

**MODEL XGB-AO: OPTIMASI XGBOOST DENGAN *AQUILA*  
OPTIMIZER UNTUK KLASIFIKASI PRODUK LARIS**

**SKRIPSI**

**OLEH:**

**Stevi Freshia Sihombing**

**228160029**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2026**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

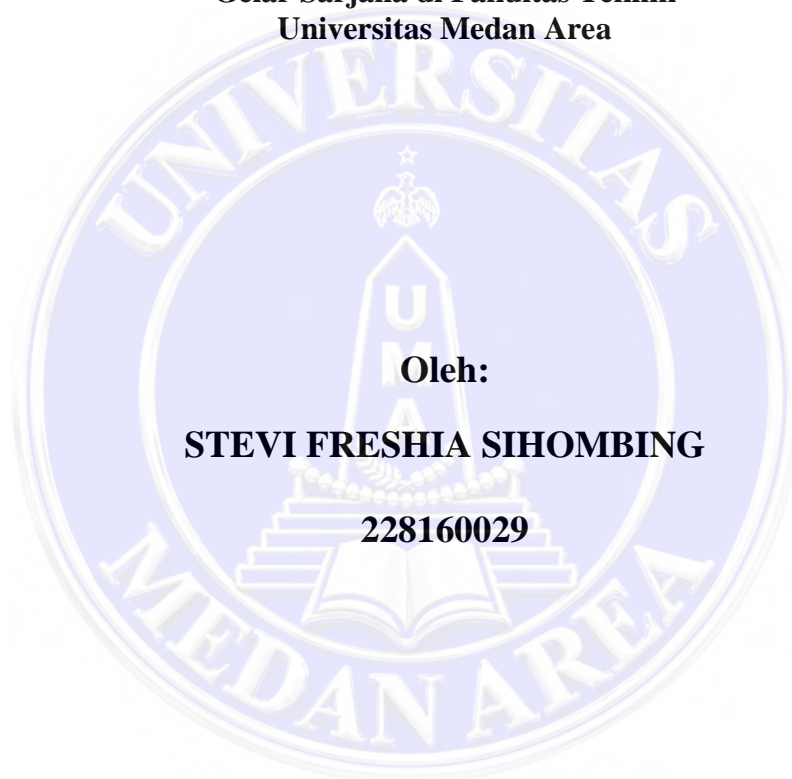
Document Accepted 1/7/26

Access From (repository.uma.ac.id)1/7/26

# **MODEL XGB-AO: OPTIMASI XGBOOST DENGAN AQUILA OPTIMIZER UNTUK KLASIFIKASI PRODUK LARIS**

## **SKRIPSI**

**Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area**



**Oleh:**

**STEVI FRESHIA SIHOMBING**

**228160029**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2026**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

## HALAMAN PENGESAHAN

Judul Skripsi : **MODEL XGB-AO: OPTIMASI XGBOOST DENGAN  
AQUILA OPTIMIZER UNTUK KLASIFIKASI PRODUK  
LARIS**

Nama : Stevi Freshia Sihombing

NPM : 228160029

Fakultas : Teknik

Disetujui Oleh  
Komisi Pembimbing

Andre Hasudungan Lubis S.Ti., M.Sc  
Pembimbing I



Dipriatno, S.T., M.T  
Dekan Fakultas Teknik



Wahono, S.Kom., M.Kom  
Ka. Prodi

Tanggal Lulus: 26 Februari 2026

## HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 26 Februari 2026



Stevi Freshia Sihombing  
228160029

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Stevi Freshia Sihombing  
NPM : 228160029  
Program Studi : Teknik Informatika  
Fakultas : Teknik  
Jenis Karya : Skripsi

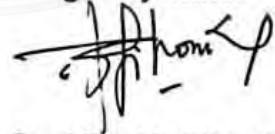
Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**MODEL XGB-AO: OPTIMASI XGBOOST DENGAN AQUILA OPTIMIZER  
UNTUK KLASIFIKASI PRODUK LARIS**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan  
Pada tanggal : 26 Februari 2026  
Yang menyatakan



Stevi Freshia Sihombing  
228160029

## RIWAYAT HIDUP

Stevi Freshia Sihombing lahir di Tarutung pada tanggal 17 Januari 2004. Penulis merupakan anak pertama dari empat bersaudara, dari pasangan Bapak Fernando Sihombing dan Ibu Kristina Hutabarat. Penulis memulai pendidikan formal di TK Negeri Pembina Tarutung pada tahun (2009-2010). SD Negeri 175782 Pollung pada tahun (2010-2016). SMP Negeri 4 Pollung pada tahun (2016-2019). Dan pendidikan menengah atas di SMA Negeri 1 Pollung pada tahun (2019-2022). Pada tahun 2022, penulis melanjutkan pendidikan di Universitas Medan Area dengan mengambil Program Studi Teknik Informatika. Selama menjalani masa perkuliahan, penulis ikut serta dalam beberapa kegiatan akademik dan nonakademik. Pada bulan Juli 2025, penulis bergabung dalam Himpunan Mahasiswa Informatika (HMIF). Pada bulan Juli 2025, penulis menjalani kegiatan Kerja Praktek di Kantor Notaris/PPAT Rachmansyah Purba, S.H., M.Kn Kota Pematang Siantar. Pada bulan Agustus 2025, penulis menerima beasiswa dari Bank Indonesia dan bergabung dengan Komunitas GenBI Sumatra Utara. Pada bulan Oktober 2025, penulis mengikuti Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi (GEMASTIK XVIII 2025). Pada bulan November 2025, penulis melaksanakan penelitian di Laboratorium Universitas Medan Area sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dengan mengambil judul “Model XGB-AO: Optimasi XGBoost dengan *Aquila Optimizer* untuk Klasifikasi Produk Laris”.

## KATA PENGANTAR

Puji dan syukur dipanjatkan kepa Tuhan Yang Maha Esa atas segala berkat dan rahmat-Nya sehingga penulis dapat melaksanakan kegiatan penelitian dan menyelesaikan skripsi dengan judul “Model XGB-AO: Optimasi XGBoost dengan *Aquila Optimizer* untuk Klasifikasi Produk Laris” yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Fakultas Teknik Universitas Medan Area.

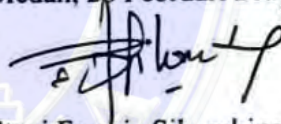
Penulis menyadari akan berbagai kekurangan atau ketidaksempurnaan dari skripsi yang disebabkan keterbatasan pengetahuan penulis, untuk itu berbagai kritik dan saran yang bersifat membangun demi penyempurnaan skripsi ini akan sangat penulis harapkan. Semoga skripsi ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Maka dari itu, penulis ingin mengungkapkan rasa terima kasih yang sebesar-besarnya kepada pihak yang telah membantu penulis dalam proses penelitian ini, yaitu:

1. Tuhan Yang Maha Esa atas kasih, penyertaan, dan berkat-Nya sehingga setiap keraguan dan kesulitan bisa penulis lewati.
2. Kedua orang tua tercinta Bapak Fernando Sihombing dan Ibu Kristina Hutabarat, yang telah memberikan kasih sayang, nasihat, motivasi, memenuhi kebutuhan penulis, dukungan serta doa yang tiada henti kepada penulis selama melalui proses perkuliahan, penulisan skripsi sampai saat ini. Harapan yang selalu penulis panjatkan “Semoga Tuhan menyertai dan memberkati Bapak dan Mama di dalam mengajari dan menuntun kami anak-anak yang telah Tuhan titipkan”.
3. Bapak Prof. Dadan Ramdan, M. Eng., M. Sc, selaku Rektor Universitas Medan Area
4. Bapak Dr.Eng. Supriatno, ST., MT, selaku Dekan Fakultas Teknik Universitas Medan Area
5. Bapak Rizki Muliono, S.Kom., M.Kom selaku Kepala Program Studi Teknik Informatika Universitas Medan Area
6. Bapak Andre Hasudungan Lubis, S.Ti., M.Sc, selaku Dosen Pembimbing yang telah memberikan bimbingan dan arahan kepada penulis.
7. Bapak dan Ibu dosen Fakultas Teknik yang telah memberikan ilmunya kepada penulis, semoga Bapak dan Ibu dosen selalu dalam rahmat dan lindungan Tuhan Yang Maha Esa. Sehingga ilmu yang telah diajarkan dapat bermanfaat dikemudia hari.
8. Seluruh staff Fakultas Teknik Universitas Medan Area, terkhusus kepada Bang Robby Kurniawan Sari Damanik, S.T selaku staff Program Studi Teknik Informatika.

9. Kepada adik-adik penulis Sherin Fiona Sihombing, Saskia Roito Rezeki Sihombing, dan Santika Putri Sihombing yang terkasih dan tersayang yang telah memberikan semangat, dan dukungan kepada penulis selama penulisan skripsi hingga saat ini.
10. Sahabat terkasih Nora Irawani Siregar dan Cessy Cindi Regina Situmorang untuk setiap dukungan, bantuan, dan dorongan positif yang telah diberikan selama proses ini, kontribusi kalian sangat berarti dalam perjalanan ini.
11. Seluruh teman-teman Teknik Informatika stambuk 22 yang turut membantu dalam proses penulisan skripsi.
12. Seluruh teman dekat dan pihak yang tidak bisa penulis sebutkan namanya satu persatu. Terima kasih sudah membantu dan memotivasi penulis.
13. Terimakasih untuk diri sendiri yang selalu berjuang, bertahan, tetap konsisten, dan tidak menyerah walaupun menghadapi banyak tantangan. Dalam sunyi, lelah, dan segala keraguan, tetap berusaha bangkit dan menyelesaikan apa yang telah dimulai. Serahkanlah perbuatanmu kepada Tuhan, maka terlaksanalah segala rencanamu (Amsal 16:3).

Medan, 26 Februari 2026



Stevi Freshia Sihombing  
228160029

## ABSTRAK

Extreme Gradient Boosting (XGBoost) merupakan algoritma yang dikenal efektif dalam menyelesaikan tugas klasifikasi melalui mekanisme boosting berbasis pohon keputusan. Meskipun menawarkan performa tinggi, XGBoost memiliki jumlah *hyperparameter* yang kompleks sehingga kinerjanya sangat bergantung pada proses pencarian konfigurasi yang optimal. Untuk itu, diperlukan metode optimasi yang lebih efektif agar model dapat mencapai performa terbaik. Penelitian ini bertujuan memperkenalkan model XGB-AO, yakni menggunakan algoritma optimasi Aquila Optimizer (AO) pada XGBoost untuk memperoleh kombinasi *hyperparameter* terbaik, kemudian dilatih dan dievaluasi. Pada penelitian ini, model tersebut dibandingkan kinerja serta efektivitasnya dengan XGBoost *Vanilla* dan XGBoost yang dioptimasi menggunakan *Grid Search* serta *Random Search* dalam klasifikasi produk laris pada tiga dataset berbeda. Hasil penelitian menunjukkan bahwa XGBoost *Vanilla* memperoleh performa yang cukup baik, namun dengan adanya optimasi menggunakan AO memberikan peningkatan signifikan. XGB-AO mencapai nilai akurasi sebesar 94,07% pada dataset pertama dan 92,18% pada dataset ketiga. Namun, model XGB-AO masih belum optimal pada dataset kedua yang disebabkan oleh minimnya jumlah data. Meskipun demikian, secara keseluruhan model XGB-AO unggul pada dua dari tiga dataset yang diuji, melampaui performa *Grid Search* dan *Random Search*. Sehingga dapat disimpulkan bahwa metode AO mampu meningkatkan kinerja model XGBoost secara efektif, khususnya pada dataset dengan pola yang lebih stabil. Model XGB-AO terbukti lebih unggul dan efektif sehingga berpotensi baik untuk prediksi produk laris. Penelitian selanjutnya disarankan untuk membandingkan penggunaan algoritma metaheuristik lainnya, serta melakukan evaluasi terhadap efisiensi komputasi.

**Kata Kunci:** *XGBoost, Aquila Optimizer, Optimisasi Hyperparameter, Klasifikasi Produk Laris, Machine Learning.*

## ABSTRACT

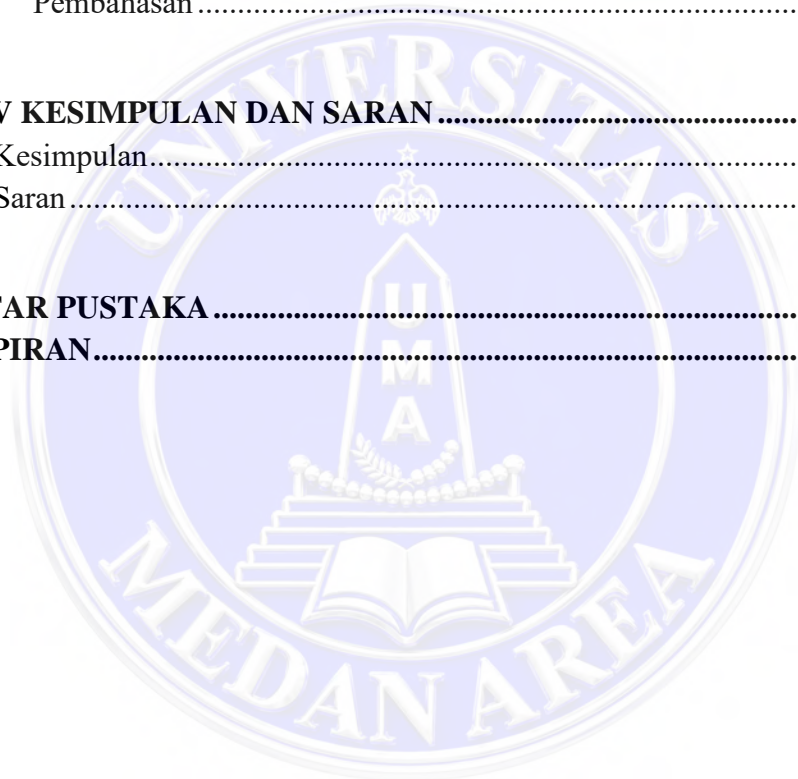
*Extreme Gradient Boosting (XGBoost) is an algorithm widely recognized for its effectiveness in solving classification tasks through a decision-tree-based boosting mechanism. Despite its strong performance, XGBoost involves a large number of complex hyperparameters, making its accuracy highly dependent on the optimal configuration search process. Therefore, a more efficient optimization approach is required to ensure the model achieves its best performance. This study propose the novel XGB-AO model, which integrates the Aquila Optimizer (AO) algorithm into XGBoost to determine the optimal hyperparameter combination prior to model training and evaluation. The proposed model is compared against three baselines, including Vanilla XGBoost, Grid Search–XGBoost, and Random Search– XGBoost for the task of classifying fast-selling products across three different datasets. The results show that the default XGBoost already yields satisfactory performance. However, applying AO optimization provides more significant improvement. XGB-AO achieves an accuracy of 94.07% on the first dataset and 92.18% on the third dataset. Nevertheless, its performance on the second dataset remains suboptimal due to the limited amount of available data. Overall, XGB-AO outperforms the other models on two out of the three datasets, surpassing both Grid Search and Random Search. These findings suggest that the AO is effective in enhancing XGBoost performance, particularly on datasets with more stable patterns. The XGB-AO model demonstrates superior accuracy and reliability, indicating strong potential for fast-moving product prediction. Future research is recommended to explore other metaheuristic algorithms utilization and investigate computational efficiency across different optimization strategies.*

**Keywords:** *XGBoost, Aquila Optimizer, Hyperparameter Optimization, Bestselling Product Classification, Machine Learning.*

## DAFTAR ISI

<b>HALAMAN PENGESAHAN</b> .....	<b>ii</b>
<b>HALAMAN PERNYATAAN</b> .....	<b>iii</b>
<b>RIWAYAT HIDUP</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>ABSTRAK</b> .....	<b>viii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>DAFTAR ISI</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiii</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xiv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	7
1.5. Manfaat Penelitian.....	7
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>8</b>
2.1. Klasifikasi.....	8
2.2. Evaluasi Performa Model Klasifikasi.....	9
2.3. Algoritma <i>Gradient Boosting</i> .....	11
2.3.1. XGBoost.....	12
2.4. <i>Hyperparameter Gradient Boosting</i> .....	14
2.5. Optimasi <i>Hyperparameter</i> .....	15
2.6. <i>Aquila Optimizer</i> .....	16
2.7. Penelitian Terdahulu.....	20
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>22</b>
3.1. Tahapan Penelitian .....	22
3.2. Pengumpulan Data .....	23
3.2.1. Dataset A.....	23
3.2.2. Dataset B .....	24
3.2.3. Dataset C .....	24
3.3. Pemodelan ( <i>XGBoost-Aquila Optimizer</i> ) .....	26
3.4. Implementasi .....	27
3.5. Parameter <i>Aquila Optimizer</i> .....	29

3.6.	Evaluasi .....	30
3.7.	Skenario Penelitian.....	30
<b>BAB IV HASIL PENELITIAN.....</b>		<b>32</b>
4.1.	Hasil dari Model XGB Vanilla.....	32
4.2.	Hasil dari Model XGBoost GS.....	33
4.3.	Hasil dari Model XGboost RS.....	35
4.4.	Hasil dari Model XGBoost AO .....	37
4.5.	Analisis Statistik.....	40
4.6.	Hasil <i>Benchmarking</i> .....	42
4.7.	Analisis <i>Effect Size</i> .....	45
4.8.	Pembahasan .....	46
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>48</b>
5.1	Kesimpulan.....	48
5.2	Saran .....	49
<b>DAFTAR PUSTAKA .....</b>		<b>50</b>
<b>LAMPIRAN.....</b>		<b>56</b>

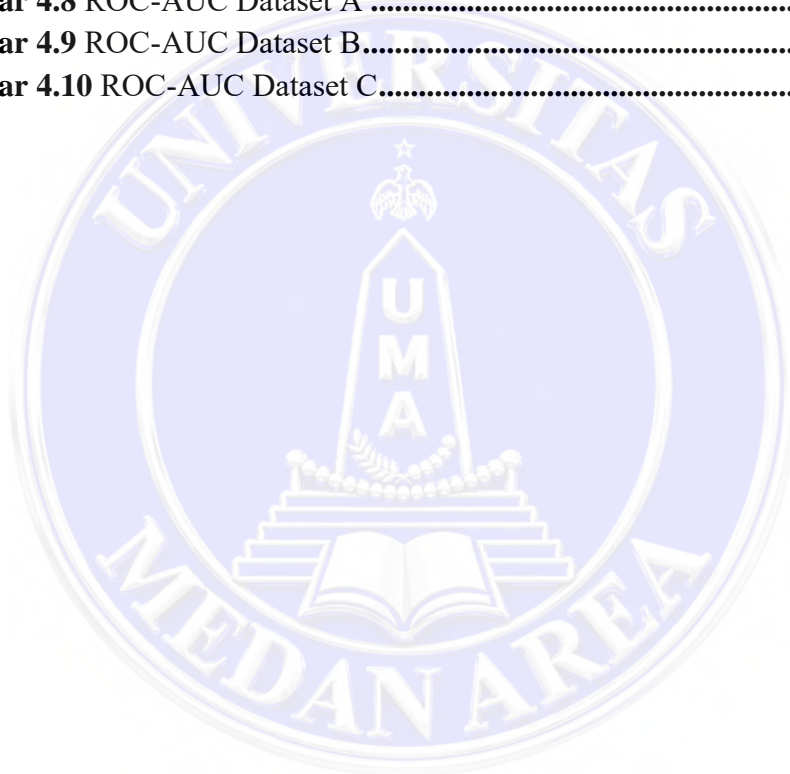


## DAFTAR TABEL

<b>Tabel 2.1</b> Penelitian Terdahulu.....	<b>21</b>
<b>Tabel 3.1</b> Informasi Data Penelitian.....	<b>23</b>
<b>Tabel 3.2</b> Dataset A .....	<b>25</b>
<b>Tabel 3.3</b> Dataset B .....	<b>25</b>
<b>Tabel 3.4</b> Dataset C .....	<b>25</b>
<b>Tabel 3.5</b> Rentang Nilai Hyperparameter.....	<b>28</b>
<b>Tabel 3.6</b> Rangkuman Skenario Penelitian .....	<b>31</b>
<b>Tabel 4.1</b> Rangkuman Hasil Performa Model XGB Vanilla.....	<b>32</b>
<b>Tabel 4.2</b> Nilai Hyperparameter yang dihasilkan Grid Search.....	<b>34</b>
<b>Tabel 4.3</b> Rangkuman Hasil Performa Model XGB Grid Search .....	<b>34</b>
<b>Tabel 4.4</b> Nilai Hyperparameter yang dihasilkan Random Search .....	<b>36</b>
<b>Tabel 4.5</b> Rangkuman Hasil Performa Model XGB Random Search.....	<b>36</b>
<b>Tabel 4.6</b> Nilai Hyperparameter yang dihasilkan Aquila Optimizer.....	<b>37</b>
<b>Tabel 4.7</b> Rangkuman Hasil Performa Model XGB Aquila Optimizer .....	<b>38</b>
<b>Tabel 4.8</b> Analisis Statistik pada Dataset A .....	<b>40</b>
<b>Tabel 4.9</b> Analisis Statistik pada Dataset B.....	<b>41</b>
<b>Tabel 4.10</b> Analisis Statistik pada Dataset C .....	<b>41</b>
<b>Table 4.11</b> Tabel Benchmarking .....	<b>42</b>
<b>Tabel 4.12</b> Analisis Effect Size .....	<b>45</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Contoh Confusion Matrix.....	<b>10</b>
<b>Gambar 3.1</b> Flowchart Penelitian .....	<b>22</b>
<b>Gambar 4.1</b> Confusion Matrix pada Model XGB Vanilla.....	<b>33</b>
<b>Gambar 4.2</b> Confusion Matrix pada Model XGB Grid Search .....	<b>35</b>
<b>Gambar 4.3</b> Confusion Matrix pada Model XGB Random Search.....	<b>37</b>
<b>Gambar 4.4</b> Confusion Matrix pada Model XGB Aquila Optimizer .....	<b>38</b>
<b>Gambar 4.5</b> Konvergensi Model AO Dataset A.....	<b>39</b>
<b>Gambar 4.6</b> Konvergensi Model AO Dataset B .....	<b>39</b>
<b>Gambar 2.7</b> Konvergensi Model AO Dataset C .....	<b>40</b>
<b>Gambar 4.8</b> ROC-AUC Dataset A .....	<b>43</b>
<b>Gambar 4.9</b> ROC-AUC Dataset B.....	<b>44</b>
<b>Gambar 4.10</b> ROC-AUC Dataset C.....	<b>44</b>



## DAFTAR LAMPIRAN

1. Dokumentasi.....	56
2. Kode Program.....	56



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi informasi yang pesat mendorong meningkatnya kebutuhan terhadap sistem yang mampu menganalisis data secara otomatis dan menghasilkan keputusan yang efektif. *Machine Learning* (ML) adalah salah satu bidang yang berperan penting dalam pengolahan data dan pengambilan keputusan berbasis pola (Rifaldi et al., 2025). ML merupakan cabang dari *Artificial Intelligence* (AI) yang fokus pada pengembangan algoritma yang mempelajari pola dari data sebelumnya untuk memprediksi hasil baru dan mengambil keputusan berbasis data (Thalange, Patil, & Athavale, 2024).

Dalam penerapannya, salah satu teknik mendasar dalam ML adalah klasifikasi, yaitu proses untuk mengelompokkan data ke dalam label atau kelas tertentu berdasarkan fitur yang dimilikinya (Ramadhania & Satyahadewi, 2025). Tujuan klasifikasi adalah untuk membangun sebuah model yang mampu memetakan data input ke label yang sesuai. Saat ini, metode klasifikasi telah banyak diterapkan untuk berbagai bidang seperti deteksi penyakit, pengenalan wajah, klasifikasi teks, atau analisis sentimen (Huang et al., 2022).

Berbagai algoritma telah dikembangkan untuk menyelesaikan permasalahan klasifikasi, diantaranya *Decision Tree* (DT) yaitu algoritma dengan membentuk struktur pohon keputusan yang memisahkan data berdasarkan nilai dari fitur paling berpengaruh., *Support Vector Machine* (SVM) mencari *hyperplane* yang optimal

untuk memisahkan kelas dengan margin maksimum., *Naïve Bayes* mengasumsikan bahwa setiap fitur saling independen antar satu dengan lainnya. Dan *Gradient Boosting* (GB) pendekatan yang menggabungkan beberapa *weak learners* untuk membentuk *strong learners* dengan akurasi yang tinggi (Rabbani, Safitri, Try Puspa Siregar, Rahmaddeni, & Efrizoni, 2023).

GB membangun model secara bertahap dimana setiap model baru yang dibentuk untuk memperbaiki kesalahan dari model sebelumnya dengan meminimalkan *loss function* melalui pendekatan gradien (Rizkallah, 2025a). Algoritma GB memiliki beberapa kelebihan seperti, tingkat akurasi yang tinggi untuk berbagai jenis data, kemampuan menangani data numerik maupun kategorikal, serta fleksibel dalam mengontrol bias dan varian melalui pengaturan parameter.

Salah satu pengembangan dari algoritma GB adalah *Extreme Gradient Boosting* (XGBoost). XGBoost dirancang untuk meningkatkan komputasi dan mengurangi risiko *overfitting* (Rizkallah, 2025a). Algoritma ini banyak digunakan karena kemampuannya menghasilkan akurasi tinggi dan efisiensi waktu yang baik (Mehdary, Chehri, Jakimi, & Saadane, 2024). Beberapa penelitian sebelumnya telah membahas klasifikasi menggunakan algoritma XGBoost. Givari et al. (2022) membahas perbandingan performa tiga algoritma klasifikasi yaitu SVM, *Random Forest*, dan XGBoost dalam memprediksi kelayakan pengajuan kredit. Ketiga algoritma klasifikasi dilatih menggunakan metrik evaluasi yaitu *accuracy*, *precision*, dan *recall*. Hasil penelitian menunjukkan bahwa XGBoost memberikan performa paling unggul dengan *accuracy* 82%, *recall* 70%, dan *precision* 92%.

Di sisi lain, Gunawan et al. (2022) berfokus pada peningkatan kinerja klasifikasi serangan SYN salah satu jenis serangan DDoS yang sering di eksploitasi dalam jaringan menggunakan algoritma XGBoost. Hasil penelitian menunjukkan bahwa XGBoost memberikan performa klasifikasi dengan nilai akurasi keseluruhan sebesar 99%. Penerapan algoritma XGBoost juga digunakan dalam klasifikasi nasabah kartu kredit yang berpotensi mengalami kredit macet (Yulianti, Soesanto, & Sukmawaty, 2022). Hasil penelitian tersebut menunjukkan bahwa penggunaan XGBoost menghasilkan akurasi 80,2%.

Meskipun demikian, kinerja algoritma XGBoost sangat dipengaruhi oleh pemilihan *hyperparameter* (HP) yang tepat. Pemilihan HP yang tidak tepat dapat menyebabkan model mengalami *overfitting* atau *underfitting*, sehingga kinerja prediksi akan menurun. Oleh karena itu, diperlukan proses optimasi HP untuk memperoleh kombinasi terbaik (Mehdary et al., 2024). HP adalah nilai yang ditentukan sebelum proses pelatihan model dan tidak dipelajari secara langsung dari data (Arnold, Biedebach, Küpfer, & Neunhoffer, 2024). Beberapa HP yang penting dalam XGBost seperti *learning rate* (kecepatan pembaruan model pada setiap iterasi), *max depth* (kedalaman maksimum pohon keputusan) dan jumlah *estimators* yang digunakan.

Sehingga, terdapat beberapa metode yang umum digunakan dalam optimasi HP seperti *Grid Search* (GS) dan *Random Search* (RS). GS melakukan pencarian menyeluruh dengan menguji semua kombinasi parameter yang telah ditentukan. Metode ini sederhana dan menjamin menemukan kombinasi terbaik jika ruang pencarian kecil, namun kelemahan seperti waktu komputasi yang tinggi dan tidak efektif untuk ruang pencarian yang besar. Sedangkan RS, memilih kombinasi

parameter secara acak dalam ruang pencarian. Metode ini lebih cepat dan efektif dibandingkan dengan GS namun, bersifat acak sehingga memungkinkan melewati kombinasi optimal (Bischl, Sonabend, Kotthoff, & Lang, 2024). Meskipun metode GS dan RS banyak digunakan untuk optimasi HP, kedua metode ini memiliki keterbatasan pada efisiensi waktu dan kemampuan adaptasi terhadap ruang pencarian yang kompleks (Rizkallah, 2025b). Oleh karena itu, dibutuhkan metode pendekatan berbasis metaheuristik yang lebih efektif dalam optimasi HP pada XGBoost.

Salah satu algoritma metaheuristik yang terbaru adalah *Aquila Optimizer* (AO), yang diperkenalkan oleh (Abualigah et al., 2021). AO merupakan algoritma yang terinspirasi dari perilaku berburu burung elang (*Aquila*). AO memodelkan empat strategi utama yaitu *Exploration* (mencari area solusi yang luas), *Exploitation* (memperdalam pencarian di area terbaik), *Transition Phase* (menggabungkan eksplorasi dan eksploitasi secara dinamis), serta *Convergence Phase* (memastikan solusi bergerak menuju optimal) (Sasmal, Hussien, Das, & Dhal, 2023). Kelebihan AO terletak pada kemampuan konvergensi yang cepat, serta efektif untuk masalah optimasi non-linear (andre), sehingga diharapkan cocok digunakan untuk optimasi HP model ML seperti XGBoost. Penelitian terdahulu menunjukkan bahwa AO mampu memberikan hasil optimasi yang lebih stabil dan akurat (Mishra et al., 2023).

Berdasarkan penelitian terdahulu, optimasi HP pada XGBoost pada umumnya menggunakan metode GS dan RS. Namun, kedua metode ini memiliki keterbatasan dalam eksplorasi ruang pencarian yang kompleks. Sehingga, diperlukan pendekatan optimasi yang lebih efektif. Penelitian mengenai penerapan

algoritma metaheuristik seperti AO untuk optimasi HP XGBoost masih terbatas. Oleh karena itu, penelitian ini penting dilakukan untuk mengevaluasi efektivitas AO dalam meningkatkan kinerja model klasifikasi XGBoost melalui pencarian parameter untuk menemukan solusi yang lebih cepat dan hasil yang lebih stabil.

Dengan demikian, penelitian ini berfokus pada optimasi HP pada algoritma XGBoost menggunakan metode AO untuk klasifikasi produk. Pendekatan ini diharapkan dapat meningkatkan akurasi model klasifikasi produk dengan akurasi yang tinggi dan lebih efektif.

## 1.2. Rumusan Masalah

Algoritma XGBoost memiliki kemampuan tinggi dalam klasifikasi data, namun performanya sangat bergantung pada penentuan HP yang optimal. Pemilihan HP sering tidak efektif dan dapat menurunkan akurasi model. Untuk mengatasi hal tersebut, diperlukan sebuah metode optimasi yang efektif. AO digunakan sebagai solusi untuk mengoptimalkan HP XGBoost agar menghasilkan kinerja klasifikasi produk yang lebih akurat dan efektif. Sehingga penelitian ini merumuskan masalah yaitu, apakah penggunaan AO sebagai optimasi HP dapat meningkatkan performa XGBoost dalam mengklasifikasi.

## 1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. HP yang dioptimasi dalam penelitian ini yaitu *learning rate*, jumlah estimator, dan kedalaman pohon.
2. Penelitian ini menggunakan tiga jenis dataset berupa data sekunder yang diperoleh dari platform Kaggle dengan tautan:

[<https://www.kaggle.com/datasets/akhileshmuthusamy/product-classification>]; [<https://www.kaggle.com/datasets/mdshaukatali06/tech-gadgets-success-prediction-dataset>]; dan [<https://www.kaggle.com/datasets/mafzal19/electric-vehicle-sales-by-state-in-india>]. Dataset ini berisi data produk dengan karakteristik fitur yang berbeda-beda, dengan target laris dan tidak laris.

3. Metode evaluasi kinerja model menggunakan lima metrik yaitu *accuracy*, *precision*, *recall*, *F1-Score*, dan ROC-AUC.
4. Ruang pencarian (*search space*) untuk proses optimasi HP ditetapkan dalam rentang nilai tertentu yang telah ditentukan sebelumnya.
5. Proses *benchmarking* dilakukan dengan membandingkan berbagai model, yakni XGB Vanilla, XGB GS, XGB RS, dan XGB-AO.

#### 1.4. Tujuan Penelitian

1. Menganalisis kinerja algoritma XGBoost dalam klasifikasi produk untuk mengetahui tingkat akurasi model sebelum dilakukan proses optimasi.
2. Menerapkan metode AO untuk melakukan optimasi HP pada algoritma XGBoost, yaitu *learning rate*, *max depth*, dan jumlah *estimators*.
3. Meningkatkan performa algoritma XGBoost dalam klasifikasi produk melalui optimasi HP menggunakan AO.

#### 1.5. Manfaat Penelitian

1. Penelitian ini diharapkan mampu memberikan kontribusi dalam pengembangan ilmu pengetahuan di bidang ML, khususnya pada penerapan algoritma metaheuristik untuk optimasi HP pada model klasifikasi.
2. Penelitian ini menjadi dasar untuk merancang model baru yang lebih efektif dan dapat diterapkan pada berbagai kasus klasifikasi di masa mendatang, khususnya di bidang pemasaran dan bisnis.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Klasifikasi

Jenis pembelajaran pada ML dapat dibedakan berdasarkan ketersediaan label datanya, yaitu *Supervised Learning* (SL), *Unsupervised Learning* (UL), dan *Reinforcement Learning* (RL). Dalam SL, model belajar dari dataset yang sudah memiliki label untuk melakukan pembelajaran, sehingga mampu mengidentifikasi label dengan menggunakan fitur yang dimiliki untuk selanjutnya melakukan prediksi, sedangkan UL digunakan untuk menemukan pola dari data yang tidak memiliki label (Nurhalizah & Ardianto, 2024). Di sisi lain, RL berfokus pada pembelajaran melalui interaksi dengan lingkungan, di mana agen memperoleh *reward* atas tindakan yang diambil guna memaksimalkan hasil jangka panjang (Wijoyo et al., 2024).

Klasifikasi merupakan salah satu metode pada ML yang merupakan bagian dari pembelajaran terawasi atau *Supervised Learning* (SL) yang bertujuan untuk memprediksi suatu kelas data dengan kelas data lainnya atau untuk menemukan pola. Tujuan dari klasifikasi adalah untuk memprediksi data yang belum memiliki label atau kelas data tertentu (Andiny, 2024). Untuk mencari pola yang dapat menggambarkan setiap kelas data adalah proses dari klasifikasi. Mengklasifikasikan data yang berbeda di kelas yang berbeda juga merupakan tujuan dari klasifikasi untuk data baru yang akan digunakan. Dalam klasifikasi terdapat dua proses, yaitu proses pelatihan (*training*) dan proses (*testing*). Proses

training dilakukan menggunakan training set, yaitu bagian dari dataset yang telah diberi label dan digunakan untuk melatih model agar mampu mengenali pola dan hubungan antar fitur (Firdaus & Lelono, 2025; Firdausi dkk., 2024). Selanjutnya proses testing dilakukan dengan test set, yaitu data yang tidak pernah digunakan selama pelatihan, untuk menilai kemampuan model dalam melakukan prediksi terhadap data baru (Prasetyo & Laksana, 2022).

Melalui proses *training* dan *testing*, model klasifikasi dapat dievaluasi berdasarkan performa prediksinya terhadap data yang belum pernah dilihat sebelumnya. Algoritma umum yang digunakan untuk metode klasifikasi yang termasuk kedalam SL yaitu *Decision Tree*, *K-Nearest Neighbor*, *Random Forest*, SVM, dan GB.

## 2.2. Evaluasi Performa Model Klasifikasi

Evaluasi performa model klasifikasi merupakan salah satu tahap yang penting dalam proses pengembangan model ML. Tujuan dari evaluasi model yaitu untuk menilai sejauh mana model mampu mengklasifikasikan data secara akurat sesuai dengan kelas atau label yang ditentukan (Syahputra, Yani, Tanjung, Verina, & Ihsan, 2025). Proses evaluasi yang tepat diperlukan agar model tidak hanya memberikan hasil yang tinggi pada data pelatihan, tetapi mampu melakukan generalisasi terhadap data baru yang belum pernah dilihat sebelumnya. Tanpa adanya evaluasi yang tepat, model yang dibangun tidak dapat mengambil keputusan yang tepat dan akurat.

Evaluasi performa model klasifikasi dilakukan dengan membandingkan hasil prediksi model terhadap label dengan data sebenarnya. *Confusion matrix* merupakan salah satu metode yang paling sering digunakan untuk menggambarkan

hubungan antara hasil prediksi model dengan label dari data uji dengan empat komponen utamanya yaitu, *True Positif* (TP), *True Negatif* (TN), *False Positif* (FP), dan *False Negatif* (FN) (Tobing, Wijayakusuma, & Ida, 2025). TP (jumlah data positif yang diprediksi benar oleh model), TN (jumlah data negatif yang diprediksi benar oleh model), FP (jumlah data negatif yang keliru diprediksi sebagai data positif), FN (jumlah data positif yang keliru diprediksi sebagai data negatif) (Suryana, Warsito, & Suparti, 2021).



**Gambar 2.1** Contoh Confusion Matrix

Keempat komponen utama ini memberikan evaluasi yang menggambarkan kualitas model yaitu dengan metrik *accuracy*, *precision*, *recall*, dan *F1-Score*. *Accuracy* merupakan metrik untuk mengukur seberapa banyak prediksi model yang benar dibandingkan dengan total keseluruhan prediksi. Adapun rumusnya dapat dilihat pada Persamaan (2.1).

$$Accuracy = \left[ \frac{TP + TN}{TP + TN + FP + FN} \right] \quad (2.1)$$

Selanjutnya, metrik *Precision* adalah metrik yang digunakan untuk menunjukkan proporsi prediksi positif yang benar. Untuk menghitung nilai dari *Precision* digunakan Persamaan (2.2). Kemudian, *Recall* merupakan metrik yang digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data

positif yang sebenarnya. Persamaan (2.3) di bawah ini digunakan untuk menghitung nilai dari metrik *Recall*. Metrik terakhir yang digunakan adalah F1-Score yang merupakan nilai dari rata-rata dari precision dan recall dimana berguna saat data tidak seimbang. Rumus untuk menghitungnya dapat dilihat pada Persamaan (2.4).

$$Precision = \left[ \frac{TP}{TP + FP} \right] \quad (2.2)$$

$$Recall = \left[ \frac{TP}{TP + FN} \right] \quad (2.3)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.4)$$

Selain keempat metrik tersebut, performa klasifikasi dapat dinilai dari Receiver Operating Characteristic Area Under the Curve (ROC-AUC). ROC-AUC merupakan ukuran yang berada pada rentang 0,5 hingga 1,0 yang menggambarkan kemampuan model dalam membedakan antara kelas positif dan negatif. Semakin mendekati nilai 1,0, maka semakin baik kemampuan model dalam melakukan diskriminasi antar kelas. Secara umum, nilai ROC-AUC di atas 0,8 sudah dianggap cukup baik dan layak digunakan dalam evaluasi performa model (Çorbacıoğlu & Aksel, 2023).

### 2.3. Algoritma Gradient Boosting

*Gradient Boosting* (GB) adalah salah satu metode yang digunakan untuk meningkatkan akurasi model dalam klasifikasi (Handayani, Arum, Lestari, & Cahyana, 2025). Prinsip utama cara kerja GB adalah dengan menggabungkan beberapa *weak learners*, seperti *decision tree* secara bertahap menjadi *strong*

*learners* (Murdiansyah, 2024). Setiap model baru akan digunakan untuk memperbaiki kesalahan model sebelumnya, sehingga kesalahan secara bertahap akan diminimalkan (Turaina & Saputra, 2024). GB telah dibuktikan mampu mencapai akurasi yang lebih tinggi, ketahanan terhadap distribusi data yang kompleks dan kemampuan tambahan untuk secara bersamaan mengakomodasi fitur numerik dan kategorikal (Lubis, Nababan, Efendi, Sihombing, & Rahmat, 2025). Fungsi loss dimanfaatkan oleh GB untuk menentukan arah dari perbaikan model pada setiap iterasi, sehingga setiap proses pembelajaran terkontrol dan bertahap.

GB memiliki alur kerja yang sederhana yaitu, melakukan inisialisasi model awal misalnya untuk prediksi rata-rata dari data target. Kemudian menghitung kesalahan antara prediksi model sekarang dengan target sebenarnya, latih *weak learners* untuk memprediksi kesalahan yang dihitung pada tahap sebelumnya, kemudian gabungkan prediksi model baru dengan model sebelumnya menggunakan *learning rate*. Hal ini dilakukan berulang hingga jumlah iterasi tertentu atau jumlah kesalahan prediksi sudah minimal.

### 2.3.1. XGBoost

*Extreme Gradient Boosting* (XGBoost) adalah pengembangan algoritma GB yang dikembangkan oleh Chen dan Guestrin (2016) yang dirancang untuk meningkatkan efisiensi komputasi, kemampuan generalisasi, dan kecepatan proses pelatihan pada model GB konvensional. Algoritma XGBoost digunakan karena kemampuannya menghasilkan model dengan performa tinggi serta waktu eksekusi yang singkat (Turaina & Saputra, 2024).

Dalam XGBoost, terdapat suatu fungsi yang berperan untuk meminimalkan selisih antara hasil prediksi model dan nilai aktual sekaligus mengontrol kompleksitas model melalui regularisasi yang disebut sebagai fungsi objektif (Yulianti et al., 2022). Sehingga proses pembelajaran tidak hanya berfokus pada peningkatan akurasi tetapi juga menjaga agar model tidak mengalami *overfitting*. Fungsi objektif dapat dirumuskan sebagai berikut.

$$Obj(\theta) = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \quad (2.5)$$

Proses XGBoost dimulai dengan melakukan inisialisasi model menggunakan nilai prediksi rata-rata dari data target. Setelah itu, algoritma menghitung nilai gradien pertama (*first derivative*) ( $g_i$ ) dan gradien kedua (*second derivative*) ( $h_i$ ) untuk setiap sampel data guna menentukan arah dan besar perbaikan model. Perhitungan ini secara umum dapat dituliskan sebagai berikut.

$$g_i = \frac{dL(y_i, \hat{y}_i^{(t-1)})}{d\hat{y}_i^{(t-1)}}, \quad h_i = \frac{d^2L(y_i, \hat{y}_i^{(t-1)})}{d(\hat{y}_i^{(t-1)})^2} \quad (2.6)$$

Selanjutnya, pohon keputusan baru dibangun berdasarkan informasi gradien tersebut. Setiap percabangan pohon merepresentasikan keputusan yang bertujuan untuk meminimalkan kesalahan prediksi. Model akhir kemudian diperbarui dengan menambahkan hasil prediksi dari pohon baru ke model sebelumnya dengan dikalikan oleh learning rate ( $\eta$ ) yang berfungsi mengontrol kontribusi setiap iterasi. Persamaan pembaruan model dapat dituliskan sebagai berikut.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \quad (2.7)$$

Proses ini dilakukan secara berulang hingga jumlah iterasi yang ditentukan tercapai atau ketika peningkatan performa model sudah tidak signifikan.

## 2.4. Hyperparameter Gradient Boosting

Dalam algoritma GB pemilihan HP sangat penting untuk meningkatkan performa model yang optimal. HP merupakan parameter yang ditentukan sebelum proses pelatihan model. Pada GB, beberapa HP yang paling utama digunakan adalah *learning rate* ( $\eta$ ), *number of trees* ( $N_{tree}$ ), dan *max depth* ( $Max_d$ ) (Kavzoglu & Teke, 2022). Nilai  $\eta$  digunakan untuk melatih setiap pohon terhadap prediksi akhir. Nilai yang kecil membuat model belajar secara bertahap, cenderung lebih stabil dan dapat mengurangi risiko *overfitting*.

Sebaliknya, jika nilai  $\eta$  besar maka akan mempercepat proses pelatihan namun dapat membuat model mengalami *overfitting* dan tidak stabil.  $N_{tree}$  digunakan untuk menentukan seberapa banyak model *weak learners* yang akan digabungkan. Jumlah pohon yang banyak memiliki kemampuan untuk menangkap pola lebih kompleks namun akan terjadi *overfitting* jika jumlah pohon terlalu besar. Dengan  $Max_d$  maka jumlah fitur yang dipilih secara acak dapat ditentukan. Dengan adanya pemilihan fitur secara acak maka dapat membantu mengurangi *overfitting*. Untuk memberikan gambaran yang lebih jelas, berikut pseudocode XGBoost yang menjelaskan seluruh proses pembelajarannya.

---

### Algoritma XGBoost

---

**Input:**Data pelatihan  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ Jumlah iterasi  $T$ Learning rate  $\eta$ Number of trees  $N_{tree}$ max depth  $Max_d$ **Ouput:**Model akhir  $F(X)$ 

Langkah-langkah :

1. inisialisasi model awal :

 $F_0(x) = \text{nilai rata-rata target (missal mean (y))}$ 2. untuk setiap iterasi  $t = 1$  hingga  $T$  lakukan :a. Hitung gradien pertama ( $g_i$ ) dan gradien kedua ( $h_i$ ) untuk setiap sampelb. Bangun pohon keputusan  $f_t(x)$  berdasarkan  $g_i$  dan ( $h_i$ )

c. Tentukan bobot setiap daun pada pohon menggunakan hasil optimasi internal

d. Perbarui model :

$$f_t(x) = F_{\{t-1\}}(x) + \eta * f_t(x)$$

e. Evaluasi fungsi objektif untuk menentukan konvergensi

3. Kembalikan model akhir  $f_t(x)$ 


---

## 2.5. Optimasi Hyperparameter

Optimasi HP merupakan suatu usaha untuk menemukan HP yang memaksimalkan tingkat akurasi, dan kemampuan generalisasi model. Optimasi HP yang tidak tepat dapat menyebabkan *overfitting* atau *underfitting* (Nasional, Informasi, Teks, Marthen, & Novita, 2024), sehingga model tidak mampu melakukan prediksi secara akurat pada data baru. Beberapa metode optimasi HP diantaranya adalah *Grid Search* (GS) dan *Random Search* (RS). GS adalah metode optimasi HP yang umum digunakan, yaitu untuk melakukan pencarian secara menyeluruh terhadap semua nilai HP yang telah ditentukan dalam satu grid. Setiap kombinasi diuji menggunakan Teknik *cross validation* untuk menilai performa model. GS memiliki kemampuan untuk menentukan kombinasi optimal dalam grid

yang ditentukan. Namun metode ini membutuhkan waktu komputasi yang tinggi (Subaşi, 2024).

Algoritma GS bekerja dengan melakukan beberapa langkah yaitu, menentukan himpunan nilai HP yang akan di uji, misalnya nilai  $\eta$ ,  $Max_d$ , dan  $N_{tree}$  pada algoritma GB atau XGBoost. Sistem akan melatih model dengan menggunakan teknik *k-fold cross validation*, dimana dataset dibagi menjadi beberapa subset untuk memastikan hasil evaluasi lebih stabil (Pradana & Atina, 2024). Setelah parameter diperoleh maka model akhir akan dilatih kembali menggunakan keseluruhan data pelatihan dengan konfigurasi terpilih untuk menghasilkan model final yang siap digunakan dalam proses prediksi.

Di sisi lain, metode RS melakukan kombinasi HP secara acak dalam ruang pencarian yang telah ditentukan (Simanjuntak et al., 2024). RS melakukan pendekatan lebih efisien dalam pencarian HP. Metode ini cukup efektif ketika beberapa HP hanya memiliki pengaruh kecil terhadap performa model. Kemampuan RS untuk mengeksplorasi ruang HP yang lebih luas dalam waktu singkat, menjadikan metode ini cocok digunakan pada model dengan dataset yang besar.

## 2.6. Aquila Optimizer

AO merupakan algoritma metaheuristik yang terinspirasi dari cara atau strategi burung elang dalam berburu menangkap mangsa. Prinsip kerja AO yaitu dengan eksplorasi dan eksploitasi yang seimbang. Dimana eksplorasi bertujuan untuk mencari area solusi dalam ruang pencarian sedangkan eksploitasi berfokus pada penyempurnaan solusi yang ditemukan. Proses ekplorasi dan eskploitasi ini

menyerupai pola berburu Aquila yaitu melakukan pengamatan jarak jauh, terbang cepat untuk mengejar target, hingga melakukan serangan presisi pada mangsa.

AO mengadaptasi kecepatan solusi dalam ruang pencarian. Setiap solusi diperbarui berdasarkan iterasi yang memperhitungkan posisi sebelumnya, dan pola acak yang meniru perilaku burung elang. Berikut penjabaran tahapan-tahapan pada algoritma AO:

1. *Exploration Phase* (Terbang tinggi dan pengamatan luas)

Pada tahap ini, burung elang melakukan pengamatan dari ketinggian untuk menjelajahi area pencarian secara luas dan mengidentifikasi posisi potensial mangsa. Tahap ini berfungsi untuk memperluas pencarian solusi secara global agar algoritma tidak terjebak pada nilai lokal optimum sejak awal proses optimasi. Proses pembaruan posisi solusi pada fase ini dilakukan berdasarkan Persamaan (2.8), yang memanfaatkan posisi terbaik saat ini, rata-rata populasi, dan nilai acak untuk menjaga keragaman solusi dalam ruang pencarian.

$$X_{t+1} = X_{best}(t) \times 1 - \frac{t}{T} + (X_{mean}(t)) - (X_{rand}(t)) \times r_1 \quad (2.8)$$

Keterangan :

$X_{best}(t)$  : posisi terbaik saat ini

$X_{mean}(t)$  : rata-rata posisi populasi

$X_{rand}(t)$  : posisi acak dalam populasi

$r_1$  : bilangan acak [0.1]

2. *Exploration – Transition Phase* (Terbang cepat untuk mendekati target)

Pada tahap ini, burung elang mulai mempersempit area pencarian dengan terbang cepat menuju arah target yang dianggap paling potensial. Fase ini berfungsi sebagai peralihan dari eksplorasi global menuju eksploitasi yang lebih fokus di sekitar solusi terbaik yang telah ditemukan. Pembaruan posisi pada tahap ini mengikuti Persamaan (2.9), yang memanfaatkan posisi terbaik dan nilai acak sebagai pengendali arah pergerakan agar pencarian tetap adaptif dan terarah menuju solusi optimal.

$$X_{t+1} = X_{best}(t) + A + (X_{best}(t) - X(t)) \times r_2 \quad (2.9)$$

3. *Exploitation Phase* (Menyerang dari jarak dekat)

Pada tahap selanjutnya, burung elang mulai menurunkan ketinggian terbangnya untuk menyerang dari jarak dekat dan fokus memperbaiki solusi di sekitar area terbaik yang telah ditemukan. Fase ini bertujuan untuk melakukan eksploitasi lokal dengan memusatkan pencarian di sekitar posisi optimal sementara. Pembaruan posisi pada tahap ini mengikuti Persamaan (2.10), yang memanfaatkan selisih antara posisi terbaik dan rata-rata populasi untuk meningkatkan akurasi pencarian terhadap solusi optimal.

$$X_{t+1} = X_{best}(t) + \delta + (X_{best}(t) - X_{mean}(t)) \times r_2 \quad (2.10)$$

Dengan nilai  $\delta$  yang merupakan parameter kontrol untuk menyesuaikan tingkat eksploitasi

4. *Final Attack Phase* (Serangan presisi terhadap mangsa)

Di tahap terakhir, burung elang melakukan serangan cepat dan presisi terhadap mangsa sebagai langkah akhir untuk memastikan konvergensi solusi

menuju nilai optimum global. Fase ini berfungsi untuk memperkuat eksploitasi akhir sekaligus menjaga keberagaman solusi agar tidak terjebak pada nilai lokal optimum. Pembaruan posisi pada tahap ini dihitung menggunakan Persamaan (2.11), yang memanfaatkan distribusi Lévy flight ( $Levy(d)$ ) untuk menghasilkan gerakan acak adaptif, sehingga memperluas jangkauan pencarian dan meningkatkan peluang menemukan solusi terbaik secara global.

$$X_{t+1} = X_{best}(t) + \alpha + Levy(d) \times (X_{best}(t) - X(t)) \quad (2.11)$$

Untuk memberikan gambaran yang lebih jelas, berikut pseudocode AO yang menjelaskan seluruh proses pembelajarannya.

---

#### **Aquila Optimizer (AO)**

---

##### **Input:**

Fungsi tujuan  $f(x)$   
 Jumlah populasi N  
 Dimensi masalah D  
 Jumlah iterasi maksimum T

##### **Output:**

Solusi terbaik  $X_{best}$

Langkah-langkah:

1. Inisialisasi populasi awal  $x(i)$ ,  $I = 1, \dots, N$  secara acak dalam batas pencarian .
  2. Evaluasi nilai fitness untuk setiap individu.
  3. Tentukan posisi terbaik  $X_{best}$
  4. Untuk setiap iterasi  $t = 1$  hingga T lakukan:
    - a. Perbarui posisi rata-rata populasi  $X_{mean}$ .
    - b. Pilih strategi pembaruan posisi berdasarkan fase:
      - jika  $t < 0.25T$ : gunakan strategi eksplorasi (terbang tinggi)
      - jika  $0.25T \leq t < 0.5T$ : gunakan strategi transisi cepat
      - jika  $0.5T \leq t < 0.75T$ : gunakan eksploitasi local
      - jika  $t \geq 0.75T$ : lakukan serangan presisi (final attack)
    - c. Evaluasi Kembali fitness setiap individu
    - d. Perbarui  $X_{best}$  jika solusi baru lebih baik
  5. Kembalikan  $X_{best}$  sebagai solusi optimal
-

## 2.7. Penelitian Terdahulu

Terdapat beberapa penelitian terdahulu yang berfokus pada proses optimasi HP pada algoritma XGBoost yang menggunakan berbagai metode optimasi. Beragam pendekatan telah dikembangkan untuk meningkatkan performa model XGBoost, seperti GS dan RS dengan hasil yang beragam pula. Tabel 2.1 menampilkan beberapa penelitian terdahulu yang menjadi rujukan pada penelitian ini.



**Tabel 2.1** Penelitian Terdahulu

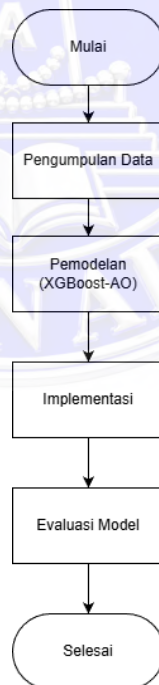
Author	Judul	Jenis Data	Algoritma optimasi	HP	Hasil
Yulianti dkk. (2022)	Penerapan Metode <i>Extreme Gradient Boosting</i> (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit	Kartu kredit	Grid Search	<i>n_estimators</i> , <i>max_depth</i> , <i>min_child_weight</i> , <i>eta</i> ( <i>learning_rate</i> ), <i>gamma</i> , <i>subsample</i> , <i>colsample_bylevel</i> .	Nilai akurasi meningkat dari 80% menjadi 83,42%
(Anggoro, 2021)	Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure	Prediksi gagal ginjal kronis	Grid Search dan Random Search	<i>Learning rate</i> , <i>gamma</i> , <i>max depth</i> , <i>subsample</i> , <i>colsample bytree</i> .	Nilai akurasi meningkat hingga 99.33 %
(Abdurrahman, Oktavianto, & Sintawati, 2022)	Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridsearch dan Random Search Pada Klasifikasi Penyakit Diabetes	Penyakit diabetes	Grid Search dan Random Search	<i>Learning rate</i> , <i>gamma</i> , <i>max depth</i> , <i>subsample</i> , <i>colsample bytree</i> .	Nilai akurasi meningkat hingga 95%
(Wijaya, Hostiadi, & Triandini, 2024)	Optimization of XGBOOST Algorithm Using Parameter Tuning In Retail Sales Prediction	Prediksi penjualan ritel	Grid Search	<i>Learning rate</i> , <i>Number of Trees</i> , <i>n_estimators</i> , <i>max depth</i> , <i>subsample</i> , <i>colsample_bytree</i>	Nilai akurasi meningkat hingga 98,41%
(Kavzoglu & Teke, 2022)	Advanced Hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost)	Prediksi spasial longsor tanah dangkal	Random Search	<i>Learning rate</i> , <i>max depth</i> , <i>subsample</i> , <i>n_estimators</i>	Nilai akurasi meningkat hingga 89%

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang disusun secara sistematis agar proses analisis berjalan secara terarah. Tahapan penelitian meliputi pengumpulan data, pemodelan dengan algoritma XGBoost yang dioptimasi menggunakan AO, implementasi model, evaluasi performa, proses skenario, serta pembahasan hasil penelitian. Alur penelitian ini digambarkan pada Gambar 3.1 di bawah ini yang menunjukkan urutan proses dari tahap awal hingga tahap akhir.



**Gambar 3.1** Flowchart Penelitian

Proses penelitian dimulai dengan pengumpulan dataset produk, dilanjutkan dengan pra-pemrosesan data untuk memastikan kualitas data, kemudian dilakukan pemodelan menggunakan algoritma XGBoost yang dioptimasi dengan AO. Selanjutnya, model diimplementasikan menggunakan bahasa pemrograman python dan library pendukung, lalu dilakukan evaluasi performa model berdasarkan metrik tertentu. Hasil dari evaluasi tersebut kemudian dianalisis dan dibahas untuk melihat pengaruh optimasi terhadap peningkatan performa model.

### 3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari sumber Kaggle, yang berisi data spesifikasi dan karakteristik produk. Data penelitian berjumlah sebanyak 3 jenis dataset yang menggambarkan berbagai jenis produk penjualan, uraian dari ketiga data tersebut dijelaskan pada sub-bab sub-bab berikut. Sebagai informasi tambahan, Tabel 3.1 di bawah ini menyajikan informasi mengenai data dalam penelitian yang terdiri dari jumlah datanya, fitur yang digunakan, serta jumlah masing-masing target.

**Tabel 3.1** Informasi Data Penelitian

Dataset	Jumlah Data	Jumlah Fitur	Jumlah Target	
			0	1
Data A	61.249	26	48.550	12.699
Data B	100	15	48	52
Data C	96.845	7	82.783	14.062

#### 3.2.1. Dataset A

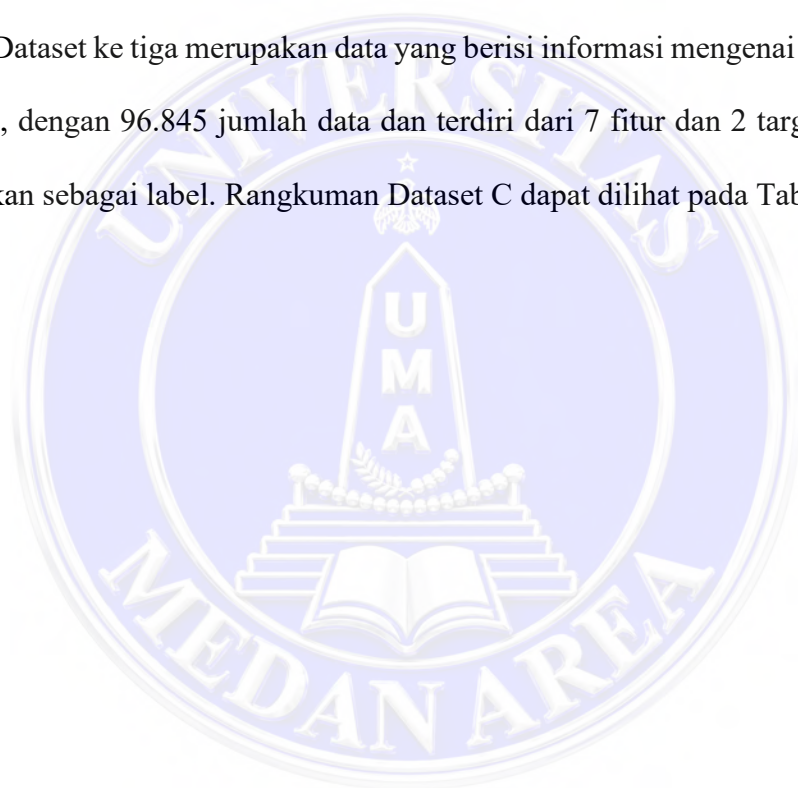
Dataset pertama pada penelitian ini bersumber dari Kaggle yang berisi informasi terkait produk baju. Data terdiri dari 61.249 jumlah data dengan 24 fitur dan jumlah target sebanyak 2 yang menjadi label yang akan di prediksi. Rangkuman Dataset A dapat dilihat pada Tabel 3.2.

### 3.2.2. Dataset B

Dataset kedua merupakan data yang berisi informasi mengenai produk gadget yang bersumber dari Kaggle, data ini memiliki 100 jumlah data dengan 15 fitur dan 2 target sebagai label dalam penelitian ini. Rangkuman Dataset B dapat dilihat pada Tabel 3.3.

### 3.2.3. Dataset C

Dataset ke tiga merupakan data yang berisi informasi mengenai produk mobil listrik , dengan 96.845 jumlah data dan terdiri dari 7 fitur dan 2 target yang yang dijadikan sebagai label. Rangkuman Dataset C dapat dilihat pada Tabel 3.4.



**Tabel 3.2** Dataset A

	Feature 1	Feature 2	Feature 3	Feature 4	...	Feature 21	Feature 22	Feature 23	Feature 24
Count	61249.0	61249.0	61249.0	61249.0	...	61249.0	61249.0	61249.0	61249.0
Mean	23.999	0.5713	94.031	472.88	...	2228.61	97.751	20.864	12.518
Std	14.780	0.3107	14.251	26.436	...	145.472	5.369	9.936	5.962
Min	0.0	0.0	60.0	445.0	...	1915.0	84.930	10.160	6.0843
Max	42.008	0.8420	100.0	518.67	...	2388.0	100.0	39.890	23.885

**Tabel 3.3** Dataset B

	Price	Battery Life	Screen Size	Storage Capacity	RAM	Camera Quality	Weight	User Ratings	Sales Volume	Success
Count	100	100	100	100	100	100	100	100	100	100
Mean	1072.5800	28.6900	9.2220	141.3600	11.4600	57.7600	1474.2200	2.9610	485333.970	0.52000
Std	532.161652	13.500239	4.561977	166.863128	10.766972	31.967825	798.391319	1.184487	29167.309264	0.502117
Min	113.000000	5.000000	1.000000	8.000000	1.000000	0.000000	88.000000	1.100000	1055.000000	0.000000
max	1999.0	49.0	17.0	512.0	32.0	104.0	298.0	5.0	98239.0	1.0

**Tabel 3.4** Dataset C

	Year	Date	EV_Sales_Quantity	Target
Count	96845	96845	96845	96845
Mean	2018.62	2018-08-21	37.1	0.14
Std	2.895	NaN	431.5	0.35
Min	2014.0	2014-01-01	0.0	0.0
max	2024.0	202-01-01	20584.0	1

### 3.3. Pemodelan (XGBoost-Aquila Optimizer)

Integrasi antara AO dan XGBoost dilakukan dengan cara menghubungkan hasil pencarian HP dari AO ke dalam fungsi pelatihan XGBoost, untuk memperoleh kombinasi HP yang menghasilkan nilai akurasi tinggi dan tingkat kesalahan terendah. Setiap individu pada populasi AO merepresentasikan satu kandidat solusi berupa kombinasi nilai HP XGBoost. Proses optimasi dilakukan dengan prinsip eksplorasi dan eksploitasi AO untuk menjelajahi ruang pencarian parameter XGBoost. Nilai *fitness* dari setiap individu dihitung menggunakan hasil evaluasi performa XGBoost pada data pelatihan. AO kemudian memperbarui posisi individu berdasarkan empat strategi perilaku burung elang untuk menemukan kombinasi parameter yang memberikan performa model terbaik. Fungsi objektif dirumuskan sebagai berikut :

$$\min F(X) = 1 - Accuracy(X) \quad (3.1)$$

Keterangan :

$F(X)$  : Fungsi objektif yang akan diminimalkan

$Accuracy(X)$  : Akurasi model XGBoost yang dihasilkan dari kombinasi HP X

Optimasi bertujuan untuk meminimalkan kesalahan prediksi model dengan cara memaksimalkan nilai akurasi. Setelah proses iterasi selesai, nilai HP terbaik yang diperoleh digunakan untuk melatih Kembali model XGBoost sehingga dihasilkan model akhir yang lebih akurat.

---

### XGBoost-AO

---

**Input:**

Data latih  $x(train), y(train)$

Fungsi evaluasi (akurasi)

Jumlah populasi N

Jumlah iterasi maksimum T

Batas nilai HP XGBoost

**Output:**

Kombinasi HP terbaik  $X_{best}$

Model XGBoost terlatih dengan parameter optimal

Langkah-langkah:

1. Inisialisasi populasi awal:  
Setiap individu merepresentasikan kombinasi HP XGBoost ( $\eta, max_{depth}, N_{tree}$ )
  2. Evaluasi fitness setiap individu:
    - a. Gunakan nilai HP individu ke-I untuk melatih XGBoost
    - b. Hitung nilai akurasi model menggunakan cross-validation
    - c. Simpan nilai fitness = 1 -akurasi
  3. Tentukan individu dengan fitness terbaik sebagai  $X_{best}$
  4. Untuk setiap iterasi  $t = 1$  hingga T lakukan:
    - a. Perbarui posisi populasi menggunakan mekanisme AO:
      - Fase eksplorasi (mencari solusi baru)
      - Fase transisi (mendekati solusi terbaik)
      - Fase eksploitasi (memperbaiki solusi sekitar  $X_{best}$ )
      - Fase serangan akhir (konvergensi global)
    - b. Evaluasi ulang setiap individu dengan melatih XGBoost
    - c. Perbarui  $X_{best}$  jika ditemukan solusi yang lebih baik
  5. Setelah iterasi selesai:  
Gunakan  $X_{best}$  untuk melatih ulang model XGBoost pada seluruh data latih
  6. Keluaran:  
Model XGBoost dengan HP terbaik dan nilai akurasi tertinggi
- 

### 3.4. Implementasi

Implementasi merupakan proses penerapan model hasil pemodelan ke dalam sistem komputasi nyata. AO memerlukan beberapa parameter inisialisasi utama yang mengatur perilaku pencarian dan konvergensi algoritma seperti jumlah populasi N, jumlah iterasi maksimum T, dimensi masalah  $D$ , *upper* dan *lower bound*, parameter  $\alpha$ , *levy* ( $d$ ), dan fungsi objektif. N menentukan jumlah individu atau solusi yang digunakan dalam proses pencarian. Semakin besar nilai populasi, semakin luas cakupan eksplorasi, tetapi waktu komputasi juga meningkat. T menentukan jumlah siklus pembaruan posisi yang dilakukan algoritma hingga

konvergen. Nilai  $T$  berpengaruh langsung terhadap stabilitas hasil.  $D$  menggambarkan jumlah parameter yang akan dioptimasi, dalam XGBoost-AO jumlah HP yang dioptimasi yaitu  $\eta$ ,  $Max_d$ , dan  $N_{tree}$ . *Upper* dan *lower bound* menentukan rentang nilai minimum dan maksimum untuk setiap HP.  $\alpha$  koefisien pengendali laju pergerakan individu pada fase *final attack*.  $(d)$  digunakan untuk menjaga diversitas populasi agar tidak cepat terjebak pada solusi local. Parameter ini tidak bernilai tunggal, tetapi bergantung pada fungsi distribusi acak *levy* yang diterapkan. Fungsi objektif merupakan fungsi evaluasi yang digunakan untuk menilai kualitas setiap solusi.

**Tabel 3.5** Rentang Nilai Hyperparameter

Hyperparameter	Simbol	Rentang Nilai
Learning rate	$\eta$	0.1-0.3
N-estimators	$N_{tree}$	50 – 500
Max_depth	$Max_d$	3-5

### 3.5. Parameter Aquila Optimizer

AO bekerja berdasarkan mekanisme eskplorasi dan eksploitasi yang meniru strategi berburu elang aquila dalam mencari mangsa. untuk proses optimasi yang efektif diperlukan penetapan parameter AO, yaitu jumlah populasi (N) dan jumlah iterasi (T).

Jumlah populasi merupakan banyak agen atau solusi yang digunakan AO pada setiap siklus optimasi. Pada penelitian ini, jumlah populasi ditetapkan sebesar  $N = 10$  yang mempertimbangkan keseimbangan antara eksplorasi dalam ruang pencarian HP dan waktu komputasi yang terkendali. Jumlah populasi yang terlalu kecil dapat mengurangi keragaman solusi sehingga meningkatkan risiko terjebak pada optimum lokal, sedangkan jumlah populasi yang terlalu besar dapat memperpanjang waktu komputasi tanpa memberikan peningkatan performa yang signifikan.

Jumlah iterasi menunjukkan total siklus pembaruan posisi dalam proses pencarian solusi. Penelitian ini menggunakan  $T = 20$  iterasi. Nilai iterasi ini dipilih agar AO dapat menemukan kombinasi HP terbaik. Jumlah iterasi yang terlalu rendah berpotensi menyebabkan proses optimasi berhenti sebelum menemukan solusi yang optimal, sedangkan jumlah iterasi yang terlalu tinggi dapat meningkatkan waktu komputasi yang berlebihan tanpa peningkatan performa. Dengan jumlah  $N = 10$  dan  $T = 20$ , AO diharapkan mampu memberikan keseimbangan yang optimal untuk efisiensi waktu komputasi.

### 3.6. Evaluasi

Evaluasi dilakukan untuk mengukur tingkat keberhasilan model dalam melakukan klasifikasi produk. Evaluasi model menggunakan beberapa metrik yaitu *accuracy*, *precision*, *recall*, F1-score. *Accuracy* digunakan untuk mengukur presentasi prediksi yang benar terhadap seluruh data uji, sedangkan *precision* dan *recall* digunakan untuk menilai ketepatan dan sensitivitas model terhadap data positif. F1- score dihitung sebagai rata-rata harmonis antara *precision* dan *recall* untuk memberikan gambaran mengenai keseimbangan performa model.

Selain itu, digunakan juga *confusion matrix* untuk menilai distribusi hasil prediksi pada setiap kelas. Proses evaluasi dilakukan pada empat model, yaitu model XGBoost tanpa optimasi, model XGBoost dengan *Grid Search*, model XGBoost dengan *Random Search*, dan model XGBoost yang telah dioptimasi menggunakan AO.

### 3.7. Skenario Penelitian

Dalam penelitian ini, terdapat beberapa skenario penelitian yang terbagi berdasarkan dataset penelitian (Dataset A, B, C). Untuk Skenario 1, digunakan XGB Vanilla dengan Dataset A, dilanjutkan dengan Skenario 2 dengan Dataset B, dan Skenario 3 dengan Dataset C. Kemudian seterusnya untuk semua model pengujian, yakni XGB GS, XGB RS, dan XGB AO. Sehingga, terdapat 12 skenario yang akan diuji dalam penelitian ini. Tabel 3.6 di bawah ini menampilkan rangkuman dari keseluruhan skenario.

**Tabel 3.6** Rangkuman Skenario Penelitian

Skenario	Dataset	Model
Skenario 1	A	XGB Vanilla
Skenario 2	B	XGB Vanilla
Skenario 3	C	XGB Vanilla
Skenario 4	A	XGB GS
Skenario 5	B	XGB GS
Skenario 6	C	XGB GS
Skenario 7	A	XGB RS
Skenario 8	B	XGB RS
Skenario 9	C	XGB RS
Skenario 10	A	XGB AO
Skenario 11	B	XGB AO
Skenario 12	C	XGB AO



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian ini dilakukan dengan tujuan untuk menganalisis performa algoritma XGBoost dalam proses klasifikasi serta meningkatkan kinerja melalui optimasi HP menggunakan metode Aquila Optimizer (AO). Berdasarkan seluruh rangkaian pengujian dan analisis yang telah dilakukan, dapat disimpulkan model XGB dengan optimasi Aquila Optimizer (XGB AO) terbukti menjadi model dengan performa terbaik pada dua dari tiga dataset yang digunakan, yaitu dataset A dan dataset C menunjukkan bahwa mekanisme pencarian solusi dari AO mampu menemukan kombinasi HP yang lebih efektif dan stabil.

Perbandingan 2 dataset unggul dan 1 dataset tidak unggul dipengaruhi oleh ketidakseimbangan jumlah data pada dataset B, jumlah data yang kecil menyebabkan model sulit mempelajari karakteristik kelas secara optimal. Meskipun demikian, analisis *effect size* menunjukkan besarnya pengaruh penggunaan AO terhadap peningkatan performa model. Hasil dari kedua ukuran yaitu Cohen's  $d$  dan Cliff's Delta menunjukkan bahwa AO benar-benar memberikan kontribusi nyata dalam meningkatkan performa model. Penelitian ini menunjukkan bahwa penggunaan AO mampu meningkatkan performa XGBoost secara signifikan, serta memberikan hasil yang lebih unggul daripada metode optimasi konvensional. Dengan demikian, XGB-AO direkomendasikan sebagai pendekatan optimasi HP yang efektif untuk permasalahan klasifikasi dengan karakteristik data yang beragam.

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan terdapat beberapa limitasi yang dijadikan sebagai saran untuk ke depannya, yakni sebagai berikut.

1. Penelitian ini hanya menggunakan metode Aquila Optimizer sebagai pendekatan optimasi HP pada algoritma XGBoost, sedangkan XGBoost memiliki HP sangat luas. Oleh karena itu, penulis menyarankan penelitian selanjutnya untuk membandingkan metode optimasi lainnya seperti Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), atau algoritma metaheuristik lain untuk memperoleh hasil yang lebih komprehensif.
2. Selain itu, penelitian ini tidak membahas aspek waktu komputasi dan beban komputasi. Maka dari itu, penelitian selanjutnya diharapkan dapat melakukan evaluasi terhadap efisiensi komputasi. Dengan demikian, penelitian lanjutan dapat memberikan gambaran mengenai kinerja dan efisiensi metode optimasi pada XGBoost dalam klasifikasi produk laris.
3. Penelitian ini hanya mengoptimasi 3 HP pada XGBoost, yakni *learning rate*, *max\_depth*, dan *n\_estimator*. Sehingga, penelitian berikutnya dapat melakukan pencarian nilai HP lainnya seperti *subsample*, *coll\_sample* dan lainnya.

## DAFTAR PUSTAKA

- Abdurrahman, G., Oktavianto, H., & Sintawati, M. (2022). Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridsearch dan Random Search Pada Klasifikasi Penyakit Diabetes, *7*(3), 193–198.
- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A. A., & Gandomi, A. H. (2021). Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Computers and Industrial Engineering*, *157*, 107250. <https://doi.org/10.1016/j.cie.2021.107250>
- Andiny, R. (2024). *Klasifikasi Karakteristik Turnover Karyawan Menggunakan Algoritma C4.5*.
- Anggoro, D. A. (2021). Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure, *14*(6), 198–207. <https://doi.org/10.22266/ijies2021.1231.19>
- Arnold, C., Biedebach, L., K pfer, A., & Neunhoeffler, M. (2024). The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, *12*(4), 841–848. <https://doi.org/10.1017/psrm.2023.61>
- Bischi, B., Sonabend, R., Kothhoff, L., & Lang, M. (2024). Applied Machine Learning Using mlr3 in R. *Applied Machine Learning Using Mlr3 in R*, 1–339. <https://doi.org/10.1201/9781003402848>
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

- Çorbacioğlu, Ş. K., & Aksel, G. (2023). Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value. *Turkish Journal of Emergency Medicine*, 23(4), 195–198.
- Firdaus, A. Z., & Lelono, D. (2025). Sistem Klasifikasi Sampah Otomatis Berbasis Deteksi Objek Real-Time Pada Single Board Computer Dengan Algoritma YOLO, 15(1), 49–60. <https://doi.org/10.22146/ijeis.104520>
- Firdausi, A. T., Arhandi, P. P., Pribadi, F. A., Damayanti, R., & Aqil, A. (2024). Pengembangan Modul Pembelajaran ERD Interaktif pada SQLearn. *Jurnal Informatika Polinema*, 10(4), 471–478.
- Givari, M. R., Sulaeman, M. R., & Umaidah, Y. (2022). Perbandingan Algoritma SVM , Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit, 16, 141–149.
- Gunawan, R., Handika, E. S., & Ismanto, E. (2022). Jurnal Computer Science and Information Technology ( CoSciTech ), 3(3), 453–463.
- Handayani, H. H., Arum, S., Lestari, P., & Cahyana, Y. (2025). Evaluasi Kinerja Algoritma Random Forest Dan Gradient Boosting Untuk Klasifikasi Penyakit Jantung, 9(1), 112–124.
- Huang, J., Li, J., Li, Z., Zhu, Z., Shen, C., Qi, G., & Yu, G. (2022). Detection of Diseases Using Machine Learning Image Recognition Technology in Artificial Intelligence. *Computational Intelligence and Neuroscience*, 2022, 5658641. <https://doi.org/10.1155/2022/5658641>
- Kavzoglu, T., & Teke, A. (2022). Advanced hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost). *Bulletin of Engineering Geology and the Environment*,

81(5), 201.

Lubis, A. H., Nababan, E. B., Efendi, S., Sihombing, P., & Rahmat, R. F. (2025). Enhancing Gradient Boosting Performance with Puma Optimizer for Predicting Customer Churn. *2025 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 445–450.

<https://doi.org/10.1109/ICIMCIS68501.2025.11327392>

Mehdary, A., Chehri, A., Jakimi, A., & Saadane, R. (2024). Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection. *Sensors*, 24(4).

<https://doi.org/10.3390/s24041230>

Mishra, S. K., Jha, A. V., Appasani, B., Bizon, N., Thounthong, P., & Mungporn, P. (2023). Ocean Wave Energy Control Using Aquila Optimization Technique. *Energies*, 16(11). <https://doi.org/10.3390/en16114495>

Murdiansyah, D. T. (2024). Prediksi Stroke Menggunakan Extreme Gradient Boosting Stroke Prediction Using Extreme Gradient Boosting, (2), 419–426.

<https://doi.org/10.26798/jiko.v8i2.1295>

Nasional, J., Informasi, S., Teks, E., Marthen, E., & Novita, I. (2024). Optimasi RoBERTa dengan Hyperparameter Tuning untuk Deteksi, 03, 240–248.

Nurhalizah, R. S., & Ardianto, R. (2024). Analisis Supervised dan Unsupervised Learning pada Machine Learning : Systematic Literature Review, 4(1), 61–72.

Pradana, A. I., & Atina, V. (2024). Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa, 239–248.

<https://doi.org/10.33364/algorithm/v.21-1.1618>

- Prasetyo, A. B., & Laksana, T. G. (2022). Optimasi Algoritma K-Nearest Neighbors dengan Teknik Cross Validation Dengan Streamlit ( Studi Data : Penyakit Diabetes ), *6*(2), 194–204.
- Rabbani, S., Safitri, D., Try Puspa Siregar, F., Rahmaddeni, R., & Efrizoni, L. (2023). Evaluation of Support Vector Machine, Naive Bayes, Decision Tree, and Gradient Boosting Algorithms for Sentiment Analysis on ChatGPT Twitter Dataset. *Indonesian Journal of Artificial Intelligence and Data Mining*, *7*(1), 11. <https://doi.org/10.24014/ijaidm.v7i1.24662>
- Ramadhania, W., & Satyahadewi, N. (2025). OPTIMALISASI MODEL GRADIENT BOOSTING MACHINE DENGAN GRID SEARCH UNTUK MENENTUKAN, *14*(1), 113–122.
- Rifaldi, D., Famuji, T. S., Wijaya, S. A., Abougarair, A. J., Chotikunnan, P., & Ma'arif, A. (2025). Machine Learning 5.0 In-depth Analysis Trends in Classification. *Scientific Journal of Computer Science*, *1*(1), 1–15. Retrieved from <https://journal.futuristech.co.id/index.php/sjcs/article/view/18>
- Rizkallah, L. W. (2025a). Enhancing the performance of gradient boosting trees on regression problems. *Journal of Big Data*, *12*(1). <https://doi.org/10.1186/s40537-025-01071-3>
- Rizkallah, L. W. (2025b). Optimizing SVM hyperparameters for satellite imagery classification using metaheuristic and statistical techniques. *International Journal of Data Science and Analytics*, *20*(5), 4945–4962. <https://doi.org/10.1007/s41060-025-00762-7>
- Sasmal, B., Hussien, A. G., Das, A., & Dhal, K. G. (2023). A Comprehensive Survey on Aquila Optimizer. *Archives of Computational Methods in*

- Engineering*, 30(7), 4449–4476. <https://doi.org/10.1007/s11831-023-09945-6>
- Simanjuntak, A., Lumbantoruan, R., Sianipar, K., Gultom, R., Simaremare, M., & Situmeang, S. (2024). Studi dan Analisis Hyperparameter Tuning IndoBERT Dalam Pendeteksian Berita Palsu, *13*, 60–67.
- Subaşı, N. (2024). Comprehensive Analysis of Grid and Randomized Search on Dataset Performance. *European Journal of Engineering and Applied Sciences*, 7(2), 77–83.
- Suryana, S. E., Warsito, B., & Suparti, S. (2021). Penerapan Gradient Boosting Dengan Hyperopt Untuk Memprediksi Keberhasilan Telemarketing Bank. *Jurnal Gaussian*, 10(4), 617–623.
- Syahputra, F., Yani, D., Tanjung, H., Verina, W., & Ihsan, O. M. (2025). Analisis Kinerja Algoritma Klasifikasi terhadap Dataset Penerimaan Pegawai Outsourcing, *14*(April), 326–335.
- Thalange, A. V., Patil, A. R., & Athavale, V. A. (2024). A Review of Artificial Intelligence and Machine Learning for Vaccine Research. *Lecture Notes in Electrical Engineering*, 1194(1), 85–101. [https://doi.org/10.1007/978-981-97-2839-8\\_7](https://doi.org/10.1007/978-981-97-2839-8_7)
- Tobing, C. J. L., Wijayakusuma, I. G. N. L., & Ida, L. P. (2025). Perbandingan Kinerja IndoBERT dan MBERT untuk Deteksi Berita Hoaks Politik dalam Bahasa Indonesia, *14*(1), 114–123.
- Turaina, R., & Saputra, R. (2024). Optimalisasi Deteksi Malware pada Platform Android dengan Pendekatan Ensemble Machine Learning, *7*(3), 394–403.
- Wijaya, H., Hostiadi, D. P., & Triandini, E. (2024). OPTIMIZATION OF XGBOOST ALGORITHM USING PARAMETER TUNNING IN RETAIL

SALES PREDICTION Jurnal Nasional Pendidikan Teknik Informatika :

JANAPATI | 770, 13(3), 769–786.

Wijoyo, A., Saputra, A. Y., Ristanti, S., Rafly, S., Ban, S., Komputer, F. I., ... No,

P. (2024). Pembelajaran Machine Learning, 3(2), 375–380.

Yulianti, S. E. H., Soesanto, O., & Sukmawaty, Y. (2022). Penerapan Metode

Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu

Kredit. *Journal of Mathematics: Theory and Applications*, 21–26.



## LAMPIRAN

### 1. Dokumentasi



### 2. Kode Program

```
import pandas as pd

from google.colab import files
upload_files = files.upload()

df = pd.read_csv('Product_Classification.csv')
df.head ()

df.info ()

df.isnull().sum()

df.describe()

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(18, 20))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of DataFrame')
plt.show()
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score,
classification_report

# Separate features (X) and target (y)
X = df.drop(['id', 'target'], axis=1) # Drop 'id' as it's an
identifier and 'target' is the variable to predict
y = df['target']

print("Features (X) shape:", X.shape)
print("target (y) shape:", y.shape)

from sklearn.preprocessing import StandardScaler

# Initialize the StandardScaler
scaler = StandardScaler()

# Fit and transform the feature columns (X)
X_scaled = scaler.fit_transform(X)

# Convert the scaled features back to a DataFrame, maintaining
column names
X_scaled = pd.DataFrame(X_scaled, columns=X.columns)

print("First 5 rows of scaled features (X_scaled):")
print(X_scaled.head())

# Update X to use the scaled features if needed for subsequent
steps
X = X_scaled

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42, stratify=y)

print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)

model = XGBClassifier(random_state=42,
use_label_encoder=False, eval_metric='logloss')
model.fit(X_train, y_train)

y_pred_vanilla = model.predict(X_test)
```

```

accuracy = accuracy_score(y_test, y_pred_vanilla)
report = classification_report(y_test, y_pred_vanilla)

import numpy as np
import xgboost as xgb
from sklearn.model_selection import cross_val_score

# === batas parameter ===
LR_BOUND = (0.1, 0.3)
DEPTH_BOUND = (3, 5)
TREE_BOUND = (50, 500)

# === CACHE UNTUK FITNESS ===
fitness_cache = {}

# fungsi random in bound
def random_solution():
    return np.array([
        np.random.uniform(*LR_BOUND),
        np.random.randint(*DEPTH_BOUND),
        np.random.randint(*TREE_BOUND)
    ], dtype=float)

# evaluasi fitness (FAST VERSION)
def evaluate(solution, X_train, y_train, cv=3):
    key = tuple(solution.round(5)) # cache key

    # cek cache
    if key in fitness_cache:
        return fitness_cache[key]

    lr, depth, n_tree = solution

    model = xgb.XGBClassifier(
        learning_rate=float(lr),
        max_depth=int(depth),
        n_estimators=int(n_tree),
        subsample=0.9,
        colsample_bytree=0.9,
        eval_metric="logloss",
        tree_method="hist"
    )

    scores = cross_val_score(model, X_train, y_train, cv=cv,
scoring='accuracy')
    fitness = 1 - scores.mean()

```

```

        fitness_cache[key] = fitness
    return fitness

# Aquila Optimizer
def aquila_optimizer(X_train, y_train, pop_size=10,
max_iter=20):
    # init
    pop = [random_solution() for _ in range(pop_size)]

from sklearn.model_selection import GridSearchCV
import xgboost as xgb

# definisi parameter grid untuk dibandingkan
param_grid = {
    "learning_rate": [0.1, 0.3],
    "max_depth": [3, 5],
    "n_estimators": [50, 500]
}

xgb_model_grid = xgb.XGBClassifier(
    subsample=0.9,
    colsample_bytree=0.9
)

from sklearn.model_selection import RandomizedSearchCV
import xgboost as xgb
import numpy as np

# definisi parameter distribusi
param_dist = {
    "learning_rate": np.linspace(0.1, 0.3),
    "max_depth": np.arange(3, 5),
    "n_estimators": np.arange(50, 500)
}

xgb_model = xgb.XGBClassifier(
    subsample=0.9,
    colsample_bytree=0.9
)

import matplotlib.pyplot as plt
import numpy as np

def plot_convergence(values, figsize=(6,4), fontsize=14,
title="Convergence Plot"):
    """

```

```
values: list atau numpy array berisi nilai fitness / loss  
per iterasi  
figsize: ukuran figure  
fontsize: ukuran tulisan label dan title  
""  
  
values = np.array(values)
```



The image shows a screenshot of an iThenticate similarity report. The report is for a document titled 'STEVI FRESHIA SIHOMBING.pdf' by the author 'STEVI FRESHIA SIHOMBING'. It provides various statistics: 12425 words, 79197 characters, 75 pages, and a file size of 1.2MB. The submission date is May 20, 2026, at 11:46 AM GMT+7, and the report date is May 20, 2026, at 11:48 AM GMT+7. The overall similarity is 9%, which is composed of 9% from the internet database, 4% from publications, and 5% from crossref. The report also lists excluded items such as submitted works, cited material, small matches, bibliographic material, and abstracts. A large watermark of Universitas Medan Area is visible in the background. At the bottom left, there is a CamScanner logo and the text 'Dipindai dengan CamScanner'.

**iThenticate** Similarity Report ID: oid.29477.139672192

PAPER NAME: **STEVI FRESHIA SIHOMBING.pdf** AUTHOR: **STEVI FRESHIA SIHOMBING**

WORD COUNT: **12425 Words** CHARACTER COUNT: **79197 Characters**

PAGE COUNT: **75 Pages** FILE SIZE: **1.2MB**

SUBMISSION DATE: **May 20, 2026 11:46 AM GMT+7** REPORT DATE: **May 20, 2026 11:48 AM GMT+7**

- **9% Overall Similarity**  
The combined total of all matches, including overlapping sources, for each database.
  - 9% Internet database
  - 4% Publications database
  - Crossref database
  - Crossref Posted Content database
- **Excluded from Similarity Report**
  - Submitted Works database
  - Cited material
  - Small Matches (Less then 15 words)
  - Bibliographic material
  - Abstract

CS Dipindai dengan CamScanner



## UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolang Nomor 1 Medan Fatate / Jalan Gedung PBSI, Medan 20223  
Kampus II : Jalan Sei Serayu Nomor 70 A / Jalan Setia Budi Nomor 79 B, Medan 20112 Telepon : (061) 8225602, 8201994  
Fax : (061) 8226331 IIP : 0811 607 259 website: www.uma.ac.id Email : univ\_medan@uma.ac.id

Nomor : 2511/FT/01.10/X/2025

6 Oktober 2025

Lampiran : -

Hal : Pembimbing Tugas Akhir

Yth. Pembimbing Tugas Akhir

**ANDRE HASUDUNGAN LBS S.Ti, M.Sc ( Sebagai Pembimbing )**

di Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : STEVI FRESHIA SIHOMBING

NIM : 228160029

Jurusan : TEKNIK INFORMATIKA

Maka dengan hormat kami mengharapkan kesediaan saudara :

**ANDRE HASUDUNGAN LBS S.Ti, M.Sc ( Sebagai Pembimbing )**

Adapun Tugas Akhir Skripsi berjudul :

**Optimasi Algoritma XGBoost menggunakan Aquilla Optimizer untuk Klasifikasi Produk Elektronik**

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,



Dr Eng. Supriatno.ST, MT.



# UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolang Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223  
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122  
Website: [www.teknik.uma.ac.id](http://www.teknik.uma.ac.id) E-mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

Nomo. : 350/FT.6/01.10/XI/2025

27 November 2025

Lamp : -

Hal : **Peneitian Dan Pengambilan Data Tugas Akhir**

Yth. Wakil Rektor Bidang Mutu Sumber Daya Manusia dan Perekonomian  
Jln. Kolang No.1  
Di  
Medan

Dengan hormat, kami mohon kesediaan bapak kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PRODI
1	Stevi Freshia Sihombing	228160029	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.**

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-matz untuk tujuan Ilmiah dan Skripsi, yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

**Optimasi Algoritma XGBoost menggunakan Aquila Optimizer untuk Klasifikasi Produk Elektronik.**

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya Surat ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Dekan

  
Widhiatno, ST, MT

Tembusan :

1. Ka. BPMPP
2. Mahasiswa
3. File

CS Dipindai dengan CamScanner



# UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolan Nomor 1 Medan Estate ☎ (061) 7360168, Medan 20223  
Kampus II : Jalan Setiabudi Nomor 79 B / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan 20122  
Website: [www.uma.ac.id](http://www.uma.ac.id) E-Mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

## SURAT KETERANGAN SELESAI PENELITIAN

Nomor : 196/UMA/B/01.7/II/2026

Yang bertanda tangan di bawah ini :

Nama : Dr. Ir. Rahmad Syah, M.Kom, IPM, ASEAN Eng, APEC Eng  
Jabatan : Wakil Rektor Bidang Mutu Sumber Daya dan Perekonomian  
NIDN : 0105058804

Dengan ini menerangkan bahwa mahasiswa yang Namanya tercantum di bawah ini :

Nama : Stevi Freshia Sihombing  
NPM : 228160029  
Program Studi : Teknik Informatika  
Fakultas : Teknik  
Status : (Mahasiswa / Dosen / Peneliti)

Telah melaksanakan dan menyelesaikan riset (penelitian) di lingkungan Universitas Medan Area dengan rincian sebagai berikut:

Judul Penelitian : Optimasi Algoritma *XGBoost* Menggunakan *Aquila Optimizer* untuk Klasifikasi Produk Elektronik  
Lokasi Penelitian : Laboratorium Komputer Program Studi Teknik Informatika Universitas Medan Area  
Hasil Penelitian : Berdasarkan penelitian ini, dapat disimpulkan bahwa penelitian ini membandingkan empat model *XGBoost* (Vanilla, Grid Search, Random Search, dan *Aquila Optimizer*) melalui 10 kali pengujian pada tiga dataset berbeda untuk memprediksi produk laris. Hasil evaluasi melalui *classification report*, *confusion matrix*, dan ROC-AUC menunjukkan bahwa *XGBoost* yang dioptimasi dengan *Aquila Optimizer* (*XGB AO*) merupakan model terbaik dengan tingkat akurasi dan stabilitas paling unggul, khususnya pada Dataset A dan C. Keunggulan ini didukung oleh analisis konvergensi yang membuktikan efektivitas *Aquila Optimizer* dalam menyeimbangkan tahap eksplorasi dan eksploitasi parameter secara konsisten. Dibandingkan metode optimasi lainnya, *XGB AO* menunjukkan kemampuan generalisasi yang lebih andal, menjadikannya solusi paling efektif untuk klasifikasi produk laris dalam penelitian ini  
Waktu Pelaksanaan : November – Desember 2025

Berdasarkan laporan hasil riset dan verifikasi data yang kami terima, yang bersangkutan telah menyelesaikan seluruh rangkaian kegiatan penelitiannya dengan baik.

Demikian surat ini diterbitkan untuk dapat digunakan seperlunya.

Medan, 02 Februari 2026  
Wakil Rektor Bidang Mutu Sumber  
Daya dan Perekonomian,

Dr. Ir. Rahmad Syah, M.Kom,  
IPM, ASEAN Eng, APEC Eng

