

Algoritma Extreme Gradient Boosting (XGBoost) dan Adaptive Boosting (AdaBoost) Untuk Klasifikasi Penyakit Tiroid

Anita Desiani¹, Siti Nurhaliza², Tri Febriani Putri³, Bambang Suprihatin⁴
^{1,2,3,4}Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Sriwijaya
Indralaya, Indonesia
Penulis Korespondensi : anita_desiani@unsri.ac.id

Abstrak— Penyakit tiroid adalah penyakit pada kelenjar tiroid yang dapat mengganggu aktivitas sehari-hari. Deteksi dini penyakit tiroid dapat memberikan dampak yang penting dalam mengoptimalkan pengembangan sistem deteksi dini yang lebih efektif dan akurat dalam mendeteksi penyakit. Pendekatan data *mining* dapat digunakan untuk menyelesaikan masalah ini dengan memanfaatkan berbagai algoritma yang tersedia, seperti *Adaptive Boosting* dan *Extreme Gradient Boosting*. Penelitian ini bertujuan untuk meningkatkan perkembangan prediksi penyakit tiroid secara dini dengan membandingkan kedua algoritma dengan memanfaatkan metode *percentage split*. Penelitian ini memberikan hasil pada algoritma *Adaptive Boosting* nilai akurasi sebesar 97%. Pada kelas 0 nilai presisi dan *recall* yaitu 98%, sedangkan pada kelas 1 sebesar 80% dan 90%. Sementara itu, pengujian menggunakan algoritma *Extreme Gradient Boosting* memberikan nilai akurasi sebesar 98%. Pada kelas 0 nilai presisi dan *recall* yaitu 99%, sedangkan untuk kelas 1 86% dan 90%. Berdasarkan perbandingan dengan mempertimbangkan nilai akurasi, presisi, dan *recall*, serta performa kedua algoritma tersebut, ditarik kesimpulan bahwa implementasi algoritma *Extreme Gradient Boosting* memiliki performa yang paling baik untuk deteksi penyakit tiroid.

Kata Kunci— Adaptive Boosting, Extreme Gradient Boosting, Klasifikasi, Tiroid

Abstract— *Thyroid disease is a disease of the thyroid gland that can interfere with daily activities. Early detection of thyroid disease can have an important impact in optimizing the development of early detection systems that are more effective and accurate in detecting the disease. Data mining approaches can be used to solve this problem by utilizing various available algorithms, such as Adaptive Boosting and Extreme Gradient Boosting. This research aims to improve the development of early thyroid disease prediction by comparing the two algorithms by utilizing the percentage split method. This research provides results if the Adaptive Boosting algorithm provides an accuracy value of 97%. In class 0, the precision and recall values are the same at 98%, while in class 1 it is 80% and 90%. Meanwhile, testing using the Extreme Gradient Boosting algorithm gives an accuracy value of 98%. In class 0, the same precision and recall values are 99%, while for class 1 it is 86% and 90%. Based on the comparison by considering the accuracy, precision, and recall values, as well as the performance of the two algorithms, it is concluded that the implementation of the Extreme Gradient Boosting algorithm has the best performance for thyroid disease detection.*

Keywords— Adaptive Boosting, Extreme Gradient Boosting, Classification, Thyroid

I. PENDAHULUAN

Tiroid adalah kelenjar kecil di leher yang terletak tepat di bawah jakun. Kelenjar tiroid tersusun atas dua bagian lobus kanan dan lobus kiri. Kelenjar tiroid memiliki ukuran sekitar 5 cm tinggi dan 2 sampai 3 cm lebar per lobus, serta tebal sekitar 2 sampai 3 cm [1]. Penyakit Tiroid merupakan gangguan pada kelenjar tiroid yang terjadi karena adanya kelainan pada bentuk dan fungsi kelenjar tiroid [2]. Penyebab atau faktor yang mengakibatkan seseorang menderita penyakit tiroid adalah usia dan jenis kelamin. Penyakit Tiroid umumnya terjadi pada orang yang berusia di atas 60 tahun dengan hipotiroidisme lebih sering menyerang wanita lansia dalam rentang usia ini. Riwayat penyakit keluarga yang berhubungan dengan autoimun dan obat-obatan atau zat yang mengandung yodium akan menaikkan risiko perkembangan penyakit tiroid [3]. Menurut statistic Globocan, Indonesia menempati urutan ke dua belas dengan jumlah kasus kanker tiroid sebesar 13.414 kasus dan paling banyak dari 35 jenis kanker yang berbeda dengan total 396.914 kasus. Secara keseluruhan, terdapat 234.511 kematian kanker dan 2.224 diantaranya merupakan kasus kematian kanker tiroid [4]. Berdasarkan data tersebut dapat dipastikan penyakit tiroid akan mengalami peningkatan yang signifikan, sehingga diperlukannya deteksi dini agar mengurangi risiko penyakit ini.

Melalui pendekatan data *mining*, sebagai metode penyelesaian matematika dapat digunakan untuk mendeteksi penyakit ini secara dini. Data *mining* adalah metode menggali informasi dan data yang sebelumnya tidak diketahui tetapi masih bisa dipahami dan diakses dengan cepat meskipun memiliki jumlah data yang besar [5]. Klasifikasi penyakit Tiroid sudah banyak dilakukan sebelumnya, salah satunya Supardianto *et al.* menerapkan

metode *KNNImputer* dalam memprediksi penyakit Tiroid dan menghasilkan nilai akurasi sebesar 93% [6], akan tetapi dalam penelitian tersebut nilai recall pada kelas 1 masih sangat rendah, yaitu sebesar 30%. Dalam penelitian lainnya yang dilakukan oleh Aversano *et al.* menggunakan metode *Extra Trees Classifier* untuk memprediksi penyakit tiroid. Penelitian ini menghasilkan nilai akurasi yang sebesar 84%, presisi sebesar 85% dan recall sebesar 84%[7]. Penelitian selanjutnya, Putri *et al.* melakukan perbandingan algoritma *Naïve Bayes* dan *K-Nearest Neighbors* (KNN) dalam memprediksi penyakit Tiroid dapat disimpulkan algoritma *Naïve Bayes* dengan data *testing* 63% dan data *training* 64% sedangkan akurasi yang dihasilkan pada algoritma *K-Nearest Neighbors* (KNN) adalah 63% [8].

Adaptive Boosting (AdaBoost) adalah salah satu teknik *ensemble* yang bertujuan untuk memperbaiki tingkat akurasi dari prediksi yang dibuat dengan menggunakan *loss function* fungsi eksponensial [9]. Sedangkan, *Extreme Gradient Boosting* (XGBoost) adalah algoritma yang lebih efisien dan *scalable* berbasis pohon [10]. Algoritma Adaboost dapat mengatasi perbedaan pada *weak classifier* dan dapat mengurangi nilai bias, selain itu Adaboost juga mampu mengurangi risiko *overfitting* dan meminimalisasi dari batas atas *error* [11]. Kekurangan dari AdaBoost, yaitu keterbatasan dalam penggunaan data yang hilang, keterbatasan dalam penggunaan pada data yang berukuran besar dan keterbatasan dalam penggunaan pada data yang kompleks [12]. Algoritma XGBoost memiliki beberapa kelebihan, yaitu dalam kinerja dan kompleksitas waktu serta memori yang terjangkau dan lebih stabil dalam melakukan prediksi terhadap kelas tidak seimbang, mengoptimalkan fungsi loss menggunakan derivatif kedua, yang memungkinkan penanganan lebih baik terhadap data yang kompleks dan non-linear [13]. Kekurangan dari algoritma XGBoost, yaitu model yang memiliki banyak parameter, sehingga penentuan nilai setiap parameter tergolong sulit karena menghabiskan waktu yang tidak sedikit [13].

Berdasarkan kelebihan dan kekurangan dari masing-masing algoritma, penelitian ini bertujuan membandingkan hasil dari algoritma XGBoost dan Adaboost serta menentukan kinerja algoritma terbaik antara *Extreme Gradient Boosting* (XGBoost) dan *Adaptive Boosting* (AdaBoost). Dalam penelitian ini, nilai akurasi dan presisi dari algoritma XGBoost dan AdaBoost juga akan diperoleh. Menggunakan nilai akurasi, recall, dan presisi dari algoritma AdaBoost dan XGBoost yang akan dievaluasi dalam penelitian ini. Nilai akurasi menggambarkan seberapa baik algoritma untuk mengklasifikasikan pasien secara akurat, presisi memberikan gambaran bahwa rasio pasien yang benar teridentifikasi positif penyakit tiroid. *Recall* menggambarkan rasio pasien yang benar teridentifikasi tiroid dari total seluruh jumlah pasien yang sebenarnya terkena penyakit tiroid [14]. Penelitian ini diharapkan dapat memberikan wawasan mengenai kinerja serta tingkat keberhasilan kedua algoritma dalam mengidentifikasi penyakit tiroid. Pemilihan algoritma yang optimal dapat membuka peluang untuk mengembangkan sistem deteksi dini yang lebih efektif dan akurat dalam mendeteksi penyakit. Diharapkan hasil penelitian ini akan berkontribusi penting dalam lingkup perawatan dan pengelolaan penyakit tiroid, serta meningkatkan perencanaan medis yang akurat dan tindakan dini untuk pasien.

II. METODOLOGI PENELITIAN

A. Deskripsi Data

Penelitian ini menggunakan dataset yang diperoleh dari situs *online* yang menyediakan berbagai dataset, yaitu dataset penyakit tiroid [15] dan format yang digunakan adalah csv. Data mengenai penyakit tiroid tersebut sebanyak 3772 data. Pada dataset tiroid tersebut terdiri dari 30 atribut diantaranya meliputi *Age*, *Sex*, *Sick*, *Pregnant*, *Thyroid Surgery*, *Lithium*, *Goitre*, *Tumor*, *Hypopituitary*, *Psych*, TSH, T3, TT4, T4U, FTI, dan atribut target pada dataset ini adalah *Class* yang terdiri dari 2 kelas yaitu *negative* dan *sick*. Berikut informasi mengenai atribut data yang dimanfaatkan dalam penelitian ini dapat dilihat pada Tabel 1.

TABLE I. ATRIBUT DAN INFORMASI DATA

Atribut	Keterangan Atribut	Data Kosong
Age	Usia subjek pada saat pemeriksaan	Null
Sex	F : Female, M : Male	Null
Thyroid surgery	Keterangan apakah pasien pernah menjalani operasi tiroid. (f : false, t : true)	Null
Sick	Keterangan apakah subjek sakit. (f : false, t : true)	Null
Pregnant	Keterangan apakah subjek hamil. (f : false, t : true)	Null
Lithium	Keterangan apakah subjek lithium. (f : false, t : true)	Null
Tumor	Keterangan apakah subjek mengalami tumor. (f : false, t : true)	Null
Goitre	Keterangan apakah pasien menderita gondok. (f : false, t : true)	Null

Hypopituitary	Keterangan apakah subjek mengalami penyakit kelenjar hipofisis. (f : false, t : true)	Null
Psych	Keterangan apakah subjek mengalami penyakit psikis. (f : false, t : true)	Null
TSH	Tingkat TSH dalam darah. (0.05-530)	369
T3	Tingkat T3 dalam darah. (0.05-10.6)	769
TT4	Tingkat TT4 dalam darah. (2-430)	231
T4U	Tingkat T4U dalam darah. (0.25-2.32)	384
FTI	Tingkat FTI dalam darah. (2-395)	285

B. Preprocessing Data

Preprocessing Data adalah sekumpulan tindakan yang diambil sebelum data digunakan untuk analisis [16]. *Preprocessing data* bertujuan untuk menghapus data yang tidak wajar, menormalkan dan mempersiapkan data sehingga model *machine learning* dapat bekerja lebih efisien [17]. Data penyakit Tiroid terdiri dari 30 atribut. Atribut yang tidak berpengaruh akan dihilangkan sehingga jumlahnya menjadi 16 atribut, yaitu *Age, Sex, Sick, Pregnant, Thyroid Surgery, Lithium, Goitre, Tumor, Hypopituitary, Psych, TSH, T3, TT4, T4U, FTI, dan Class*. 15 atribut tersebut akan digunakan untuk menentukan apakah seseorang memiliki penyakit Tiroid, sedangkan atribut *Class* merupakan hasil atau label dari faktor yang dialami penderita.

1) Missing Value (Data Kosong/Hilang)

Missing value adalah tantangan dalam analisis data, dimana nilai atribut dalam dataset tidak terdefinisi atau kosong. Dataset penyakit Tiroid memiliki 5 atribut yang mempunyai *missing value*, yaitu 369 pada kolom TSH, 769 pada kolom T3, 231 pada kolom TT4, 384 pada kolom T4U, dan 285 pada kolom FTI. Data *missing value* atau data hilang ini akan dihapus.

2) Normalisasi Data

Tabel 1 menunjukkan bahwa atribut TSH, T3, TT4, T4U, dan FTI pada dataset penyakit Tiroid memiliki rentang nilai yang sangat beragam, sehingga normalisasi data sangat diperlukan. Normalisasi data bertujuan untuk menyeimbangkan rentang nilai atribut pada skala tertentu sehingga memberikan klasifikasi yang lebih akurat. Normalisasi data dilakukan melalui penerapan metode penskalaan min-max, yang dihitung berdasarkan persamaan yang ditunjukkan dalam persamaan (1) [18].

$$\text{Normalized } (x) = \frac{\min \text{ Range } x + (x \text{ min Value}) (\max \text{ Range} - \min \text{ Range})}{\max \text{ Value} - \min \text{ Value}} \quad (1)$$

3) Pengujian Data

Pada pengujian data penyakit Tiroid menggunakan teknik *percentage split*. Pada *data mining*, *percentage split* membagikan dataset menjadi dua bagian yaitu subset pelatihan (*training set*) dan subset pengujian (*testing set*) [19].

C. Adaptive Boosting (AdaBoost)

Adaptive Boosting (AdaBoost) adalah teknik data *mining* pada metode klasifikasi yang berguna untuk meningkatkan tingkat akurasi [11]. AdaBoost digunakan untuk mengklasifikasi data kelasnya masing-masing, dalam penerapannya diawali dengan mencari kategori kelas berdasarkan nilai bobot yang dimiliki oleh kelas. AdaBoost memilih fungsi klasifikasi yang lemah dan digabungkan menjadi satu untuk menghasilkan fungsi klasifikasi yang baru, sehingga AdaBoost dapat dengan mudah digabungkan dengan metode klasifikasi pada data *mining*. Pada setiap iterasi, nilai bobot akan terus bertambah pada bobot nilai yang salah [20]. Beberapa tahapan algoritma AdaBoost, yaitu [21]:

1) Menentukan bobot pada tiap data dengan menggunakan rumus seperti yang terlihat pada persamaan (2)

$$w(x_i, y_i) = \frac{1}{N} \quad (2)$$

2) Mencari nilai Entropi dan *Information Gain* dengan menggunakan rumus seperti yang terlihat pada persamaan (3) dan (4)

$$Entropi(S) = \sum_{i=1}^n -p_i \log_2(p_i) \quad (3)$$

$$Information\ Gain = Entropi(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Entropi(S_i) \quad (4)$$

- 3) Mengklasifikasi data berdasarkan *Information Gain* tertinggi
- 4) Memperbarui nilai bobot baru
- 5) Menentukan error rate dan bobot yang dapat menggunakan rumus pada persamaan (5) dan (6)

$$w\varepsilon_t = \frac{1}{N}(x) \quad (5)$$

$$\alpha_t = \log\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (6)$$

- 6) Memperbarui bobot dengan pengamatan yang salah dengan menggunakan rumus pada persamaan (7)

$$D_{t+1} = \frac{1}{N} e^{\alpha_t} \quad (7)$$

Penjelasan rumus dalam konteks ini adalah w merepresentasikan bobot. Kemudian, data pelatihan ke- i dengan $i = 1, 2, \dots, n$, dinyatakan sebagai (x_i, y_i) . Himpunan kasus dilambangkan dengan S , sedangkan jumlah kasus pada partisi ke- i disebut S_i . Probabilitas S_i terhadap S dinyatakan sebagai P_i . Notasi $|S|$ mengacu pada jumlah kasus pada S . Nilai tingkat kesalahan (*error rate*) diberi simbol ε_t , dan jumlah percobaan atau iterasi diwakili oleh t . Banyaknya data secara keseluruhan dilambangkan dengan N , sedangkan banyaknya data yang berbeda antara hasil dan target dinyatakan sebagai x . Nilai bobot untuk pengklasifikasi diwakili oleh α_t , dan pembaruan bobot untuk pengamatan yang salah dinyatakan sebagai D_{t+1} .

D. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting merupakan kepanjangan dari XGBoost. Algoritma XGBoost merupakan teknik paralel yang mampu membangun pohon keputusan secara efisien dengan peningkatan berdasarkan *Gradient Boosting Decision Tree* (GBDT). Secara fundamental, XGBoost adalah metode *ensemble* yang didasarkan pada *gradient boosting tree* dan dirancang untuk menangani masalah klasifikasi serta regresi [22]. Keputusan diwakili oleh leaf node di dalam pohon. Dalam XGBoost, prediksi akhir diperoleh dengan menambahkan prediksi yang dibuat oleh model baru ke residual model sebelumnya. Leaf node pertama dalam model XGBoost diisi dengan nilai probabilitas untuk atribut yang diprediksi. Langkah-langkah dalam membangun pohon menggunakan metode XGBoost adalah sebagai berikut [23]:

- 1) Melakukan inisialisasi probabilitas pada prediksi awal Pr_i^1 di mana $i = 1, 2, \dots, n$
- 2) Menghitung residual dengan rumus yang ditunjukkan pada persamaan (8)

$$Residual_i^t = Y_i - Pr_i^r \quad (8)$$

- 3) Menghitung nilai similarity dengan rumus yang ditunjukkan pada persamaan (9)

$$SS_{mode} = \frac{(\sum Residual_i)^2}{\sum [Pr_i^t \times (1 - Pr_i^t)] + \lambda} \quad (9)$$

- 4) Menghitung nilai gain dengan rumus yang ditunjukkan pada persamaan (10)

$$Entropi(S) = \sum_{i=1}^n -p_i \log_2(p_i) \quad (10)$$

- 5) Menghitung nilai leaf output dengan rumus yang ditunjukkan pada persamaan (11)

$$Output(A)_i = \frac{\sum Residual_i}{\sum [Pr_i^t \times (1 - Pr_i^t)] + \lambda} \quad (11)$$

- 6) Menghitung nilai log odds dengan rumus yang ditunjukkan pada persamaan (12)

$$Log\ odds_i^t = \log\left(\frac{Pr_i^t}{1 - Pr_i^t}\right) \quad (12)$$

- 7) Memperbarui nilai probabilitas dengan rumus yang ditunjukkan pada persamaan (13)

$$Pr_i^{t+1} = \log(odds_i^t) + (\eta \times output(A)_i) \quad (13)$$

- 8) Menormalisasi nilai probabilitas menggunakan fungsi sigmoid biner yang ditunjukkan pada persamaan (14)

$$Sigmoid(Pr_i^{t+1}) = \frac{\exp^{Pr_i^{t+1}}}{1 + \exp^{Pr_i^{t+1}}} \quad (14)$$

- 9) Ulangi langkah-langkah dari langkah 2 sampai dengan 8

- 10) Melakukan evaluasi kinerja algoritma klasifikasi

E. Confussion Matriks

Confusion matrix merupakan metode evaluasi yang digunakan untuk mengukur performa algoritma [24]. Penilaian kinerja model menggunakan confusion matrix melibatkan empat istilah utama: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) [25]. Gambaran *confusion matrix* dapat dilihat pada Tabel 2.

TABLE II. TABEL CONFUSION MATRIKS

Class		Prediction	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Dari Tabel 2 dapat dihitung kinerja model pada penelitian dengan memanfaatkan nilai akurasi, presisi dan *recall*.

1) Akurasi

Akurasi adalah nilai prediksi diperoleh dari kumpulan data. Jika nilai akurasi tinggi maka menunjukkan model dapat memahami pola pada data dengan baik dan akan memberikan hasil yang sesuai dengan kenyataan. Jika nilai akurasi rendah maka menunjukkan model tidak dapat memahami pola pada data dengan baik dan akan memberikan hasil yang tidak sesuai dengan kenyataan. Untuk menghitung nilai akurasi digunakan persamaan (15) berikut [26]:

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \quad (15)$$

2) Presisi

Presisi merujuk pada kemampuan untuk menghasilkan hasil yang konsisten dari pengukuran yang dilakukan berulang kali dengan metode yang identik. Untuk menghitung nilai presisi digunakan persamaan (16) berikut [26]:

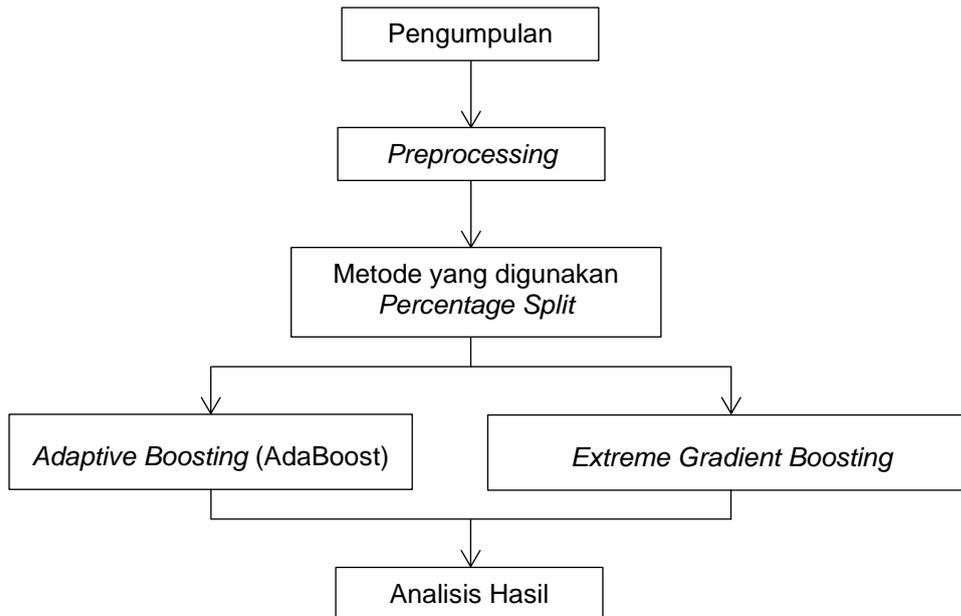
$$Presisi = \frac{TP}{TP + FP} \quad (16)$$

3) Recall

Recall adalah perbandingan rasio prediksi benar dengan total data yang benar. Untuk menghitung nilai recall digunakan persamaan (17) berikut [26]:

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

Hasil yang diperoleh dari evaluasi algoritma AdaBoost dan XGBoost akan dibandingkan untuk menentukan algoritma yang paling akurat dalam membuat model terbaik. Alur dari metode yang digunakan pada penelitian ini bisa dilihat pada Gambar 1.



Gambar 1. Metode yang digunakan dalam penelitian ini

III. HASIL DAN PEMBAHASAN

A. Adaptive Boosting (AdaBoost)

Pembagian data dilakukan dengan menggunakan metode *percentage split*, di mana 70% data digunakan untuk *training* dan 30% sisanya untuk *testing*. Penelitian ini menerapkan model AdaBoost serta pendekatan *percentage split*. Pada model AdaBoost digunakan hyperparameter *learning rate* sebesar 0.05 untuk mengontrol kontribusi setiap model dasar terhadap hasil akhir, sehingga memungkinkan proses pembelajaran berlangsung secara lebih stabil dan mengurangi risiko *overfitting*. Jumlah estimator ditetapkan sebesar 100 yang merepresentasikan jumlah iterasi boosting atau banyaknya model dasar (weak learners) yang dilatih secara berurutan. Model dasar yang digunakan adalah *DecisionTreeClassifier* dengan kedalaman maksimum (*max_depth*) sebesar 1 dan umum digunakan pada AdaBoost karena meminimalkan kompleksitas setiap learner. Algoritma boosting yang digunakan adalah SAMME, yaitu adaptasi dari AdaBoost untuk klasifikasi yang berbasis pada kesalahan klasifikasi. Hasil evaluasi performa model ditunjukkan melalui perhitungan *confusion matrix* yang dapat dilihat pada Tabel 3.

TABLE III. HASIL PEMODELAN ALGORITMA ADABOOST

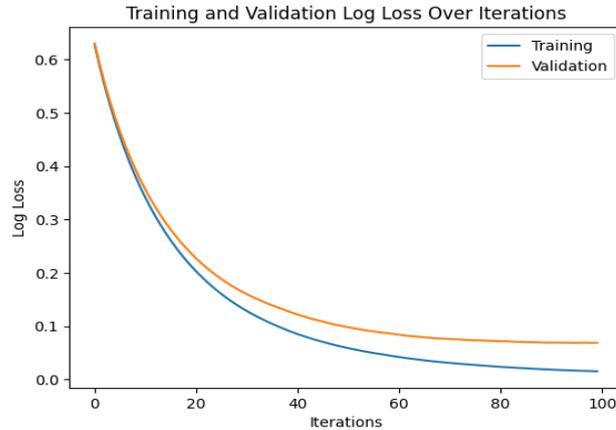
Class	Presisi	Recall	Accuracy
0	98%	98%	97%
1	80%	90%	

Hasil evaluasi kinerja pemodelan algoritma AdaBoost pada tahap pengujian menggunakan dataset penyakit Tiroid ditunjukkan dalam Tabel 3. Evaluasi dilakukan dengan menghitung nilai akurasi, *recall*, dan presisi. Presisi untuk kelas 0 mencapai 98%, yang menunjukkan bahwa sebagian besar data yang diklasifikasikan ke dalam kelas 0 diprediksi dengan benar. Selain itu, nilai *recall* untuk kelas 0 juga sebesar 98%. Pada kelas 1 nilai presisi pada model, yaitu 80% yang artinya model mampu mengidentifikasi data untuk kelas 1 diprediksi benar dan nilai *recall* adalah 90%. Hasil kinerja pada pengujian model AdaBoost pada penelitian ini diperoleh nilai akurasi sebesar 97%, artinya model mampu melakukan klasifikasi dengan benar. Berdasarkan hasil pemodelan algoritma AdaBoost dapat disimpulkan bahwa model AdaBoost yang digunakan untuk mengklasifikasi penyakit Tiroid memiliki performa yang sangat baik dalam hal ini. Dapat disimpulkan bahwa hasil evaluasi ini mengindikasikan bahwa model *percentage split* pada dataset penyakit Tiroid menghasilkan prediksi yang tepat dan andal dalam mengklasifikasikan penyakit Tiroid.

B. Extreme Gradient Boosting (XGBoost)

Pada penelitian ini, model XGBoost dilatih untuk mengklasifikasi pada dataset penyakit Tiroid. Model ini dipilih karena salah satu kelebihanannya adalah dapat menangani data yang kompleks dan tidak seimbang, seperti pada dataset penyakit Tiroid. Penerapan algoritma XGBoost pada penelitian ini menggunakan *percentage split*. Model XGBoost dibangun dengan serangkaian hyperparameter yang telah disesuaikan untuk meningkatkan performa klasifikasi, khususnya dalam menghadapi ketidakseimbangan kelas. Parameter *scale_pos_weight* digunakan untuk memberikan bobot lebih pada kelas minoritas dengan nilai ditentukan berdasarkan rasio jumlah kelas mayoritas

terhadap kelas minoritas. *learning rate* ditetapkan sebesar 0.05 untuk memperlambat proses pembelajaran dan memungkinkan model belajar secara lebih stabil dari kesalahan sebelumnya. Jumlah estimator diatur sebanyak 100 sebagai batas jumlah iterasi boosting. Untuk mengontrol kompleksitas model dan mencegah overfitting digunakan parameter regularisasi, yaitu alpha (regularisasi L1) sebesar 0.05, lambda (regularisasi L2) sebesar 0.1, dan gamma sebesar 0.1 yang berfungsi untuk meminimalkan pembelahan simpul yang tidak terlalu signifikan dalam pohon keputusan. Evaluasi performa model selama pelatihan dilakukan menggunakan metrik *log loss* yang cocok untuk tugas klasifikasi biner karena menghitung kehilangan informasi probabilistik. Perbandingan kinerja model pada data pelatihan dan validasi dapat dilihat melalui kurva *log loss* yang ditampilkan pada Gambar 2.



Gambar 2. Perbandingan nilai log loss training dan validasi

Gambar 2 menunjukkan nilai *log loss* menurun secara signifikan yang berarti model XGBoost yang dibangun sudah optimal dan mencapai konvergensi pada nilai yang rendah. Model XGBoost telah belajar dengan baik dari data *training* dan dapat mengaplikasikan hasilnya dengan baik pada data baru. Lebih lanjut, kurva pembelajaran menunjukkan bahwa model tidak mengalami *overfitting*. *Overfitting* terjadi ketika model terlalu kompleks dan belajar terlalu banyak dari data *training*, sehingga performanya buruk pada data baru. Oleh karena itu, model XGBoost yang dibangun dengan parameter yang dioptimalkan ini menunjukkan potensi untuk diterapkan dalam praktik untuk mengklasifikasikan data dengan akurasi yang tinggi. Pada proses ini, model XGBoost akan memberikan nilai akurasi di mana nilai tersebut mengartikan sejauh mana model dapat melakukan klasifikasi dengan benar. Tabel 4 menampilkan akurasi, presisi, dan *recall* dari data yang telah dilatih sebelumnya.

TABLE IV. HASIL PEMODELAN ALGORITMA XGBOOST

Class	Presisi	Recall	Accuracy
0	99%	99%	99%
1	86%	90%	

Tabel 4 memperlihatkan bahwa hasil evaluasi kinerja model saat pengujian memanfaatkan dataset penyakit Tiroid. Nilai akurasi, presisi, dan *recall* dihitung untuk menyelesaikan evaluasi Untuk kelas 0, model menunjukkan nilai presisi sebesar 99%, yang berarti mayoritas data yang diklasifikasikan ke dalam kelas 0 diprediksi dengan benar, dan nilai *recall* untuk kelas 0 juga 99%. Pada kelas 1 nilai presisi pada model sebesar 86% yang artinya model mampu mengidentifikasi data untuk kelas 1 diprediksi benar dan nilai *recall* adalah 90%. Hasil kinerja pada pengujian model XGBoost pada penelitian ini diperoleh nilai akurasi sebesar 98%, artinya model mampu melakukan klasifikasi dengan benar. Dapat disimpulkan bahwa model XGBoost yang digunakan untuk mengklasifikasi penyakit Tiroid memiliki prediksi yang akurat dan dapat diterapkan untuk memprediksi penyakit Tiroid.

C. Perbandingan Hasil

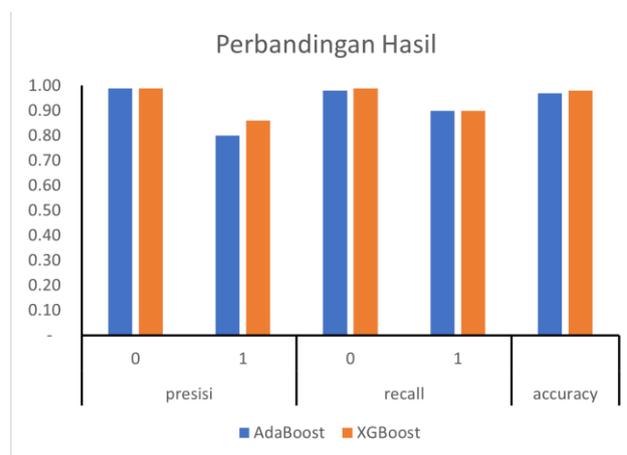
Berdasarkan penelitian sebelumnya, Supardianto *et al.* menerapkan metode *KNNImputer* dalam memprediksi penyakit tiroid dan menghasilkan nilai akurasi sebesar 93%. Namun, nilai recall pada kelas 1 masih tergolong sangat rendah, yaitu sebesar 30% yang menunjukkan bahwa model tersebut kurang mampu mengenali pasien dengan penyakit tiroid secara optimal. Aversano *et al.* menerapkan metode *Extra Trees Classifier* untuk memprediksi penyakit tiroid. Penelitian ini memperoleh nilai akurasi sebesar 84%, presisi sebesar 85% dan recall sebesar 84%. Putri *et al.* melakukan perbandingan antara algoritma *Naïve Bayes* dan *K-Nearest Neighbors* (KNN) dalam memprediksi penyakit Tiroid. Hasil penelitian menunjukkan bahwa algoritma *Naïve Bayes* menghasilkan akurasi sebesar 64% dan algoritma KNN menghasilkan akurasi sebesar 63%. Berbeda dengan penelitian-penelitian sebelumnya yang masih menunjukkan keterbatasan dalam evaluasi model, penelitian ini menerapkan dua metode *ensemble learning*, yaitu AdaBoost dan XGBoost pada dataset penyakit Tiroid dengan menggunakan

teknik *percentage split*. Tabel 5 menunjukkan perbandingan hasil evaluasi AdaBoost dan XGBoost dengan penelitian lainnya.

TABLE V. PERBANDINGAN HASIL EVALUASI ADABOOST DAN XGBOOST DENGAN PENELITIAN LAIN

<i>Authors</i>	<i>Method</i>	<i>Presisi</i>	<i>Recall</i>	<i>Accuracy</i>
Supardianto <i>et al.</i> [6]	<i>KNNImputer</i>	86,33%	50,66%	93%
Aversano <i>et al.</i> [7]	<i>Extra Trees Classifier</i>	85%	84%	84%
Putri <i>et al.</i> [8]	<i>Naïve Bayes</i>	-	-	64%
	<i>K-Nearest Neighbors (KNN)</i>	53%	51,5%	63%
(Diajukan)	<i>Adaptive Boosting (AdaBoost)</i>	89%	94%	97%
	<i>Extreme Gradient Boosting (XGBoost)</i>	92,5%	94,5%	99%

Berdasarkan tabel 5, metode AdaBoost dan XGBoost menunjukkan peningkatan performa yang signifikan dibandingkan dengan metode-metode yang digunakan pada penelitian sebelumnya. Untuk memberikan gambaran yang lebih jelas, gambar 3 menunjukkan perbandingan hasil evaluasi performa model pada metode AdaBoost dan XGBoost yang meliputi presisi, *recall*, dan akurasi.



Gambar 3. Perbandingan hasil performa pada metode AdaBoost dan XGBoost

Gambar 3 menunjukkan bahwa algoritma AdaBoost dan XGBoost memiliki potensi yang menjanjikan dalam mengklasifikasikan pasien penyakit Tiroid. Algoritma AdaBoost mencapai akurasi sebesar 97%, menunjukkan kemampuannya dalam mengklasifikasikan data dengan benar secara keseluruhan. Analisis presisi dan *recall* untuk masing-masing kelas menunjukkan performa yang baik. Metode AdaBoost mencapai tingkat presisi sebesar 98% pada kelas 0, yang menunjukkan bahwa 98% dari data yang diprediksi sebagai kelas 0 benar-benar berasal dari kelas tersebut. Nilai *recall* juga mencapai 98%, yang berarti bahwa 98% dari data kelas 0 benar ditemukan dengan menggunakan model ini. Kemudian, pada kelas 1 algoritma AdaBoost mencapai presisi 80%, menunjukkan bahwa 80% dari data yang diprediksi sebagai kelas 1 benar berasal dari kelas tersebut. Sedangkan *recall* mencapai 90%, yang berarti bahwa 90% dari data kelas 1 benar ditemukan dengan menggunakan model AdaBoost.

Performa XGBoost menunjukkan keunggulan dengan akurasi mencapai 98%, yang menandakan bahwa model ini sangat efektif dalam melaksanakan tugas klasifikasi secara keseluruhan. Untuk kelas 0, algoritma XGBoost mencapai presisi 99%, yang menunjukkan bahwa data yang diprediksi sebagai kelas 0 memang berasal dari kelas tersebut. *Recall* untuk kelas 0 juga mencapai 99%, yang menunjukkan bahwa model dapat mengidentifikasi 99% dari seluruh data yang benar-benar berasal dari kelas 0. Pada kelas 1, algoritma XGBoost mencapai presisi 86%, yang menunjukkan bahwa 86% data yang diprediksi sebagai kelas 1 memang berasal dari kelas tersebut. *Recall* untuk kelas 1 mencapai 90%, yang menunjukkan bahwa algoritma berhasil mengidentifikasi 90% dari semua data yang benar-benar berasal dari kelas 1. Meskipun tidak terlalu signifikan, algoritma XGBoost menunjukkan nilai presisi, akurasi, *recall* yang lebih tinggi dibandingkan dengan algoritma AdaBoost. Dapat disimpulkan bahwa model ini memberikan kinerja prediksi yang sangat baik dalam mengklasifikasikan dataset penyakit tiroid. Pemilihan algoritma terbaik tergantung pada kebutuhan dan preferensi, AdaBoost menawarkan kemudahan implementasi, sedangkan XGBoost memberikan performa yang lebih baik dengan akurasi, presisi, dan *recall* yang lebih tinggi.

IV. KESIMPULAN

Berdasarkan penelitian ini, dapat ditunjukkan bahwa penerapan algoritma *Adaptive Boosting* (AdaBoost) dan *Extreme Gradient Boosting* (XGBoost) sama-sama menunjukkan kinerja yang baik dalam mengklasifikasikan penyakit tiroid. Namun, dari analisis performa masing-masing algoritma, terlihat bahwa algoritma XGBoost menunjukkan performa yang lebih unggul dibandingkan dengan algoritma AdaBoost. Hal ini disebabkan oleh persentase performa XGBoost yang secara konsisten lebih tinggi dibandingkan dengan AdaBoost, yaitu nilai akurasi AdaBoost sebesar 97%, sedangkan nilai akurasi XGBoost sebesar 98%. Kemudian, pada kelas 0 nilai presisi dan *recall* algoritma AdaBoost sebesar 98%, sedangkan pada algoritma XGBoost sebesar 99% meskipun perbedaan nilai-nilai tersebut tidak terlalu signifikan. Namun, untuk nilai presisi pada kelas 1, algoritma AdaBoost sebesar 80%, sedangkan pada XGBoost sebesar 86%. Algoritma XGBoost lebih unggul dalam perbandingan klasifikasi penyakit tiroid. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa penerapan algoritma XGBoost adalah metode yang paling efisien dalam mengklasifikasikan penyakit tiroid. Hasil dari pemodelan yang dikembangkan dalam penelitian ini memiliki potensi untuk dioptimalkan menjadi sistem deteksi dini untuk penyakit tiroid, yang diharapkan dapat memberikan manfaat substansial bagi pengguna, terutama bagi tenaga kesehatan dan masyarakat umum.

DAFTAR PUSTAKA

- [1] K. Salman and E. Sonuc, "Thyroid Disease Classification Using Machine Learning Algorithms," *J. Phys. Conf. Ser.*, vol. 1963, no. 1, 2021, doi: 10.1088/1742-6596/1963/1/012140.
- [2] Y. A. Ginting, F. Lumongga, and E. R. D. Sitorus, "Hubungan Kadar T3, T4, dan TSH dengan Lesi pada Kelenjar Tiroid," *J. Kedokt. Methodist*, vol. 13, no. 4, p. 519, 2020.
- [3] S. Jubair, A. S. Nsaif, A. H. Abdullah, and I. H. Dhefer, "Vitamin D deficiency is associated with thyroid diseases," *J. Phys. Conf. Ser.*, vol. 1853, no. 1, 2021, doi: 10.1088/1742-6596/1853/1/012036.
- [4] Mahyuni and Munar, "Sistem Pakar Diagnosa Penyakit Campak," *J. TIKA*, vol. 06, no. 01, pp. 81–87, 2021, [Online]. Available: <http://journal.umuslim.ac.id/index.php/tika/article/view/435>
- [5] Y. Mardi, "Data Mining: Klasifikasi Menggunakan Algoritma C4.5," *Edik Inform.*, vol. 2, no. 2, pp. 213–219, 2019, doi: 10.22202/ei.2016.v2i2.1465.
- [6] Supardianto, Lalu Mutawalli, and Wafiah Murniati, "Penerapan Knnimputer Dalam Mengolah Data Missing Value Untuk Membantu Meningkatkan Akurasi Support Vector Machine Klasifikasi Penyakit Tiroid," *J. Inform. Teknol. dan Sains*, vol. 4, no. 4, pp. 386–390, 2022, doi: 10.51401/jinteks.v4i4.2077.
- [7] L. Aversano *et al.*, "Thyroid disease treatment prediction with machine learning approaches," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 1031–1040. doi: 10.1016/j.procs.2021.08.106.
- [8] S. J. Putri, A. Pratama, K. Attaqwa, and Rahmaddeni, "Perbandingan Kinerja Algoritma Naïve Bayes Dan K-Nearest Neighbors (KNN) Dalam Memprediksi Penyakit Tiroid," vol. 7, no. 1, pp. 20–24, 2020.
- [9] A. Bisri and R. S. Wahono, "Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree," *J. Intell. Syst.*, vol. 1, no. 1, pp. 27–32, 2015.
- [10] M. K. Nasution, R. R. Saedudin, and V. P. Widartha, "Perbandingan Akurasi Algoritma Naïve Bayes Dan Algoritma Xgboost Pada Klasifikasi Penyakit Diabetes," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 9765–9772, 2021.
- [11] R. T. Febianto, D. Suranti, and R. T. Alinse, "Penerapan Algoritma Adaboost dalam Mengetahui Pola Pengguna KB di Puskesmas Tanjung Harapan," *J. Sci. Soc. Res.*, vol. 4307, no. 1, pp. 145–155, 2024, [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [12] S. Mulyati, T. Informatika, U. Pamulang, and C. Pelanggan, "261256-Penerapan-Resampling-Dan-Adaboost-Untuk-a52Cf03a," vol. 2, no. 4, pp. 190–199, 2017.
- [13] P. Septiana Rizky, R. Haiban Hirzi, and U. Hidayaturrohmah, "Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang," *J Stat. J. Ilm. Teor. dan Apl. Stat.*, vol. 15, no. 2, pp. 228–236, 2022, doi: 10.36456/jstat.vol15.no2.a5548.
- [14] I. G. A. Purnajiwa Arimbawa and N. A. Sanjaya ER, "Penerapan Metode Adaboost Untuk Multi-Label Classification Pada Dokumen Teks," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 9, no. 1, p. 127, 2020, doi: 10.24843/jik.2020.v09.i01.p13.
- [15] N. S. W. I. S. A. Garavan Institute and J. Ross Quinlan, "Thyroid Sickness Determination." Accessed: Jun. 13, 2024. [Online]. Available: <https://www.kaggle.com/datasets/bidemiayinde/thyroid-sickness-determination>
- [16] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 273–281, 2024, doi: 10.57152/malcom.v4i1.1085.
- [17] E. D. Wahyuni, A. A. Arifiyanti, and M. Kustiyani, "Exploratory Data Analysis dalam Konteks Klasifikasi Data Mining," *Pros. Nas. Rekayasa Teknol. Ind. dan Inf. XIV Tahun 2019*, vol. 2019, no. November, pp. 263–269, 2019, [Online]. Available: <http://journal.itny.ac.id/index.php/ReTII>
- [18] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.
- [19] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.

- [20] N. Novianti, M. Zarlis, and P. Sihombing, "Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes," *J. Media Inform. Budidarma*, vol. 6, no. 2, p. 1200, 2022, doi: 10.30865/mib.v6i2.4017.
- [21] T. R. Mahesh *et al.*, "AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/9005278.
- [22] I. Muslim Karo Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," 2020.
- [23] M. W. Dwinanda, N. Satyahadewi, and W. Andani, "CLASSIFICATION OF STUDENT GRADUATION STATUS USING XGBOOST ALGORITHM," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 3, pp. 1785–1794, Sep. 2023, doi: 10.30598/barekengvol17iss3pp1785-1794.
- [24] A. Nugroho and Y. Religia, "Analisis Optimasi Algoritma Klasifikasi Naive Bayes menggunakan Genetic Algorithm dan Bagging," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, pp. 504–510, 2021, doi: 10.29207/resti.v5i3.3067.
- [25] I. Düntsch and G. Gediga, "Confusion Matrices and Rough Set Data Analysis," *J. Phys. Conf. Ser.*, vol. 1229, no. 1, pp. 0–6, 2019, doi: 10.1088/1742-6596/1229/1/012055.
- [26] D. Pramadhana, "Klasifikasi Penyakit Diabetes Menggunakan Metode CFS dan ROS dengan Algoritma J48 Berbasis Adaboost," *Edumatic J. Pendidik. Inform.*, vol. 5, no. 1, pp. 89–98, 2021, doi: 10.29408/edumatic.v5i1.3336.