena itu, pemilihan metode optimasi hiperparameter dalam praktik tidak sebaiknya didasarkan semata-mata pada performa rata-rata, tetapi juga perlu mempertimbangkan keseimbangan antara kualitas hasil yang diperoleh dan efisiensi komputasi yang dibutuhkan.



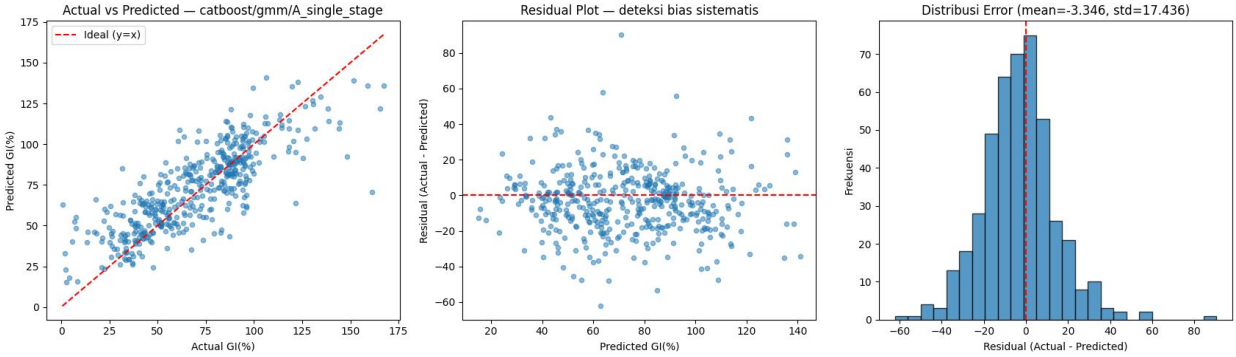
Gambar 4.10 *Perbandingan RMSE dan Waktu Komputasi 3 Metode HPO*

4.4.5. Analisis Error dan Diagnostik Residual

Diagnostik residual untuk model terbaik, yaitu kombinasi CatBoost dengan augmentasi GMM pada Arsitektur A (Single-Stage) yang telah dilatih ulang menggunakan seluruh dataset beserta data hasil augmentasi GMM, disajikan pada Gambar 4.10. Hasil uji normalitas Shapiro-Wilk terhadap residual menghasilkan nilai statistik sebesar 0,978 dengan $p < 0{,}0001$, yang menunjukkan bahwa distribusi residual tidak sepenuhnya mengikuti distribusi normal.

Analisis lebih lanjut menunjukkan bahwa rata-rata residual (mean residual = Actual − Predicted) bernilai −3,346 dengan simpangan baku sebesar 17,436. Nilai rata-rata residual yang negatif mengindikasikan adanya kecenderungan model untuk melakukan over-prediction, yaitu memprediksi nilai GI sedikit lebih tinggi daripada nilai aktual secara rata-rata.

Selain itu, scatter plot antara nilai aktual dan prediksi memperlihatkan bahwa sebagian besar titik observasi terkonsentrasi di sekitar garis ideal (y = x) pada rentang GI rendah hingga menengah. Namun, penyebaran residual cenderung meningkat pada nilai GI yang lebih tinggi, khususnya di atas 100%. Pola ini mengindikasikan adanya peningkatan variabilitas kesalahan prediksi pada area tersebut, yang kemungkinan berkaitan dengan keterbatasan jumlah sampel pada kategori GI ekstrem.



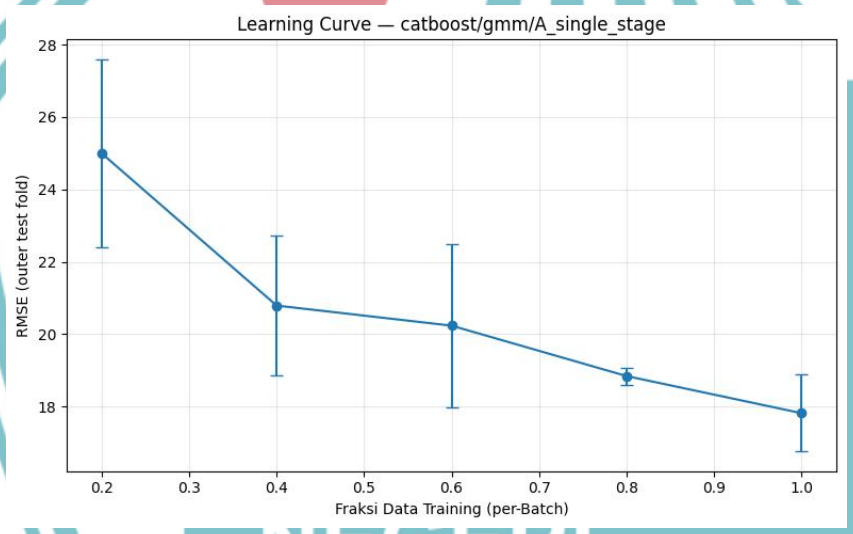
Gambar 4.11 *Analisis Error: Actual vs Predicted, Residual Plot, dan Distribusi Error*

4.4.6. Learning Curve

Gambar 4.12 menyajikan learning curve, yaitu grafik yang menggambarkan perubahan performa model seiring dengan bertambahnya jumlah data latih. Pengujian dilakukan menggunakan lima variasi ukuran data latih, mulai dari 20% hingga 100%, dengan pembagian berdasarkan batch, bukan berdasarkan jumlah baris data.

Hasil pengujian menunjukkan bahwa peningkatan jumlah data latih secara konsisten diikuti oleh penurunan nilai RMSE. Ketika model hanya dilatih menggunakan 20% data, yang setara dengan sekitar tujuh batch pada setiap fold, nilai RMSE yang diperoleh masih relatif tinggi, yaitu 24,996 dengan simpangan baku antar-fold sebesar ±2,605. Seiring bertambahnya jumlah data hingga mencapai 100% atau sekitar 34 batch per fold, nilai RMSE menurun menjadi 17,813 dengan simpangan baku yang lebih kecil, yaitu ±1,061. Temuan ini menunjukkan bahwa penambahan data tidak hanya meningkatkan akurasi model, tetapi juga menghasilkan performa yang lebih stabil dan konsisten pada berbagai fold pengujian.

Selain itu, kurva pembelajaran belum menunjukkan kecenderungan untuk mencapai kondisi jenuh (plateau) pada titik penggunaan data maksimum. Nilai RMSE masih terus menurun seiring penambahan data latih, yang mengindikasikan bahwa kapasitas model kemungkinan belum mencapai batas optimalnya. Oleh karena itu, peningkatan jumlah data observasi asli diperkirakan masih berpotensi meningkatkan performa model secara lebih lanjut. Temuan ini juga menunjukkan bahwa pengumpulan data tambahan di lapangan lebih menjanjikan dibandingkan hanya mengandalkan penambahan data sintetis melalui teknik augmentasi, sehingga dapat menjadi salah satu fokus utama dalam penelitian berikutnya.



Gambar 4.12 *Learning Curve — Pengaruh Fraksi Data Latih terhadap RMSE*

4.4.7. Perbandingan Pendekatan Ensemble vs Model Tunggal

Pendekatan ensemble yang menggabungkan model CatBoost, Random Forest, dan XGBoost dievaluasi pada seluruh lima skema augmentasi menggunakan rancangan pengujian yang setara (apple-to-apple comparison). Pendekatan ini dipilih untuk menghindari bias seleksi terhadap metode augmentasi tertentu yang sebelumnya telah menunjukkan performa terbaik pada model individual.

Bobot masing-masing model dalam ensemble ditentukan melalui proses optimasi menggunakan algoritma Sequential Least Squares Programming (SLSQP) berdasarkan prediksi out-of-fold. Optimasi dilakukan dengan dua kendala, yaitu seluruh bobot harus bernilai non-negatif dan total bobot harus sama dengan satu.

Fungsi objektif yang digunakan adalah minimisasi mean absolute error (MAE) tertimbang, sehingga kombinasi bobot yang diperoleh merepresentasikan kontribusi optimal dari setiap model terhadap performa prediksi gabungan.

Tabel 4.15 *Ringkasan Ensemble per Augmentasi*

Augmentasi	Bobot CatBoost	Bobot RF	Bobot XGB	RMSE Mean	R^2
baseline	0,389	0,534	0,078	17,866	0,624
smote	0,154	0,439	0,407	17,913	0,620
gmm	0,140	0,680	0,180	18,711	0,588
cmixup	0,128	0,000	0,872	18,815	0,586
ctgan	0,573	0,182	0,245	19,063	0,574

Tabel 4.15 menyajikan rata-rata bobot serta performa model ensemble pada masing-masing skema augmentasi. Konfigurasi ensemble terbaik, diperoleh pada skema tanpa augmentasi (baseline), menghasilkan nilai RMSE sebesar 17,866, MAE sebesar 13,147, dan koefisien determinasi (R^2) sebesar 0,624. disusul dengan model ensemble dengan augmentasi SMOTE menghasilkan nilai RMSE sebesar 17,913 dan koefisien determinasi (R^2) sebesar 0,620. Meskipun memberikan performa yang kompetitif, hasil tersebut belum mampu melampaui kinerja model tunggal terbaik, yaitu CatBoost dengan augmentasi GMM, yang mencapai RMSE sebesar 17,729 sebagaimana ditunjukkan pada Tabel 4.6.

Selanjutnya dilakukan perbandingan langsung antara model ensemble dan model tunggal pada skema augmentasi yang sama, sebagaimana disajikan pada Tabel 4.16, menunjukkan bahwa pendekatan ensemble tidak berhasil memperoleh performa terbaik pada satu pun dari lima skema augmentasi yang diuji. Bahkan pada skema SMOTE, yang menghasilkan selisih performa paling kecil, model ensemble masih tertinggal sebesar 0,061 poin RMSE. Secara keseluruhan, baik CatBoost maupun Random Forest sebagai model tunggal tetap menunjukkan performa yang lebih unggul dibandingkan pendekatan ensemble pada seluruh skenario augmentasi yang dievaluasi.

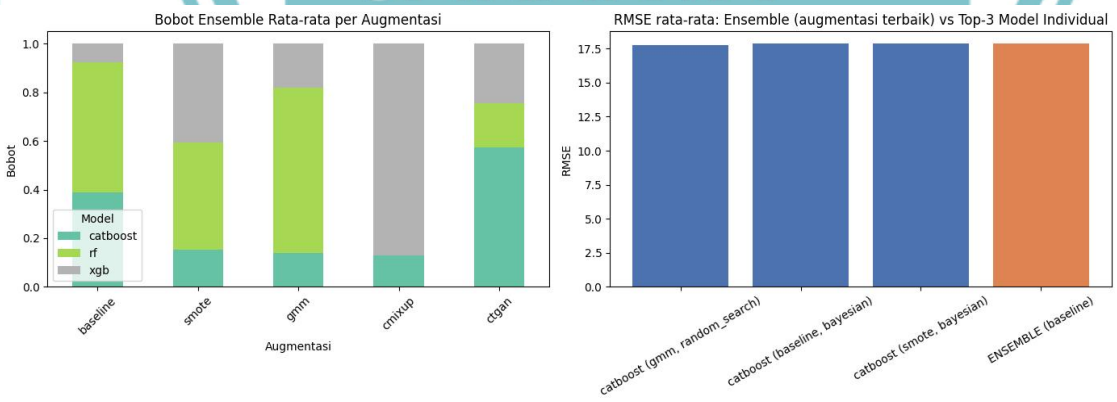
Tabel 4.16 *Perbandingan Apple-to-Apple: Ensemble vs Model Tunggal Terbaik per Augmentasi*

Augmentasi	Model Tunggal Terbaik	RMSE Tunggal	RMSE Ensemble	Pemenang
baseline	CatBoost	17,852	17,866	Model Tunggal
gmm	CatBoost	17,729	18,711	Model Tunggal
ctgan	Random Forest	18,388	19,063	Model Tunggal
cmixup	CatBoost	18,111	18,815	Model Tunggal
smote	CatBoost	17,852	17,913	Model Tunggal

Berdasarkan keseluruhan hasil evaluasi, kombinasi CatBoost dengan augmentasi GMM pada Arsitektur A (Single-Stage) dipilih sebagai arsitektur akhir untuk tahap deployment. Keputusan ini didasarkan pada pertimbangan efisiensi komputasi, mengingat pendekatan tersebut hanya memerlukan satu model untuk proses pelatihan dan inferensi, berbeda dengan metode ensemble yang mengharuskan tiga model dilatih dan dijalankan secara bersamaan.

Selain menawarkan kompleksitas komputasi yang lebih rendah, model CatBoost + GMM juga mampu mempertahankan, bahkan sedikit melampaui, tingkat akurasi yang dicapai oleh pendekatan ensemble pada penelitian ini. Oleh karena itu, kombinasi tersebut dinilai memberikan keseimbangan terbaik antara performa prediksi dan kebutuhan komputasi, sehingga lebih sesuai untuk diterapkan pada lingkungan deployment.

Visualisasi mengenai distribusi bobot ensemble pada masing-masing skema augmentasi, serta perbandingan nilai RMSE antara konfigurasi ensemble terbaik dan tiga model individual dengan performa tertinggi, disajikan pada Gambar 4.13.



Gambar 4.13 *Bobot Ensemble per Augmentasi dan Perbandingan RMSE*

4.4.8. Validasi Hold-out Dataset Independen

Sebagai evaluasi tambahan di luar skema nested cross-validation, dilakukan pengujian menggunakan himpunan data validasi terpisah yang terdiri atas tujuh batch, yaitu Batch 9, 19, 20, 22, 25, 35, dan 37, dengan total 52 baris data. Pemisahan batch dilakukan secara acak menggunakan random number generator dengan seed tetap sebelum proses augmentasi dan pelatihan dimulai.

Dengan rancangan tersebut, seluruh batch validasi benar-benar tidak pernah digunakan pada tahap pelatihan maupun optimasi hiperparameter, sehingga dapat berfungsi sebagai data uji eksternal untuk mengevaluasi kemampuan generalisasi model terhadap data yang sepenuhnya baru.

Selanjutnya, model akhir dilatih ulang (refit) menggunakan 400 baris data yang tersisa, yang berasal dari 39 batch lainnya, dengan tetap mempertahankan konfigurasi hiperparameter dan metode augmentasi GMM yang sama seperti pada model final yang diperoleh dari proses nested cross-validation.

Tabel 4.17 *Hasil Validasi Hold-out vs Estimasi Nested Cross-Validation*

Metode Evaluasi	RMSE [Rentang]	MAE	R^2
Nested Cross-Validation (CI95%)	17,729 [16,030 ; 19,429]	13,342	0,628
Hold-out Independen (7 batch, n=52)	14,311	10,436	0,731

Hasil validasi hold-out menunjukkan performa yang secara numerik lebih baik dibandingkan estimasi yang diperoleh melalui nested cross-validation, dengan nilai RMSE sebesar 14,311, MAE sebesar 10,436, dan koefisien determinasi (R^2) sebesar 0,731. Meskipun demikian, nilai RMSE tersebut berada di luar rentang interval kepercayaan 95% dari hasil nested cross-validation, yaitu [16,030; 19,429].

Sebagai bentuk transparansi metodologis, perbedaan ini mengindikasikan bahwa tujuh batch hold-out yang dipilih secara acak kemungkinan memiliki karakteristik yang relatif lebih mudah diprediksi oleh model dibandingkan keseluruhan dataset. Oleh karena itu, hasil validasi yang lebih baik ini tidak dapat secara langsung diinterpretasikan sebagai bukti peningkatan kemampuan generalisasi model maupun sebagai indikasi terjadinya overfitting, mengingat performa yang diperoleh justru lebih tinggi daripada estimasi cross-validation.

Dengan mempertimbangkan hal tersebut, hasil hold-out tetap dilaporkan sebagai bagian dari evaluasi model, tetapi tidak digunakan sebagai satu-satunya dasar untuk mengklaim tingkat akurasi model. Interpretasi performa akhir tetap mengacu pada hasil nested cross-validation beserta interval kepercayaan 95% yang dinilai lebih representatif dalam menggambarkan variabilitas kinerja model secara keseluruhan.

4.4.9. Simulasi Deployment: Latensi Inferensi dan Akurasi Klasifikasi Kematangan

Sebagai implementasi dari prosedur pengujian latensi yang telah dirancang pada subbab sebelumnya, dilakukan simulasi deployment secara menyeluruh menggunakan pipeline hold-out, yaitu model yang dilatih ulang tanpa melibatkan tujuh batch hold-out. Model tersebut kemudian dijalankan pada seluruh 52 baris data dari ketujuh batch tersebut secara berurutan, sehingga menyerupai kondisi operasional sebenarnya, ketika data sensor diterima dan diproses secara bertahap dari hari ke hari.

Sebelum pengukuran latensi dilakukan, model terlebih dahulu menjalani sepuluh kali proses warm-up untuk meminimalkan pengaruh waktu inisialisasi dan menstabilkan performa eksekusi. Dengan demikian, nilai latensi yang diperoleh diharapkan lebih merepresentasikan waktu inferensi aktual ketika sistem telah berada pada kondisi operasi normal.

Tabel 4.18 *Statistik Latensi Inferensi (n = 52 inferensi)*

Statistik	Nilai
Mean	1,528 ms
Standar Deviasi	0,203 ms
Minimum	1,206 ms
Maksimum	2,114 ms
Persentil ke-95	1,916 ms
Kriteria (< 50 ms)	Terpenuhi (YA)

Hasil pengujian menunjukkan bahwa latensi inferensi rata-rata yang diperoleh hanya sebesar 1,528 ms, jauh di bawah kriteria keberhasilan yang telah ditetapkan, yaitu kurang dari 50 ms. Dengan demikian, sistem memiliki margin kinerja lebih dari 32 kali lipat terhadap ambang yang ditentukan. Seluruh proses inferensi pada

52 data uji (100%) berhasil memenuhi kriteria tersebut tanpa ditemukan satu pun outlier yang melampaui batas latensi.

Latensi yang sangat rendah ini sejalan dengan karakteristik CatBoost sebagai model berbasis pohon keputusan (*tree-based*), yang proses inferensinya hanya melibatkan operasi penelusuran pohon (*tree-traversal*) dengan kebutuhan komputasi yang relatif ringan. Temuan ini semakin memperkuat kesimpulan bahwa penyederhanaan arsitektur model tidak hanya mampu mempertahankan performa prediksi, tetapi juga memberikan keuntungan praktis dari sisi efisiensi komputasi pada perangkat edge computing. Hasil tersebut konsisten dengan temuan sebelumnya bahwa penggunaan model tunggal lebih menguntungkan dibandingkan pendekatan ensemble yang menggabungkan tiga model secara bersamaan.

Selain dievaluasi menggunakan metrik regresi kontinu, seperti RMSE, MAE, dan koefisien determinasi (R^2), performa model dalam simulasi deployment ini juga dianalisis dari perspektif yang lebih praktis bagi pengguna. Evaluasi tambahan dilakukan melalui pengukuran akurasi klasifikasi terhadap empat kategori kematangan kompos, yaitu Fitotoksik, Sedang, Matang, dan Sangat Matang, berdasarkan metode kategorisasi Zucconi. Kategori tersebut diperoleh dengan memetakan nilai GI kontinu hasil prediksi ke dalam kelas-kelas diskrit yang sesuai. Ringkasan akurasi klasifikasi untuk masing-masing kategori disajikan pada Tabel 4.19.

Tabel 4.19 *Akurasi Klasifikasi Kematangan per Kategori (n = 52)*

Kategori	Benar / Total	Akurasi
Fitotoksik (GI ≤ 50)	7 / 11	63,6%
Sedang (50 < GI ≤ 80)	15 / 17	88,2%
Matang (80 < GI ≤ 100)	8 / 12	66,7%
Sangat Matang (GI > 100)	9 / 12	75,0%
Overall	39 / 52	75,0%

Secara keseluruhan, model mencapai akurasi klasifikasi sebesar 75,0%, dengan 39 prediksi yang sesuai dari total 52 sampel uji. Performa klasifikasi relatif seimbang pada seluruh kategori kematangan kompos, meskipun terdapat variasi tingkat akurasi antar kategori. Kategori Sedang menunjukkan performa terbaik dengan akurasi sebesar 88,2%, diikuti oleh kategori Sangat Matang

sebesar 75,0% dan kategori Matang sebesar 66,7%. Sementara itu, kategori Fitotoksik memperoleh akurasi terendah, yaitu 63,6%.

Pola tersebut sejalan dengan karakteristik model yang dikembangkan sebagai model regresi, yaitu memprediksi nilai GI secara kontinu, bukan melakukan klasifikasi kategori secara langsung. Kesalahan klasifikasi yang terjadi, khususnya pada kategori Fitotoksik dan Matang, kemungkinan besar disebabkan oleh nilai prediksi yang berada di sekitar batas ambang kategorisasi Zucconi, yaitu 50%, 80%, dan 100%. Dalam kondisi tersebut, perbedaan prediksi yang relatif kecil dapat menyebabkan perubahan kategori, meskipun nilai GI yang diprediksi sebenarnya masih cukup dekat dengan nilai aktual.

Tabel 4.20 menyajikan ringkasan hasil simulasi pada tingkat batch, termasuk nilai GI dan kategori kematangan pada observasi terakhir dari setiap batch. Informasi tersebut merepresentasikan skenario penggunaan nyata di lapangan, di mana pengguna umumnya lebih memerlukan informasi mengenai status kematangan terkini dari proses pengomposan yang sedang berlangsung.

Tabel 4.20 *Ringkasan Simulasi per Batch Hold-out*

Batch	N	Hari Maks	R^2	MAE	RMSE	Akurasi	GI Akhir	Kategori Akhir
9	6	30	0,676	12,338	14,211	66,7%	85,12	Matang
19	7	32	0,885	4,198	4,298	71,4%	91,56	Sangat Matang
20	7	32	0,909	4,064	4,899	71,4%	99,27	Sangat Matang
22	8	39	0,843	5,918	6,567	87,5%	106,06	Sangat Matang
25	8	39	0,520	14,461	17,068	75,0%	104,09	Sangat Matang
35	8	40	0,860	5,422	7,646	75,0%	89,41	Matang
37	8	40	−0,131	23,541	26,087	75,0%	86,89	Matang

Enam dari tujuh batch hold-out menunjukkan performa prediksi yang baik hingga sangat baik, dengan nilai R^2 berkisar antara 0,520 hingga 0,909 dan akurasi klasifikasi berada pada rentang 66,7% hingga 87,5%. Namun, Batch 37 muncul sebagai satu-satunya kasus dengan performa yang kurang memuaskan, ditunjukkan oleh nilai R^2 sebesar −0,131. Nilai tersebut mengindikasikan bahwa prediksi model pada batch ini lebih buruk dibandingkan pendekatan sederhana yang hanya menggunakan nilai rata-rata, meskipun akurasi klasifikasinya tetap relatif tinggi, yaitu sebesar 75,0%.

Menariknya, Batch 37 merupakan batch yang sebelumnya menjadi dasar penambahan fitur pH_x_Day dan Acidity_Deficit pada tahap rekayasa fitur. Batch ini memiliki karakteristik pH awal yang sangat asam, yaitu 5,76 pada Day = 0. Walaupun kedua fitur tersebut telah ditambahkan untuk menangkap pengaruh kondisi asam terhadap proses pematangan, model masih menunjukkan kecenderungan over-prediction yang cukup besar pada tahap awal pengomposan. Sebagai contoh, pada Day = 0 nilai GI aktual sebesar 7,62 diprediksi menjadi 43,84, sedangkan pada Day = 7 nilai GI aktual sebesar 24,66 diprediksi sebesar 43,24. Hasil tersebut menunjukkan bahwa kondisi pH awal yang sangat rendah pada Batch 37 merupakan edge case yang secara alami sulit digeneralisasi oleh model. Kemungkinan penyebab utamanya adalah keterbatasan variasi data pelatihan, mengingat hanya terdapat satu batch dengan tingkat keasaman awal serendah ini. Akibatnya, model tidak memiliki cukup contoh untuk mempelajari pola pematangan yang terhambat akibat kondisi asam ekstrem secara memadai.

Meskipun demikian, performa model pada tahap akhir Batch 37 mengalami perbaikan yang signifikan. Pada Day = 40, model memprediksi nilai GI sebesar 86,89, sedangkan nilai aktualnya adalah 81,17. Kedua nilai tersebut sama-sama berada dalam kategori Matang, sehingga kesalahan regresi yang cukup besar pada fase awal tidak secara langsung menurunkan akurasi klasifikasi pada kondisi akhir batch yang lebih relevan dalam konteks penggunaan praktis.

Selain Batch 37, Batch 25 juga menunjukkan performa yang relatif lebih rendah dengan nilai R^2 sebesar 0,520. Pada batch ini, model secara konsisten melakukan under-prediction terhadap nilai GI yang sangat tinggi, yaitu pada rentang 120% hingga 127% selama periode Day 27 hingga Day 39, dengan prediksi yang hanya berkisar antara 97% hingga 104%. Pola tersebut konsisten dengan hasil analisis kesalahan pada Subbab 4.4.5, yang menunjukkan bahwa variabilitas prediksi meningkat pada nilai GI ekstrem di atas 100% akibat terbatasnya jumlah sampel pelatihan pada rentang tersebut.

Secara keseluruhan, metrik evaluasi pada 52 sampel hold-out dalam simulasi ini menghasilkan nilai R^2 sebesar 0,745, MAE sebesar 10,127, dan RMSE sebesar 13,934. Nilai tersebut sangat dekat dengan hasil validasi hold-out pada Subbab 4.4.8 (Tabel 4.9), yang menghasilkan R^2 sebesar 0,731, MAE sebesar 10,436, dan

RMSE sebesar 14,311. Perbedaan kecil antara kedua hasil tersebut dapat dianggap wajar karena adanya perbedaan skenario evaluasi, yaitu pemrosesan data secara berurutan per batch pada simulasi deployment dibandingkan evaluasi hold-out konvensional.Konsistensi hasil antara kedua skenario tersebut memberikan indikasi bahwa pipeline yang diekspor untuk keperluan deployment telah dimuat dan dijalankan dengan benar pada lingkungan simulasi perangkat. Selain itu, tidak ditemukan perbedaan numerik yang signifikan akibat kesalahan konversi model maupun ketidaksesuaian proses preprocessing antara lingkungan pelatihan dan lingkungan deployment.

4.4.10. Analisis Kompromi (Trade-off): Kompleksitas vs Akurasi vs Efisiensi Komputasi

Berdasarkan keseluruhan hasil pengujian pada Subbab 4.4, penelitian ini mengidentifikasi empat trade-off utama yang berkaitan dengan kompleksitas model, strategi optimasi, dan performa sistem.

a. Kompleksitas arsitektur (Two-Stage vs Single-Stage)

Penambahan tahap estimasi parameter laboratorium pada Arsitektur B (Two-Stage Soft-Sensor) tidak memberikan peningkatan akurasi. Sebaliknya, hasil pengujian menunjukkan bahwa Arsitektur A (Single-Stage) secara statistik menghasilkan performa yang lebih baik. Temuan ini mengindikasikan bahwa peningkatan kompleksitas arsitektur tidak selalu berbanding lurus dengan peningkatan kemampuan prediksi.

b. Kompleksitas model (ensemble vs model tunggal)

Penggabungan tiga model melalui pendekatan ensemble, yaitu CatBoost, Random Forest, dan XGBoost, beserta optimasi bobot menggunakan metode SLSQP, tidak berhasil melampaui performa model tunggal pada satu pun dari lima skema augmentasi yang diuji. Selain menambah kompleksitas sistem, pendekatan ensemble juga meningkatkan kebutuhan komputasi selama deployment karena proses inferensi harus dilakukan oleh tiga model secara bersamaan. Dengan demikian, peningkatan kompleksitas tersebut tidak diimbangi oleh peningkatan akurasi yang memadai.

c. Metode optimasi hiperparameter (Bayesian Optimization vs Random Search dan Grid Search)

Bayesian Optimization menghasilkan performa rata-rata terbaik di antara ketiga metode optimasi yang dibandingkan. Namun, keunggulan tersebut diperoleh dengan konsekuensi waktu komputasi yang jauh lebih tinggi, yaitu sekitar 4,3 kali lebih lama dibandingkan Random Search. Menariknya, kombinasi terbaik pada penelitian ini justru ditemukan melalui Random Search. Hasil ini menunjukkan bahwa untuk ruang pencarian hiperparameter yang relatif terbatas, metode optimasi yang lebih sederhana dan efisien tetap mampu menghasilkan performa yang kompetitif.

d. Akurasi regresi dan akurasi klasifikasi praktis

Simulasi deployment menunjukkan bahwa performa regresi model dengan nilai R^2 sebesar 0,745 dapat diterjemahkan menjadi akurasi klasifikasi kematangan kompos sebesar 75,0% pada skema empat kategori Zucconi. Meskipun demikian, performa antar-kategori tidak sepenuhnya seragam, dengan kategori Fitotoksik menunjukkan akurasi terendah sebesar 63,6%. Perbedaan ini terutama disebabkan oleh prediksi yang berada di sekitar batas ambang kategori, sehingga perubahan nilai GI yang relatif kecil dapat menghasilkan perubahan kelas. Temuan tersebut menunjukkan bahwa performa regresi yang baik tidak selalu berbanding lurus dengan konsistensi hasil klasifikasi praktis. Oleh karena itu, evaluasi model sebaiknya mempertimbangkan kedua perspektif tersebut secara bersamaan.

Secara keseluruhan, hasil penelitian menunjukkan bahwa model CatBoost dengan augmentasi GMM pada Arsitektur A (Single-Stage) yang dioptimalkan menggunakan Random Search merupakan konfigurasi yang paling seimbang dari segi akurasi, efisiensi komputasi, dan kemudahan implementasi pada perangkat edge computing. Model ini mencapai nilai R^2 sebesar 0,628 pada nested cross-validation, 0,731 pada validasi hold-out, dan 0,745 pada simulasi deployment, dengan akurasi klasifikasi sebesar 75,0%. Selain itu, latensi inferensi rata-ratanya hanya 1,528 ms, jauh di bawah ambang keberhasilan yang ditetapkan, yaitu 50 ms.

.

BAB V

PENUTUP

5.1. Simpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, penelitian ini menghasilkan beberapa simpulan sebagai berikut.

1. Arsitektur A (Single-Stage) terbukti memiliki performa yang lebih baik dibandingkan Arsitektur B (Two-Stage Soft-Sensor) dalam memprediksi Germination Index (GI) menggunakan algoritma berbasis pohon keputusan, yaitu CatBoost, Random Forest, dan XGBoost. Secara rata-rata, Arsitektur A menghasilkan RMSE sebesar 19,008, lebih rendah sekitar 1,9% dibandingkan Arsitektur B yang memperoleh RMSE sebesar 19,383. Perbedaan tersebut terbukti signifikan secara statistik berdasarkan hasil paired t-test ((t = -3,795) ; (p = 0,000222)) dan Wilcoxon signed-rank test ((W = 3153) ; (p = 0,001599)) pada 135 kombinasi pengujian. Hasil ini menunjukkan bahwa penambahan tahap estimasi parameter laboratorium pada Arsitektur B, yang semula diharapkan mampu meningkatkan akurasi prediksi, tidak memberikan keuntungan performa. Sebaliknya, kompleksitas tambahan tersebut justru menyebabkan penurunan akurasi. Dari seluruh kombinasi yang dievaluasi, model terbaik diperoleh dari penggunaan CatBoost dengan augmentasi GMM pada Arsitektur A yang dioptimalkan menggunakan Random Search. Model tersebut mencapai RMSE sebesar 17,729, MAE sebesar 13,342, dan koefisien determinasi R^2 sebesar 0,628 berdasarkan evaluasi nested cross-validation.
2. Model terbaik yang diperoleh kemudian diimplementasikan dan diuji melalui simulasi *deployment* menggunakan tujuh *batch hold-out* yang terdiri atas 52 observasi dan tidak pernah digunakan selama proses pelatihan maupun optimasi hiperparameter. Pengujian ini dirancang untuk merepresentasikan kondisi penggunaan nyata pada sistem smart composter berbasis Raspberry Pi. Hasil simulasi menunjukkan bahwa model mampu mencapai nilai R^2 sebesar 0,745 pada prediksi GI kontinu, dengan akurasi klasifikasi kematangan kompos sebesar 75,0%. Selain itu, waktu inferensi rata-rata yang dibutuhkan

3. model hanya sebesar 1,528 ms, jauh di bawah ambang batas 50 ms yang ditetapkan sebagai kriteria keberhasilan sistem real-time. Temuan ini menunjukkan bahwa model dapat dijalankan secara mandiri pada perangkat edge computing dengan sumber daya terbatas tanpa memerlukan koneksi internet maupun layanan cloud.
4. Evaluasi terhadap lima metode augmentasi data (Baseline, GMM, C-Mixup, CTGAN, dan SMOTE), tiga algoritma pembelajaran mesin (CatBoost, Random Forest, dan XGBoost), serta tiga metode optimasi hiperparameter (Bayesian Optimization, Random Search, dan Grid Search) menghasilkan beberapa temuan penting.
   a) Pertama, penelitian ini menunjukkan bahwa tidak terdapat satu metode augmentasi yang selalu memberikan hasil terbaik pada seluruh kombinasi model dan arsitektur. Secara rata-rata, C-Mixup menghasilkan RMSE terendah pada seluruh eksperimen. Namun, pada kombinasi terbaik penelitian ini, yaitu CatBoost dengan Arsitektur A, augmentasi GMM justru memberikan performa tertinggi. Di sisi lain, SMOTE menghasilkan performa yang mendekati baseline pada kombinasi terbaik, tetapi kontribusinya terhadap peningkatan akurasi secara keseluruhan relatif terbatas.
   b) Kedua, Bayesian Optimization menghasilkan performa rata-rata terbaik, tetapi membutuhkan waktu komputasi yang jauh lebih besar dibandingkan Random Search. Menariknya, kombinasi model terbaik dalam penelitian ini justru ditemukan melalui Random Search. Temuan ini mengindikasikan bahwa metode optimasi yang lebih sederhana masih mampu memberikan hasil yang kompetitif ketika ruang pencarian hiperparameter relatif kecil.
   c) Ketiga, pendekatan ensemble yang menggabungkan CatBoost, Random Forest, dan XGBoost tidak berhasil mengungguli model tunggal pada seluruh skema augmentasi yang diuji. Dengan demikian, peningkatan kompleksitas model tidak selalu menghasilkan peningkatan performa yang sebanding.

d) Keempat, performa klasifikasi menunjukkan variasi antar kategori kematangan kompos. Kategori Sedang memperoleh akurasi tertinggi, sedangkan kategori Fitotoksik memiliki akurasi terendah. Sebagian besar kesalahan klasifikasi terjadi pada sampel yang nilai GI-nya berada di sekitar batas antarkategori. Hal ini wajar karena model yang digunakan dirancang untuk memprediksi nilai GI secara kontinu, bukan secara langsung menentukan kelas kematangan kompos.

Secara keseluruhan, hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu memprediksi tingkat kematangan kompos secara objektif berdasarkan data sensor, sehingga dapat menjadi alternatif terhadap metode penilaian manual yang cenderung subjektif. Keberhasilan tersebut ditunjukkan oleh nilai R^2 sebesar 0,628 pada evaluasi nested cross-validation, R^2 sebesar 0,745 pada simulasi *deployment*, serta akurasi klasifikasi sebesar 75,0%. Analisis *learning curve* juga memperlihatkan bahwa performa model masih terus meningkat seiring bertambahnya jumlah data pelatihan dan belum menunjukkan kondisi jenuh (*plateau*). Dengan demikian, pengumpulan data observasi asli dalam jumlah yang lebih besar berpotensi meningkatkan performa model pada penelitian selanjutnya. Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Dataset yang digunakan berasal dari sumber publik dan belum sepenuhnya merepresentasikan karakteristik sampah organik rumah tangga di Indonesia. Selain itu, implementasi pada Raspberry Pi masih diuji melalui simulasi berbasis data historis dan belum divalidasi pada proses pengomposan nyata menggunakan sensor fisik secara kontinu. Penelitian ini juga menemukan kasus khusus (*edge case*) pada batch dengan kondisi pH awal yang sangat asam, yang sulit diprediksi secara akurat karena kondisi serupa hanya sedikit terwakili dalam data pelatihan.

5.2. Saran

Berdasarkan keterbatasan dan temuan yang diperoleh, beberapa saran untuk penelitian selanjutnya dapat dirumuskan sebagai berikut.

1. Penelitian selanjutnya disarankan untuk membangun dan mengumpulkan dataset secara mandiri dari proses pengomposan sampah organik rumah

tangga di Indonesia, disertai pengujian laboratorium secara berkala. Langkah ini diperlukan agar model dapat dilatih dan divalidasi menggunakan data yang lebih representatif terhadap karakteristik bahan baku dan kondisi pengomposan yang sesungguhnya.

2. Validasi sistem melalui pengujian lapangan secara langsung selama satu siklus pengomposan penuh perlu dilakukan untuk mengevaluasi konsistensi performa model di luar skenario simulasi. Pengujian tersebut juga penting untuk mengamati potensi penurunan akurasi sensor, khususnya sensor pH dan sensor amonia, akibat paparan lingkungan yang panas, lembap, dan korosif dalam jangka panjang.
3. Hasil analisis learning curve menunjukkan bahwa performa model masih terus meningkat seiring bertambahnya jumlah data dan belum mencapai kondisi jenuh (plateau). Oleh karena itu, penambahan data observasi asli perlu menjadi prioritas utama pada penelitian lanjutan, terutama untuk kategori kematangan yang masih kurang terwakili dan kondisi ekstrem, seperti kompos dengan pH awal yang sangat asam.
4. Penelitian berikutnya juga dapat mengeksplorasi metode augmentasi lain maupun kombinasi beberapa metode augmentasi. Hal ini didasarkan pada temuan bahwa efektivitas augmentasi bergantung pada model dan arsitektur yang digunakan, sehingga belum terdapat satu metode yang unggul secara universal.
5. Pengembangan mekanisme online learning atau re-training berkala dapat dipertimbangkan agar model mampu beradaptasi terhadap variasi komposisi sampah organik dari pengguna yang berbeda tanpa memerlukan proses pelatihan ulang secara menyeluruh.
6. Selain pendekatan berbasis pohon keputusan yang digunakan dalam penelitian ini, studi lanjutan dapat mengevaluasi arsitektur pembelajaran mesin lainnya sebagai pembanding untuk memperoleh pemahaman yang lebih komprehensif mengenai metode yang paling sesuai dalam memprediksi tingkat kematangan kompos.

DAFTAR PUSTAKA

Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M., 2019. *Optuna: A Next-generation Hyperparameter Optimization Framework*. [online] Available at: <http://arxiv.org/abs/1907.10902>.

Aylaj, M. and Adani, F., 2023. The Evolution of Compost Phytotoxicity during Municipal Waste and Poultry Manure Composting. *Journal of Ecological Engineering*, 24(6), pp.281–293. https://doi.org/10.12911/22998993/161822.

Bergstra, J. and Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13, pp.281–305. https://doi.org/10.5555/2188385.2188395.

Breiman, L., 2001. Random Forests. *Journal of Machine Learning Research*, 45(1), pp.5–32. https://doi.org/10.1023/A:1010933404324.

Cawley, G.C. and Talbot, N.L.C., 2010. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, 11, pp.2079–2107. https://doi.org/10.5555/1756006.1859921.

Chawla, N. v., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp.321–357. https://doi.org/10.1613/jair.953.

Chen, T. and Guestrin, C., 2016. XGBoost. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM. pp.785–794. https://doi.org/10.1145/2939672.2939785.

Dorogush, A.V., Ershov, V. and Gulin, A., 2018. *CatBoost: gradient boosting with categorical features support*. Available at: <http://arxiv.org/abs/1810.11363>.

Farou, Z., Wang, Y. and Horváth, T., 2024. Cluster-based oversampling with area extraction from representative points for class imbalance learning. *Intelligent Systems with Applications*, 22, p.200357. https://doi.org/10.1016/j.iswa.2024.200357.

Fonseca, J. and Bacao, F., 2023. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10(1). https://doi.org/10.1186/s40537-023-00792-7.

Hidayaturrohman, Q.A. and Hanada, E., 2025. A Comparative Analysis of Hyper-Parameter Optimization Methods for Predicting Heart Failure Outcomes. *Applied Sciences*, 15(6), p.3393. https://doi.org/10.3390/app15063393.

Ji, C., Ma, F., Wang, J. and Sun, W., 2023. A Conditional Entropy Based Feature Selection for Soft Sensor Development in Chemical Processes. *Chemical Engineering Transactions*, 103, pp.61–66. https://doi.org/10.3303/CET23103011.

Khandakar, A., Ashraf, A., Ayari, M.A., Esmaeili, A., Aljarrah, M., Michael, P., Nahiduzzaman, Md., Kibria, H.B., Gerokosta, V.M., Shehbaz, A.A., Al-Mansoori, M.A.R.A. and Khattab, F., 2025. Compost maturity prediction and gas emissions monitoring: A sensor-based and interpretable machine learning approach. *Computers and Electrical Engineering*, 123, p.110115. https://doi.org/10.1016/j.compeleceng.2025.110115.

Kementerian Lingkungan Hidup. (2025). Sistem Informasi Pengelolaan Sampah Nasional (SIPSN). Data komposisi sampah tahun 2025. Tersedia di: SIPSN Komposisi Sampah (Diakses: 7 Juli 2026).

Li, Y., Xue, Z., Li, S., Sun, X. and Hao, D., 2023. Prediction of composting maturity and identification of critical parameters for green waste compost using machine learning. Bioresource Technology, 385, p.129444. https://doi.org/10.1016/j.biortech.2023.129444.

Liu, J., Li, W., Shan, G., Huang, Y., Feng, H., Wang, S. and Bai, C., 2026. Deciphering the dynamic interactions among ammonia emissions and composting parameters in sewage sludge composting using multi-stage machine learning. Journal of Environmental Management, 402, p.129027. https://doi.org/10.1016/j.jenvman.2026.129027.

Mujiyanti, S.F., Aisyah, P.Y., Salsabilla, A.F., Darmawan, T.R. and Rohid, A., 2022. IoT-based for Monitoring and Control System of Composter to Accelerate Production Time of Liquid Organic Fertilizer. IPTEK The Journal of Engineering, 8(2), p.49. https://doi.org/10.12962/j23378557.v8i2.a14081.

Mullick, M.A.O., Mufti, A.H.R.A., Ratul, R.R. and Noor, J., 2025. An Explainable and Scalable Framework for Compost Maturity Prediction Using Ensemble Learning and Conversational AI. In: Proceedings of the 12th International Conference on Next Generation Computing, Communication, Systems and Security. New York, NY, USA: ACM. pp.29–34. https://doi.org/10.1145/3777555.3777574.

Pural, Y.E., 2023. Developing a data-driven soft sensor to predict silicate impurity in iron ore flotation concentrate. Physicochemical Problems of Mineral Processing, 59(5), pp.1–9. https://doi.org/10.37190/ppmp/169823.

Putri, R.E., Maharani, I.P. and Putri, I., 2025. Real-Time Monitoring System for Temperature, Humidity, and pH for Composting Process. Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering), 14(2), p.380. https://doi.org/10.23960/jtep-l.v14i2.380-390.

Rozaq Rais, N.A. and Rokhmah, S., 2025. Smart Composting System: Inovasi Iot Untuk Pengolahan dan Pemantauan Sampah Organik Rumah Tangga. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika, 10(4), pp.3950–3963. https://doi.org/10.29100/jipi.v10i4.9365.

Shi, S., Guo, Z., Bao, J., Jia, X., Fang, X., Tang, H., Zhang, H., Sun, Y. and Xu, X., 2025. Machine learning-based prediction of compost maturity and identification of key parameters during manure composting. Bioresource Technology, 419, p.132024. https://doi.org/10.1016/j.biortech.2024.132024.

Shi, Y., Sardar, M.F., Qiu, L., Li, B., Zia-ur-Rehman, M., Xue, Y., Zhang, X., Zhao, W. and Li, H., 2026. Achieving high-quality and safe compost: a study on multi-optimization of a non-linear system using machine learning. Bioresource Technology, 457, p.135003. https://doi.org/10.1016/j.biortech.2026.135003.

Wan, X., Li, J., Xie, L., Wei, Z., Wu, J., Wah Tong, Y., Wang, X., He, Y. and Zhang, J., 2022. Machine learning framework for intelligent prediction of compost maturity towards automation of food waste composting system. Bioresource Technology, 365, p.128107. https://doi.org/10.1016/j.biortech.2022.128107.

Wang, H., Lin, S., Zhang, H., Guo, D., Liu, D. and Zheng, X., 2023. Batch-fed composting of food waste: Microbial diversity characterization and removal of antibiotic resistance genes. Bioresource Technology, 385, p.129433. https://doi.org/10.1016/j.biortech.2023.129433.

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K., 2019. Modeling Tabular data using Conditional GAN. Available at: <http://arxiv.org/abs/1907.00503>.

Yang, S.J. and Cha, K.J., 2023. Gaussian-Based Minority Oversampling Technique for Imbalanced Classification Adapting Tail Probability of Outliers. https://doi.org/10.2139/ssrn.4445222.

Yao, H., Wang, Y., Zhang, L., Zou, J. and Finn, C., 2022. C-Mixup: Improving Generalization in Regression. Available at: <http://arxiv.org/abs/2210.05775>.

Zou, J., Tang, S., Chen, C., Tao, J., Liang, R., Jiang, X., Hua, Y., Chen, F., Yang, S. and Shen, F., 2026. Machine learning assisted-hyperspectral imaging for in-situ evaluation of compost maturity. Waste Management, 217, p.115496. https://doi.org/10.1016/j.wasman.2026.115496.

Zucconi, F., Pera, A., Forte, M. and Bertoldi, M. de, 1981. Evaluating Toxicity of Immature Compost. BioCycle, 22, pp.54–57.

DAFTAR RIWAYAT HIDUP

Penulis Lahir di salah satu kota di Riau pada tahun 2004 dari pasangan Bapak Sugiarto, SST, MM dan Ibu Dr. Din Nurika Agustina, SST, M.Si. Saat ini penulis berdomisili di daerah Depok. Penulis menuntaskan jenjang sekolah dasar di dua SD berbeda yaitu SD muhammadiyah 09 di Jakarta Timur pada tahun 2010, lalu berpindah ke SDIT Bina Insan Kamil di Depok pada tahun 2012. setelah itu melanjutkan jenjang sekolah menengah pertama di SMP Negeri 1 Depok pada tahun 2016, selanjutnya penulis melanjutkan sekolah pada jenjang sekolah menengah atas di SMAIT Budi Cendekia Islamic School dengan jurusan Ilmu Pengetahuan Alam (IPA) pada tahun 2019 dan lulus pada tahun 2022. Penulis menyelesaikan Pendidikan Diploma Empat (D4) di Politeknik Negeri Jakarta dengan Program Studi Teknik Multi Media dan Jaringan pada tahun 2026.

Penulis memiliki dedikasi yang kuat pada Internet of Things dan sistem embedded, didukung oleh pengalaman sebagai IoT Engineer pada proyek Smart Trash Bin (PT Karinova Skala Indonesia) dan Telkom Direktorat IT Digital, mencakup integrasi hardware-software, pengembangan model AI, hingga peningkatan efisiensi operasional berbasis data real-time.