KLASIFIKASI PENYAKIT ALZHEIMER MENGGUNAKAN METODE *XGBOOST*

SKRIPSI

Oleh :
AISYAH NUR FITRIYAH
NIM. 220605110033

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

KLASIFIKASI PENYAKIT ALZHEIMER MENGGUNAKAN METODE *XGBOOST*

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
AISYAH NUR FITRIYAH
NIM. 220605110033

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

HALAMAN PERSETUJUAN

KLASIFIKASI PENYAKIT ALZHEIMER MENGGUNAKAN METODE *XGBOOST*

SKRIPSI

Oleh :
AISYAH NUR FITRIYAH
NIM. 220605110033

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 9 Desember 2025

Pembimbing I,	Pembimbing II,
Prof. Dr. Muhammad Faisal, M.T	Shoffin Nahwa Utama, M.T
NIP. 19740510 200501 1 007	NIP. 19860703 202012 1 003

Mengetahui
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

KLASIFIKASI PENYAKIT ALZHEIMER MENGGUNAKAN METODE *XGBOOST*

SKRIPSI

Oleh :
AISYAH NUR FITRIYAH
NIM. 220605110033

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 30 Desember 2025

Susunan Dewan Penguji

Ketua Penguji	: Supriyono, M. Kom NIP. 19841010 201903 1 012	( )
Anggota Penguji I	: Okta Qomaruddin Aziz, M.Kom NIP. 19911019 201903 1 013	( )
Anggota Penguji II	: Prof. Dr. Muhammad Faisal, M.T NIP. 19740510 200501 1 007	( )
Anggota Penguji III	: Shoffin Nahwa Utama, M.T NIP. 19860703 202012 1 003	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Aisyah Nur Fitriyah
NIM	: 220605110033
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi Penyakit Alzheimer Menggunakan Metode *XGBoost*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 30 Desember 2025
Yang membuat pernyataan,

Aisyah Nur Fitriyah
NIM.220605110033

MOTTO

"Berapapun usiamu, selalu ada hal baru untuk dipelajari. Jika kau menganggapnya penyesalan, tamatlah sudah. Namun jika kau menganggapnya sebagai pelajaran, itu akan menjadi hal yang baru"
(18 Again)

"Kehidupan adalah rangkaian sebuah pilihan. Kita perlu memilih hal yang tepat untuk mendapatkan hasilnya"
(Itaewon Class)

HALAMAN PERSEMBAHAN

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT
atas rahmat dan hidayah-Nya, sehingga penulis dapat
menyelesaikan penulisan skripsi ini dengan baik.
Penulis persembahkan karya ini kepada:

Ibu tercinta, Yeni Mustikasari
Yang tidak pernah lelah mendoakan, menguatkan, dan
menemani penulis sampai sejauh ini

Ayah tercinta, Alm. Ach. Fathori
Semoga Allah SWT melapangkan kuburnya, mengampuni segala salahnya,
dan menempatkannya di tempat terbaik di sisi-Nya. .

Saudara-saudaraku, St. Sukma Fany Karimah, Nur Karomatus Sufiati,
Norma Aulia Safitri, dan Nayla Faadiyah Nur Safitri, terima kasih atas dukungan
dan doa yang selalu menyertai.

Ponakan-ponakanku, Muhammad Kenzie Arkana dan
Nadzira Almahyra Mecca , terima kasih sudah menjadi
penyemangat kecil di rumah.

Dan untuk diri sendiri, terima kasih karena sudah bertahan
tetap berusaha dan tidak menyerah sampai skripsi ini selesai.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbil 'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah *subhanahu wa ta'ala* atas rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Klasifikasi Penyakit Alzheimer Menggunakan Metode *XGBoost*". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika.

Penyusunan skripsi ini bukan perjalanan yang penulis lalui sendirian. Ada banyak tangan yang membantu, banyak doa yang diam-diam dipanjatkan, dan banyak perhatian yang mungkin terlihat sederhana, tetapi begitu berarti. Karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membersamai penulis sampai skripsi ini selesai, khususnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika sekaligus dosen penguji pertama, yang telah memberikan arahan dan dukungan selama penulis menempuh studi, serta berkenan menguji dan memberikan kritik, saran, dan masukan yang membangun demi penyempurnaan skripsi ini.

4. Prof. Dr. Muhammad Faisal, M.T, selaku dosen pembimbing utama yang dengan sabar membimbing, mengarahkan, dan meluangkan waktu untuk memberikan masukan sehingga skripsi ini dapat terselesaikan.
5. Shoffin Nahwa Utama, M.T, selaku dosen pembimbing kedua yang telah memberikan bimbingan, perhatian, serta saran yang sangat membantu dalam proses penyusunan skripsi ini.
6. Okta Qomaruddin Aziz, M.Kom, selaku dosen penguji kedua yang telah memberikan evaluasi serta arahan yang berharga sehingga skripsi ini menjadi lebih baik.
7. Nia Faricha, S.Si., selaku admin Program Studi Teknik Informatika, yang telah banyak membantu penulis dalam urusan administrasi, memberikan informasi, serta memudahkan berbagai proses selama perkuliahan maupun dalam penyusunan skripsi ini.
8. Seluruh Dosen, Laboran, dan Staf Program Studi Teknik Informatika yang telah memberikan ilmu, dukungan, serta bantuan selama penulis menjalani perkuliahan hingga penyusunan skripsi ini selesai.
9. *My Hero*, Ayah tercinta, almarhum Ach. Fathori, terima kasih atas kasih sayang, perjuangan, dan jejak kebaikan yang menjadi pegangan penulis sampai hari ini. Meski Ayah tidak lagi mendampingi secara langsung, doa dan nama Ayah selalu penulis bawa dalam setiap langkah. Semoga Allah subhanahu wa ta'ala mengampuni dosa-dosa Ayah, melapangkan kuburnya, dan menempatkannya di tempat terbaik di sisi-Nya.

10. Ibu tercinta, Yeni Mustikasari, terima kasih atas doa yang tidak pernah putus, kesabaran yang tidak pernah habis, serta ketulusan yang selalu menjadi kekuatan terbesar bagi penulis dalam menyelesaikan skripsi ini.
11. Saudara-saudara penulis, St. Sukma Fany Karimah, Nur Karomatus Sufiati, Norma Aulia Safitri, dan Nayla Faadiyah Nur Safitri, terima kasih atas perhatian, dukungan, dan doa yang selalu mengiringi penulis.
12. Ponakan-ponakan penulis, Muhammad Kenzie Arkana dan Nadzira Almahyra Mecca, terima kasih sudah menjadi penyemangat dengan cara kalian sendiri, menghadirkan tawa kecil yang sering kali membuat penulis kembali kuat.
13. Firna, terima kasih sudah menjadi teman sejak awal perkuliahan, ketika semuanya masih terasa baru dan penulis masih belajar menyesuaikan diri. Terima kasih karena selalu ada dalam prosesnya, bukan hanya saat semuanya berjalan lancar, tetapi juga ketika penulis berada di titik lelah dan mulai ragu. Dukungan, semangat, dan kehadiranmu sering kali menjadi pengingat bahwa penulis tidak perlu melewati semuanya sendirian, dan itu sangat berarti hingga skripsi ini akhirnya selesai.
14. Dita Suci Yofana, terima kasih sudah hadir dengan caramu sendiri di perjalanan ini. Terima kasih untuk kebersamaan yang sederhana tapi menguatkan, untuk perhatian yang sering datang di saat penulis butuh, dan untuk bantuan yang mungkin tidak selalu terlihat besar, tetapi sangat berarti dalam proses penulis menyelesaikan skripsi ini.

15. Qonita Nashifa Anabila, terima kasih karena selalu meluangkan waktu, meski sama-sama sibuk, dan tetap memilih untuk ada. Kehadiranmu menjadi salah satu hal yang membuat penulis merasa tidak sendirian, dan dukunganmu membuat langkah penulis terasa lebih ringan untuk dijalani sampai akhir.
16. Radifan Roihanul Fiqri, terima kasih atas semangat yang kamu tularkan, atas bantuan yang kamu berikan tanpa banyak bicara, dan atas dukungan yang kamu jaga dengan konsisten. Hal-hal seperti itu yang sering kali justru paling menenangkan dan paling penulis ingat dalam proses panjang ini.
17. Lailatul Ilmi Silviana, terima kasih atas dukungan, perhatian, dan kebaikan yang kamu berikan selama proses penulis menyusun skripsi ini. Kehadiranmu, meski di tengah kesibukan masing-masing, tetap terasa berarti dan membantu penulis menjaga semangat sampai akhir.
18. Seluruh warga Teknik Informatika, khususnya angkatan 2022 Infinity, terima kasih atas kebersamaan, semangat, dan dukungan yang penulis rasakan selama menjalani perkuliahan. Penulis bersyukur pernah menjadi bagian dari perjalanan ini, berbagi cerita, tugas, tawa, dan proses belajar bersama.
19. Annora Putri Ardelia dan Adinda Nur Brillyana, terima kasih sudah menjadi teman sejak masa SMP hingga sekarang. Meski perjalanan membawa kita ke kampus yang berbeda-beda, penulis bersyukur persahabatan ini tetap bertahan dan saling menjaga serta berdoa satu sama lain.

20. Utami Trisna Febriyanti dan Nisfal Laily, terima kasih sudah menjadi teman di rumah yang penulis anggap seperti keluarga sendiri. Penulis bersyukur punya kalian, karena kalian membantu penulis tetap bertahan sampai skripsi ini selesai.
21. Moh. Musa Al Kadzim, terima kasih karena sudah memilih untuk tetap tinggal dan berjalan bersama penulis. Terima kasih atas bantuan yang tulus tanpa pamrih, karena selalu menjadi orang pertama yang penulis cari saat senang, sedih, sakit, maupun saat membutuhkan pertolongan. Terima kasih sudah banyak mengalah, menjaga, dan memberikan perhatian, serta meluangkan waktu, tenaga, dan pengorbanan yang tidak sedikit demi penulis. Kehadiranmu adalah bagian penting yang membuat penulis mampu menyelesaikan skripsi ini sampai akhir.
22. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu penulis baik secara langsung maupun tidak langsung sejak awal perkuliahan hingga terselesaikannya skripsi ini. Semoga Allah subhanahu wa ta'ala membalas semua kebaikan dengan sebaik-baik balasan.
23. Dan terakhir, penulis berterima kasih kepada diri sendiri, karena sudah bertahan sampai sejauh ini. Terima kasih karena tetap berusaha meski tidak selalu mudah, karena memilih menyelesaikan apa yang sudah dimulai, dan karena tidak menyerah saat proses terasa panjang. Semoga setelah ini penulis tetap menjadi pribadi yang terus belajar, terus berproses, dan terus berani melangkah.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis menerima kritik dan saran yang membangun agar skripsi ini dapat menjadi lebih baik. Penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca, serta menjadi kontribusi kecil bagi perkembangan ilmu pengetahuan.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Malang, 9 Desember 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Fitriyah, Aisyah Nur. 2025. **Klasifikasi Penyakit Alzheimer Menggunakan Metode *XGBoost***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Muhammad Faisal, M.T (II) Shoffin Nahwa Utama, M.T.

.

Kata kunci: *Alzheimer, XGBoost, ADASYN, Bayesian Optimization, Feature Selection, K-Fold Cross Validation*

Penyakit Alzheimer merupakan gangguan neurodegeneratif progresif yang memerlukan deteksi dini untuk memaksimalkan efektivitas penanganan medis. Penelitian ini bertujuan untuk membangun model klasifikasi penyakit Alzheimer yang akurat menggunakan algoritma *Extreme Gradient Boosting* (XGBoost) serta menganalisis pengaruh teknik *preprocessing* data terhadap performa model. Dataset yang digunakan terdiri dari 2.149 data pasien dengan 35 fitur klinis. Penelitian ini menerapkan seleksi fitur berbasis korelasi *pearson*, penanganan ketidakseimbangan data menggunakan ADASYN (*Adaptive Synthetic Sampling*), serta optimasi *hyperparameter* menggunakan *Bayesian Optimization*. Validasi model dilakukan secara menyeluruh menggunakan metode *5-Fold Cross Validation* untuk menguji konsistensi performa. Hasil penelitian menunjukkan bahwa model XGBoost mampu memberikan performa klasifikasi yang sangat baik. Capaian terbaik diperoleh pada skenario 10, yaitu tanpa *balancing* dengan rasio pembagian data 60:40 dan seleksi fitur *threshold* 0.15, yang menghasilkan akurasi pengujian sebesar 95.58%, presisi 94.63%, *recall* 92.76%, dan *f1-score* 93.69%. Stabilitas model ini terkonfirmasi melalui *5-Fold Cross Validation* yang mencatatkan rata-rata akurasi sebesar 95.58% dengan standar deviasi yang rendah yaitu 0.46%. Analisis mendalam menunjukkan bahwa penggunaan ADASYN pada dataset ini justru menurunkan performa model dibandingkan data asli akibat distorsi distribusi, sementara strategi *slow learning* dengan struktur pohon sederhana terbukti paling efektif. Penelitian ini memberikan kontribusi dalam penyediaan model deteksi dini Alzheimer yang tidak hanya berakurasi tinggi dan stabil, tetapi juga efisien dalam penggunaan fitur.

ABSTRACT

Fitriyah, Aisyah Nur. 2025. **Classification of Alzheimer's Disease Using XGBoost Method**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University Maulana Malik Ibrahim Malang. Supervisors: (I) Prof. Dr. Muhammad Faisal, M.T (II) Shoffin Nahwa Utama, M.T

Keywords: Alzheimer, XGBoost, ADASYN, Bayesian Optimization, Feature Selection, K-Fold Cross Validation.

Alzheimer's disease is a progressive neurodegenerative disorder in which early detection is essential to maximize the effectiveness of medical intervention. This study aims to develop an accurate Alzheimer's disease classification model using the Extreme Gradient Boosting (XGBoost) algorithm and to analyze the impact of data preprocessing techniques on model performance. The dataset consists of 2,149 patient records with 35 clinical features. This research applies Pearson correlation-based feature selection, handles class imbalance using Adaptive Synthetic Sampling (ADASYN), and optimizes hyperparameters through Bayesian Optimization. Model validation is conducted comprehensively using 5-Fold Cross Validation to evaluate performance consistency. The results indicate that XGBoost achieves excellent classification performance. The best outcome is obtained in Scenario 10, which uses no balancing, a 60:40 train–test split, and feature selection with a Pearson threshold of 0.15, yielding a test accuracy of 95.58%, precision of 94.63%, recall of 92.76%, and an F1-score of 93.69%. The model's stability is confirmed by 5-Fold Cross Validation, achieving an average accuracy of 95.58% with a low standard deviation of 0.46%. Further analysis reveals that applying ADASYN to this dataset reduces performance compared to the original data due to distributional distortion, whereas a slow-learning strategy with a simple tree structure proves to be the most effective. This study contributes an early detection model for Alzheimer's disease that is not only highly accurate and stable, but also efficient in feature utilization.

مستخلص البحث

فطرية، عائشة نور. ٢٠٢٥. **صنيف مرض الزهايمر باستخدام طريقة *XGBoost***. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. الأستاذ الدكتور محمد فيصل،الماجستير في الهندسة الثاني: شوقّين ناهوا أوتاما، الماجستير في الهندسة.

الكلمات الرئيسية : ألزهايمر، XGBoost، ADASYN، التحسين البايزي، اختيار السمات، التحقق المتقاطعK-Fold

رض ألزهايمر هو اضطراب تنكسي عصبي تقدمي يتطلب الكشف المبكر لتعظيم فعالية العلاج الطبي. يهدف هذا البحث إلى بناء نموذج دقيق لتصنيف مرض ألزهايمر باستخدام خوارزمية تعزيز التدرج الأقصى XGBoost وتحليل تأثير تقنيات المعالجة المسبقة للبيانات على أداء النموذج. تتكون مجموعة البيانات المستخدمة من ٢١٤٩ سجلاً للمرضى مع ٣٥ ميزة سريرية. يطبق هذا البحث اختيار الميزات استنادًا إلى ارتباط بيرسون، ومعالجة عدم توازن البيانات باستخدامADASYN ، وتحسين المعلمات الفائقة باستخدام التحسين البايزي .Bayesian Optimization تم إجراء التحقق من صحة النموذج بشكل شامل باستخدام طريقة التحقق المتقاطع بخمس طيات 5-Fold Cross Validation لاختبار اتساق الأداء. أظهرت نتائج البحث أن نموذج XGBoost قادر على توفير أداء تصنيف ممتاز. تم الحصول على أفضل إنجاز في السيناريو ١٠، أي بدون موازنة وبنسبة تقسيم بيانات ٦٠:٤٠ وعتبة اختيار الميزات ٠.١٥، مما أدى إلى دقة اختبار بنسبة ٩٥.٥٨٪، ودقة Precision ٩٤.٦٣٪، واستدعاء Recall ٩٢.٧٦٪، ودرجة F1 بنسبة ٩٣.٦٩٪. تم تأكيد استقرار هذا النموذج من خلال التحقق المتقاطع بخمس طيات الذي سجل متوسط دقة قدره ٩٥.٥٨٪ مع انحراف معياري منخفض يبلغ ٠.٤٦٪. أظهر التحليل المتعمق أن استخدام ADASYN في مجموعة البيانات هذه أدى في الواقع إلى انخفاض أداء النموذج مقارنة بالبيانات الأصلية بسبب تشويه التوزيع، بينما أثبتت استراتيجية التعلم البطيء slow learningذات هيكل الشجرة البسيط أنها الأكثر فعالية. يساهم هذا البحث في توفير نموذج للكشف المبكر عن مرض ألزهايمر الذي لا يتميز بالدقة العالية والاستقرار فحسب، بل يتميز أيضًا بالكفاءة في استخدام الميزات.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Alzheimer merupakan salah satu penyakit neurodegeneratif progresif yang banyak menyerang kelompok lanjut usia di berbagai negara, termasuk Indonesia, dan jumlah penderitanya terus meningkat setiap tahunnya menurut (Pratama, 2025). Gangguan ini terjadi akibat kerusakan sel saraf otak yang mengakibatkan penurunan memori, kesulitan berkomunikasi, serta perubahan perilaku sehingga memengaruhi aktivitas sehari-hari menurut (Goel et al., 2022). Jika tidak segera terdiagnosis dan ditangani secara tepat, Alzheimer dapat memperburuk kualitas hidup penderita, menambah beban psikologis keluarga, serta meningkatkan risiko terjadinya komplikasi medis serius, seperti pneumonia aspirasi, dehidrasi, atau malnutrisi, yang pada akhirnya dapat berujung pada kematian (Alzheimer's Association, 2025).

Penelitian ini berfokus pada pengembangan sistem klasifikasi penyakit Alzheimer menggunakan model *machine learning* XGBoost. Upaya ini selaras dengan nilai yang diajarkan dalam Islam, yaitu pentingnya berikhtiar (berusaha) mencari solusi dan obat untuk setiap penyakit. Dalam surat An-Nahl ayat 69, Allah SWT berfirman:

ثُمَّ كُلِيْ مِنْ كُلِّ الثَّمَرٰتِ فَاسْلُكِيْ سُبُلَ رَبِّكِ ذُلُلًاۗ يَخْرُجُ مِنْۢ بُطُوْنِهَا شَرَابٌ مُّخْتَلِفٌ اَلْوَانُهٗ فِيْهِ شِفَاۤءٌ لِّلنَّاسِۗ اِنَّ فِيْ ذٰلِكَ لَاٰيَةً لِّقَوْمٍ يَّتَفَكَّرُوْنَ

"Kemudian makanlah dari segala (macam) buah-buahan lalu tempuhlah jalan Tuhanmu yang telah dimudahkan (bagimu)." Dari perut lebah itu keluar minuman

(madu) yang bermacam-macam warnanya, di dalamnya terdapat obat yang menyembuhkan bagi manusia. Sungguh, pada yang demikian itu benar-benar terdapat tanda (kebesaran Allah) bagi orang yang berpikir." (QS. An-Nahl : 69).

Ayat ini menegaskan bahwa Allah menghadirkan sarana penyembuhan melalui ciptaan-Nya, salah satunya madu yang memiliki manfaat besar bagi kesehatan. Makna yang terkandung dalam ayat ini menegaskan bahwa manusia diperintahkan untuk menggunakan akal dan ilmu pengetahuan dalam menggali potensi ciptaan Allah demi kemaslahatan, termasuk dalam bidang kesehatan.

Rasulullah SAW juga memberikan peringatan penting tentang nikmat kesehatan dalam sabdanya:

عَنْ ابْنِ عَبَّاسٍ رَضِيَ اللهُ عَنْهُمَا قَالَ: قَالَ النَّبِيُّ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ: نِعْمَتَانِ مَغْبُونٌ فِيهِمَا كَثِيرٌ مِنَ النَّاسِ: الصِّحَّةُ وَالْفَرَاغُ

"Dari Ibnu Abbas ra., ia berkata: Rasulullah ﷺ bersabda: 'Ada dua kenikmatan yang kebanyakan manusia tertipu karenanya, yaitu kesehatan dan waktu luang." (HR. Al-Bukhari, No. 6412).

Hadis ini menegaskan bahwa kesehatan merupakan salah satu nikmat besar yang sering diabaikan oleh manusia. Banyak orang tidak memanfaatkan masa sehatnya untuk berbuat kebaikan dan baru menyadari nilainya ketika telah sakit. Islam memandang bahwa menjaga kesehatan merupakan bagian dari ibadah, karena dengan tubuh dan akal yang sehat manusia dapat menjalankan kewajiban kepada Allah SWT dengan optimal.

Islam mendorong umatnya untuk berikhtiar mencari pengobatan dan solusi terhadap penyakit. Hal ini sebagaimana sabda Rasulullah SAW:

عَنْ أُسَامَةَ بْنِ شَرِيكٍ قَالَ: قَالَ النَّبِيُّ ﷺ: تَدَاوَوْا، فَإِنَّ اللَّهَ عَزَّ وَجَلَّ لَمْ يَضَعْ دَاءً إِلَّا وَضَعَ لَهُ دَوَاءً، غَيْرَ دَاءٍ وَاحِدٍ، الْهَرَمُ

"Dari Usamah bin Syarik, ia berkata: Rasulullah ﷺbersabda: 'Berobatlah kalian, karena sesungguhnya Allah tidak menurunkan penyakit kecuali menurunkan pula obatnya, kecuali satu penyakit, yaitu tua (pikun)." (HR. Abu Dawud, No. 3855; Tirmidzi, No. 2038).

Hadis ini menunjukkan bahwa Islam sangat mendorong umatnya untuk berikhtiar mencari pengobatan dan solusi terhadap penyakit dengan memanfaatkan ilmu pengetahuan dan teknologi. Upaya seperti pengembangan model klasifikasi berbasis *machine learning* untuk deteksi dini penyakit Alzheimer merupakan bentuk nyata dari ikhtiar pencegahan yang sejalan dengan prinsip Islam dalam menjaga kesehatan.dan menghindari mudarat.

Ayat dan hadis ini dapat dimaknai sebagai motivasi untuk terus berupaya mengembangkan ilmu pengetahuan dan teknologi dalam rangka menemukan cara pencegahan maupun terapi. Dengan begitu, menjaga kesehatan otak dan melakukan ikhtiar pengobatan merupakan wujud tanggung jawab manusia dalam merawat amanah tubuh yang telah dianugerahkan oleh Allah SWT (Lumbantobing & Nirwana, 2023).

Deteksi dini Alzheimer menjadi semakin mendesak dengan bertambahnya populasi lanjut usia dan meningkatnya volume data medis yang dihasilkan. Data medis ini sangat kompleks dan beragam, mencakup riwayat klinis, hasil tes kognitif, dan faktor demografis. Dengan volume data yang begitu besar, diagnosis manual yang hanya mengandalkan observasi klinis dapat menjadi subjektif dan memakan waktu (Maulaningrum et al., 2025). Oleh karena itu, sistem otomatis

yang dapat mengidentifikasi pola dan mengklasifikasikan penyakit secara cepat dan efisien sangat dibutuhkan.

Machine learning menawarkan solusi potensial karena mampu memproses dan menganalisis data dalam jumlah besar secara otomatis dan menghasilkan hasil klasifikasi yang akurat (Wardhana et al., 2023). Penerapan *machine learning* pada data medis Alzheimer menghadapi beberapa tantangan besar. Salah satu tantangan utama adalah mengidentifikasi faktor-faktor penentu yang paling signifikan dari tumpukan data klinis. Faktor usia diketahui menjadi salah satu determinan utama, di mana insiden Alzheimer meningkat tajam pada individu berusia di atas 65 tahun (Susanti et al., 2024).

Algoritma *machine learning* telah terbukti efektif dalam mengolah data medis. Di antara metode yang paling sering digunakan adalah XGBoost (*Extreme Gradient Boosting*), yang dikenal handal dalam menangani data besar serta memberikan performa prediksi yang konsisten (Rayadin et al., 2024).

XGBoost merupakan salah satu algoritma *boosting* yang meningkatkan performa prediksi dengan cara menggabungkan banyak model sederhana (*weak learners*) menjadi sebuah model yang lebih kuat (Kusuma et al., 2025). Algoritma ini menggunakan pendekatan berbasis gradien yang memungkinkan penyesuaian bobot secara lebih optimal pada setiap iterasi, sehingga mampu memperbaiki kesalahan dari model sebelumnya (Saputra et al., 2024). Dengan efisiensi komputasi yang tinggi dan akurasi yang baik, *XGBoost* banyak diaplikasikan dalam berbagai bidang, termasuk analisis data medis (Susanto et al., 2025).

XGBoost menjadi pilihan yang sangat tepat untuk tugas klasifikasi dalam penelitian ini. Kemampuan XGBoost untuk menangani berbagai jenis fitur dan mengidentifikasi hubungan non-linier yang kompleks menjadikannya ideal untuk menemukan pola-pola halus dalam data medis yang mungkin terlewat oleh metode lain. Pola-pola inilah yang sangat penting untuk membedakan secara akurat antara pasien sehat dan pasien yang menderita Alzheimer, yang seringkali sulit dilakukan oleh metode klasifikasi yang lebih sederhana. Dengan memanfaatkan kekuatan ini, penelitian ini dapat membangun model yang tidak hanya akurat tetapi juga lebih *robust* dalam menangani kompleksitas data klinis Alzheimer (Nalasari et al., 2025).

Evaluasi model menjadi langkah kunci untuk memastikan model XGBoost dapat mengklasifikasikan Alzheimer dengan akurat. Pengukuran kinerja ini sangat relevan dalam konteks data medis, di mana kesalahan diagnosis memiliki konsekuensi besar (Carbonell & Paulini, 2025). Dalam hal ini, metrik evaluasi seperti akurasi, presisi, *recall*, dan *f1-score* sangat relevan untuk menilai kemampuan model. Evaluasi juga membantu mengidentifikasi potensi *overfitting*, di mana model tampil baik pada data pelatihan namun gagal menggeneralisasi pada data baru (Nguyen et al., 2023).

Penelitian ini juga berfokus pada penyetelan *hyperparameter*, penelitian ini mengadopsi pendekatan optimasi yang lebih cerdas, yaitu *Bayesian Optimization*. *Bayesian Optimization* adalah metode pencarian terinformasi (*informed search*) yang menggunakan hasil dari evaluasi sebelumnya untuk membuat keputusan cerdas tentang kombinasi *hyperparameter* mana yang akan diuji selanjutnya. Metode ini membangun model perkiraan (*surrogate model*) untuk memetakan

hyperparameter ke skor kinerjanya, dan menggunakan *acquisition function* untuk menyeimbangkan antara eksplorasi atau mencoba area baru dan eksploitasi (menyempurnakan area yang sudah bagus).

Bayesian Optimization akan diterapkan untuk menemukan nilai optimal dari lima *hyperparameter* XGBoost yang paling berpengaruh yaitu, *max_depth* untuk mengontrol kompleksitas pohon, *learning_rate* untuk mengatur laju pembelajaran, *n_estimators* untuk menentukan jumlah total pohon, *subsample* untuk mengontrol fraksi sampel data, dan *colsample_bytree* untuk mengontrol fraksi fitur (Sari et al., 2024). Dengan menerapkan metode ini, diharapkan model klasifikasi Alzheimer dapat mencapai tingkat akurasi dan kestabilan yang maksimal dengan proses komputasi yang efisien.

Tantangan umum pada data medis adalah ketidakseimbangan kelas (*imbalanced data*), di mana jumlah pasien sehat mungkin jauh lebih banyak daripada pasien Alzheimer (Zhu et al., 2024). Penelitian ini juga berfokus pada penerapan metode *balancing* data, seperti ADASYN, dan metode *feature selection* berbasis *Pearson* untuk memilih fitur yang paling relevan.

Penelitian ini bertujuan menerapkan model XGBoost pada tugas klasifikasi penyakit Alzheimer. Penelitian sebelumnya, seperti yang dilakukan oleh (Yahya et al., 2025), telah menunjukkan keefektifan XGBoost, namun masih berfokus pada *balancing* data menggunakan SMOTE. Selain itu, dalam hal optimasi model, penelitian tersebut diketahui masih mengandalkan metode konvensional seperti *Grid Search* untuk *hyperparameter tuning*. Kelemahan utama *Grid Search* adalah tidak efisien secara komputasi, karena metode ini menguji setiap kemungkinan

kombinasi parameter secara *brute-force* (menyeluruh), yang memakan banyak waktu dan sumber daya komputasi.

Penelitian ini akan mengombinasikan algoritma XGBoost dengan metode *balancing* ADASYN sebagai alternatif dari SMOTE, dan metode optimasi cerdas *Bayesian Optimization* sebagai pengganti *Grid Search* yang lebih efisien dalam menemukan *hyperparameter* optimal. Penelitian ini juga akan menerapkan *feature selection* berbasis *Pearson* untuk memilih fitur paling relevan. Dengan demikian, diharapkan model yang dikembangkan dapat memberikan solusi deteksi dini Alzheimer yang tidak hanya akurat dan stabil, tetapi juga dapat dicapai melalui proses komputasi yang lebih efisien.

1.2 Rumusan Masalah

Penelitian ini bertujuan untuk mengevaluasi secara mendalam kemampuan metode *Machine Learning* dalam mengklasifikasikan penyakit Alzheimer. Fokus utama diletakkan pada algoritma *XGBoost* yang dikenal memiliki performa tinggi. Oleh karena itu, perlu dirumuskan pertanyaan-pertanyaan kunci yang akan memandu proses penelitian dan analisis, khususnya terkait efektivitas dan faktor-faktor yang memengaruhi performa model dalam konteks diagnosis penyakit ini.

a. Bagaimana performa metode *XGBoost* dalam mengklasifikasikan penyakit Alzheimer?
b. Faktor-faktor apa saja yang mempengaruhi performa metode *XGBoost*?

1.3 Batasan Masalah

Untuk menjaga fokus penelitian, perlu ditetapkan batasan-batasan yang jelas. Batasan ini mencakup spesifikasi sumber data, variabel yang digunakan, serta metode pra-pemrosesan dan optimasi model yang akan diterapkan. Dengan adanya batasan yang jelas dan terperinci, penelitian ini dapat terhindar dari ruang lingkup yang terlalu luas dan hasilnya dapat diinterpretasikan secara spesifik.

a. Data yang digunakan dalam penelitian ini adalah *Alzheimer's Disease Dataset* yang bersumber dari https://www.kaggle.com/. Dataset ini terdiri dari 2.149 *record* data serta 35 atribut yang memuat informasi demografis, gaya hidup, riwayat kesehatan, serta hasil pemeriksaan klinis terkait penyakit Alzheimer.
b. Menggunakan atribut PatientID, *Age, Gender, Ethnicity, EducationLevel, BMI, Smoking, AlcoholConsumption, PhysicalActivity, DietQuality, SleepQuality, FamilyHistoryAlzheimers, CardiovascularDisease, Diabetes, Depression, HeadInjury, Hypertension, SystolicBP, DiastolicBP, CholesterolTotal, CholesterolLDL, CholesterolHDL, CholesterolTriglycerides, MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems, ADL, Confusion, Disorientation, PersonalityChanges, DifficultyCompletingTasks, Forgetfulness*, Diagnosis pada Alzheimer dataset.
c. Penelitian ini menggunakan normalisasi *Min-Max* dan penanganan data tidak seimbang dengan metode ADASYN. Selanjutnya, proses seleksi fitur hanya berfokus pada korelasi *Pearson* untuk menentukan fitur paling berpengaruh,

proses *hyperparameter tuning* model juga dibatasi hanya menggunakan metode *Bayesian Optimization*.

1.4 Tujuan Penelitian

Tujuan penelitian ini dirumuskan berdasarkan rumusan masalah yang ada, berfokus pada evaluasi kinerja model XGBoost. Tujuan ini secara spesifik mencakup pengujian kemampuan model dalam klasifikasi, serta menganalisis kontribusi setiap tahapan pra-pemrosesan dan optimasi. Hal ini penting untuk memastikan model yang dihasilkan tidak hanya akurat secara keseluruhan, tetapi juga efektif dalam mengidentifikasi kasus kelas minoritas.

a. Mengevaluasi kinerja metode XGBoost dalam klasifikasi penyakit Alzheimer berdasarkan metrik evaluasi komprehensif (akurasi, presisi, *recall*, dan *f1-score*).
b. Menerapkan proses normalisasi *min-max* dan seleksi fitur, mengukur pengaruh *balancing data*, serta menerapkan *hyperparameter tuning* terhadap performa model *XGBoost* terutama pada kemampuannya mendeteksi kelas minoritas yaitu pasien Alzheimer.

1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan dampak positif baik bagi pengembangan keilmuan maupun aplikasi praktis di bidang kesehatan. Manfaat ini mencakup kontribusi yang berharga dalam penerapan teknik *machine learning* dan optimasi pada data medis yang tidak seimbang. Selain itu, penelitian ini memberikan panduan praktis mengenai efektivitas algoritma XGBoost dan metode

data balancing ADASYN untuk deteksi dini penyakit Alzheimer, yang dapat menjadi dasar pertimbangan untuk sistem penunjang keputusan klinis di masa mendatang.

a. Memberikan kontribusi dalam pengembangan metode klasifikasi penyakit Alzheimer berbasis data medis.
b. Memberikan pengetahuan mengenai efektivitas metode *XGBoost*.
c. Hasil dari penelitian ini dapat digunakan untuk referensi atau acuan bagi penelitian selanjutnya yang berfokus pada pemanfaatan *machine learning* dalam diagnosis dini Alzheimer.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian yang dilakukan oleh Bernardus Septian Cahya Putra dkk. (Putra et al., 2024) tentang penerapan algoritma *Random Forest*, *XGBoost*, dan *Logistic Regression* untuk mendeteksi penyakit paru-paru. Penelitian ini bertujuan untuk membandingkan performa ketiga algoritma tersebut dalam klasifikasi penyakit paru-paru dengan menggunakan metrik evaluasi akurasi, presisi, *recall*, dan *F1-score*. Hasil penelitian menunjukkan bahwa *XGBoost* memberikan performa terbaik setelah dilakukan *hyperparameter tuning* dengan akurasi sebesar 94,44%, presisi 94,98%, recall 94,44%, dan F1-score 94,41%. Algoritma *Random Forest* juga menghasilkan performa yang sebanding dengan *XGBoost* dengan akurasi dan presisi yang tinggi, sedangkan *Logistic Regression* menunjukkan keterbatasan dalam menangani data yang kompleks dengan performa yang lebih rendah pada seluruh metrik evaluasi. Dari hasil penelitian ini terlihat bahwa performa metode *XGBoost* merupakan metode yang paling baik di antara metode lainnya.

Penelitian yang dilakukan oleh Opitasari dkk. (2024) dengan judul "Klasifikasi Diagnosis untuk Penyakit Kanker Serviks Menggunakan Algoritma *Extreme Gradient Boosting (XGBoost)*" memanfaatkan metode *XGBoost* untuk mengklasifikasikan gejala awal penyakit kanker serviks. Dataset yang digunakan berasal dari *UCI Repository* dengan penerapan teknik *boosting* untuk

meminimalisir nilai *loss function* sehingga model dapat bekerja lebih optimal. Hasil evaluasi menunjukkan bahwa model *XGBoost* mampu mencapai akurasi sebesar 86%, dengan nilai presisi 100%, *recall* 82%, dan *F1-score* 90%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa *XGBoost* sangat baik dalam melakukan klasifikasi penyakit kanker serviks.

Penelitian yang dilakukan oleh R. Soelistijadi dkk. (2024) dengan judul "Pemodelan Prediktif Menggunakan Metode *Ensemble Learning XGBoost* dalam Peningkatan Akurasi Klasifikasi Penyakit Ginjal" bertujuan untuk meningkatkan akurasi dalam mengklasifikasikan pasien Penyakit Ginjal Kronis (PGK). Model *XGBoost* dilatih menggunakan dataset PGK sebanyak 400 record yang dibagi menjadi data latih (70%) dan data uji (30%). Optimasi dilakukan dengan teknik *parameter tuning* menggunakan metode *grid search* pada lima parameter utama, yaitu *n_estimators*, *max_depth*, *learning_rate*, *subsample*, dan *colsample_bytree*. Hasil evaluasi dengan *confusion matrix* menunjukkan bahwa model mencapai akurasi 99,16%, presisi 98,17%, *recall* 99,16%, dan *F1-score* 99,16%. Dari pengujian tersebut menunjukkan bahwa *XGBoost* dengan penerapan *parameter tuning* terbukti sebagai metode klasifikasi yang sangat baik untuk kasus klasifikasi penyakit ginjal.

Penelitian yang dilakukan oleh Murdiansyah (2024) dengan judul "Prediksi Stroke Menggunakan *Extreme Gradient Boosting*" membandingkan performa algoritma *XGBoost* dengan beberapa model *machine learning* lain, seperti *Stacking*, *Random Forest*, dan *Majority Voting*. Data yang digunakan dalam penelitian ini merupakan dataset stroke dengan penerapan teknik SMOTE untuk menangani data

imbalance serta *Bayesian Optimization* untuk optimasi *hyperparameter*. Hasil penelitian menunjukkan bahwa *XGBoost* mampu mencapai akurasi 95,4%, presisi 94,3%, *recall* 96,6%, *F1-score* 95,4%, dan AUC 95,4%. Meskipun demikian, performa *XGBoost* masih belum mampu mengungguli *Stacking* yang mencapai akurasi terbaik sebesar 98%, serta *Random Forest* yang juga menunjukkan hasil lebih tinggi. Penelitian ini juga menekankan bahwa performa *XGBoost* masih berpotensi ditingkatkan melalui optimasi *hyperparameter* yang lebih luas, sehingga di masa mendatang metode ini dapat menghasilkan kinerja yang lebih optimal.

Penelitian yang dilakukan oleh Sitompul dkk. (2025) dengan judul "*Comparison of Xgboost, Random Forest and Logistic Regression Algorithms in Stroke Disease Classification*" mengevaluasi kinerja tiga algoritma *machine learning* dalam mengklasifikasikan penyakit stroke. Dataset yang digunakan terdiri dari 5.110 catatan pasien dengan 12 atribut yang mencakup faktor demografi, gaya hidup, dan kondisi kesehatan. Untuk mengatasi ketidakseimbangan data antara kasus stroke dan non-stroke, digunakan metode *SMOTE*. Hasil evaluasi menunjukkan bahwa algoritma *XGBoost* memberikan performa terbaik dengan akurasi 95%, diikuti *Random Forest* dengan akurasi 94%, sedangkan *Logistic Regression* hanya mencapai 82%. Analisis fitur mengidentifikasi usia, kadar glukosa darah rata-rata, dan riwayat penyakit jantung sebagai prediktor paling berpengaruh dalam diagnosis stroke. Penelitian ini menegaskan bahwa pendekatan *ensemble learning*, khususnya *XGBoost*, lebih unggul untuk mendukung diagnosis dini penyakit stroke.

Penelitian yang dilakukan oleh Sausan dkk. (2025) dengan judul "Perbandingan Metode *Decision Tree Classifier* dan *XGBoost Classifier* dalam Memprediksi Penyakit Jantung" membandingkan efektivitas dua algoritma dalam memprediksi risiko penyakit jantung. Data yang digunakan berasal dari *UCI Machine Learning Repository* dengan total 920 data observasi dan 14 fitur, yang melalui tahap *preprocessing, encoding*, serta *oversampling* sebelum dianalisis. Hasil evaluasi menunjukkan bahwa algoritma *XGBoost* memperoleh akurasi sebesar 93%, lebih tinggi dibandingkan *Decision Tree* yang hanya mencapai 90%. Dengan demikian, *XGBoost* dinilai lebih efektif dalam mendukung diagnosis dini penyakit jantung.

Penelitian yang dilakukan oleh Andryan dkk. (2022) dengan judul "Komparasi Kinerja Algoritma *XGBoost* dan Algoritma *Support Vector Machine (SVM)* untuk Diagnosis Penyakit Kanker Payudara" membandingkan performa dua algoritma dalam klasifikasi kanker payudara dengan menggunakan dataset *Wisconsin Breast Cancer Diagnostic* dari *UCI Machine Learning Repository*. Metode yang digunakan adalah *Knowledge Data Discovery* (KDD) untuk memproses data sebelum dilakukan klasifikasi menjadi *benign* atau *malignant*. Hasil penelitian menunjukkan bahwa *XGBoost* memberikan performa terbaik dengan akurasi 95,12% dan nilai *ROC_AUC* 0,99, sedangkan *SVM* hanya mencapai akurasi 90,24% dengan nilai *ROC_AUC* 0,98. Dengan demikian, *XGBoost* dinilai lebih unggul dalam diagnosis penyakit kanker payudara.

Penulis merangkum hasil temuan dari sejumlah studi yang telah dibahas dalam bentuk perbandingan pada Tabel 2.1. menampilkan rangkuman penelitian

sebelumnya yang memanfaatkan berbagai algoritma dalam klasifikasi penyakit. Tabel ini memuat informasi mengenai judul penelitian, peneliti, metode yang digunakan, serta hasil yang diperoleh. Informasi yang tercantum dalam tabel ini memberikan gambaran tentang efektivitas metode yang digunakan di bidang kesehatan, beserta tingkat akurasi yang dicapai oleh masing-masing penelitian.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
1	Putra et al. (2024)	Penerapan Algoritma *Random Forest*, *XGBoost*, dan *Logistic Regression* untuk Mendeteksi Penyakit Paru-Paru	*Random Forest*, *XGBoost*, *Logistic Regression*	*XGBoost* terbaik dengan akurasi 94,44%, presisi 94,98%, *recall* 94,44%, dan *f1-score* 94,41%. *Random Forest* mendekati *XGBoost, Logistic Regression* lebih rendah.
2	Opitasari et al. (2024)	Klasifikasi Diagnosis untuk Penyakit Kanker Serviks Menggunakan Algoritma XGBoost	XGBoost	Akurasi 86%, presisi 100%, recall 82%, F1-score 90%. XGBoost terbukti efektif dalam klasifikasi kanker serviks
3	Soelistijadi et al. (2024)	Pemodelan Prediktif Menggunakan Metode *Ensemble Learning XGBoost* dalam Peningkatan Akurasi Klasifikasi Penyakit Ginjal	XGBoost dengan *Grid Search* (parameter tuning)	Akurasi 99,16%, presisi 98,17%, recall 99,16%, F1-score 99,16%. XGBoost sangat baik untuk klasifikasi penyakit ginjal.
4	Murdiansyah (2024)	Prediksi Stroke Menggunakan *Extreme Gradient Boosting*	XGBoost, *Stacking, Random Forest*, *Majority Voting*	XGBoost akurasi 95,4%, presisi 94,3%, recall 96,6%, F1-score 95,4%, AUC 95,4%. Namun Stacking unggul dengan akurasi 98%.
5	Sitompul et al. (2025)	*Comparison of XGBoost, Random Forest, and Logistic Regression Algorithms in Stroke Disease Classification*	XGBoost, *Random Forest, Logistic Regression*	XGBoost akurasi 95%, Random Forest 94%, Logistic Regression 82%. Faktor dominan: usia, glukosa darah, riwayat jantung.
6	Sausan et al. (2025)	Perbandingan Metode *Decision Tree Classifier* dan *XGBoost Classifier* dalam Memprediksi Penyakit Jantung	*Decision Tree*, XGBoost	XGBoost akurasi 93%, lebih tinggi daripada Decision Tree 90%.
7	Andryan et al. (2022)	Komparasi Kinerja Algoritma *XGBoost* dan *Support Vector Machine* (SVM) untuk Diagnosis Penyakit Kanker Payudara	XGBoost, SVM	XGBoost akurasi 95,12%, ROC_AUC 0,99. SVM lebih rendah dengan akurasi 90,24% dan ROC_AUC 0,98.

Pada Tabel 2.1, penulis menyajikan rangkuman penelitian sebelumnya yang memanfaatkan berbagai algoritma dalam klasifikasi penyakit. Tabel ini memuat informasi mengenai judul penelitian, peneliti, metode yang digunakan, serta hasil yang diperoleh. Pada tabel tersebut ditampilkan sejumlah penelitian yang menerapkan algoritma XGBoost dan algoritma pembanding lainnya, seperti *Random Forest*, *Logistic Regression*, *Decision Tree*, *Support Vector Machine (SVM)*, *Stacking*, dan *Majority Voting*, untuk mendeteksi dan mengklasifikasikan berbagai jenis penyakit, di antaranya paru-paru, kanker serviks, ginjal, stroke, jantung, dan kanker payudara. Informasi yang tercantum dalam tabel ini memberikan gambaran tentang efektivitas metode yang digunakan di bidang kesehatan, beserta tingkat akurasi yang dicapai oleh masing-masing penelitian.

2.2 Alzheimer

Alzheimer merupakan penyakit kronis yang ditandai dengan penurunan daya ingat dan fungsi kognitif akibat kerusakan sel saraf otak (Nurmawanti & Sudaryanto, 2025). Kondisi ini biasanya disebabkan oleh penumpukan protein abnormal yang mengganggu komunikasi antar sel saraf sehingga fungsi otak menurun. Bagian otak yang paling sering terdampak adalah hippocampus, yang berperan penting dalam menyimpan memori dan orientasi (Digambiro et al., 2024). Apabila tidak ditangani sejak dini, Alzheimer dapat menimbulkan dampak serius pada kualitas hidup penderita, termasuk kesulitan dalam aktivitas sehari-hari, gangguan perilaku, hingga penurunan fungsi tubuh secara menyeluruh (Shabariah et al., 2022).

Jumlah kasus Alzheimer terus meningkat di Indonesia seiring dengan bertambahnya populasi lansia. Berdasarkan studi yang dilakukan di Provinsi Daerah Istimewa Yogyakarta, prevalensi demensia, termasuk Alzheimer, meningkat drastis dari 3,9 per 1.000 orang pada usia 60–64 tahun menjadi 104,8 per 1.000 pada usia di atas 90 tahun, yang menunjukkan tren eksponensial seiring penambahan usia (Suriastini et al., 2020). Selain itu, penelitian lain menyatakan bahwa sekitar 27,9% lansia di Pulau Jawa dan Bali mengalami demensia, dengan lebih dari 4,2 juta penduduk Indonesia yang diperkirakan menderita kondisi ini secara keseluruhan (Bestari, 2023). Dampaknya pun signifikan tidak hanya bagi penderita tetapi juga pengasuh; satu studi di Jawa Barat menemukan bahwa hampir setengah dari *caregiver* demensia melaporkan kualitas hidup yang buruk terutama dalam dimensi fisik, psikologis, dan sosial (Rahmi & Putri, 2021). Lebih jauh, penyakit Alzheimer juga dapat memicu berbagai komplikasi medis, seperti pneumonia aspirasi, infeksi saluran pernapasan dan kemih, dehidrasi, malnutrisi, luka tekan, patah tulang akibat jatuh, serta gangguan jantung (Borders et al., 2020). Selain itu, komplikasi psikiatri seperti depresi, halusinasi, dan perubahan perilaku juga kerap muncul, sehingga memperberat beban pasien maupun keluarganya (Krinitski et al., 2021).

Beberapa faktor risiko utama diketahui turut memengaruhi perkembangan penyakit Alzheimer. Faktor usia merupakan salah satu yang signifikan, di mana risiko Alzheimer meningkat drastis seiring bertambahnya usia (Komala & Nasution, 2025). Selain itu, faktor genetik seperti varian APOE-ε4, yaitu gen yang berperan dalam metabolisme lemak dan sering dikaitkan dengan peningkatan

akumulasi plak beta-amyloid di otak, telah diidentifikasi sebagai faktor risiko penting, terutama ketika dikombinasikan dengan gaya hidup dan faktor lingkungan lainnya (Rumah Sakit Advent, 2024). Kondisi kardiovaskular seperti hipertensi, diabetes, dan kadar kolesterol tinggi juga turut berkontribusi terhadap peningkatan risiko Alzheimer karena berpotensi merusak suplai darah ke otak (Sidarta et al., 2025).

Mengadopsi gaya hidup sehat diyakini dapat menekan risiko terjadinya penyakit Alzheimer. Penelitian menunjukkan bahwa konsumsi makanan bergizi, terutama yang kaya antioksidan seperti buah, sayur, dan ikan berlemak, turut mendukung kesehatan otak dan dapat memperlambat penurunan kognitif. Selain itu, stimulasi mental melalui aktivitas seperti bermain teka-teki atau memasak juga efektif menjaga fungsi kognitif (Susanti et al., 2024). Intervensi edukasi, khususnya melalui program workshop dan pelatihan online, terbukti meningkatkan pengetahuan maupun keterampilan masyarakat tentang deteksi dini dan perawatan demensia Alzheimer (Nurbaiti et al., 2023). Dengan demikian, Langkah pencegahan yang melibatkan kombinasi gaya hidup sehat dan edukasi terpadu dapat menjadi strategi yang menjanjikan dalam mencegah atau menunda timbulnya penyakit Alzheimer.

2.3 *Machine Learning*

Machine Learning merupakan komponen utama dalam kecerdasan buatan yang memungkinkan sistem komputer belajar dari data dan membuat prediksi tanpa perlu diprogram secara langsung. Esensi pembelajaran ini terletak pada kemampuannya mengenali pola dan hubungan dalam data, sehingga

memungkinkan pembuatan keputusan yang lebih akurat di masa yang akan datang. Teknologi ini sangat berguna di berbagai bidang, khususnya di sektor medis, di mana ia mampu mengolah data besar dan kompleks untuk membangun model prediksi yang akurat (Mirafuddin & Supriyatna, 2024). Selain itu, *Machine Learning* juga meningkatkan proses pengambilan keputusan dalam layanan kesehatan, seperti mempermudah diagnosis penyakit, mempercepat pengolahan data medis, dan meningkatkan akurasi hasil deteksi (Idris et al., 2025). Dengan demikian, penerapannya di bidang medis menjanjikan efisiensi dan peningkatan mutu layanan kesehatan berbasis data (Wardhana et al., 2023).

Pemanfaatan *Machine Learning* terbukti mampu membantu dalam mengidentifikasi faktor risiko penyakit yang sulit dikenali melalui pendekatan analisis tradisional (Ginting et al., 2022). Teknologi ini memiliki kemampuan dalam mengolah data berukuran besar dengan karakteristik yang kompleks, sehingga mampu menyingkap pola tersembunyi yang tidak mudah diperoleh dengan metode klasik (Natsir et al., 2024). Oleh karena itu, *Machine Learning* dinilai berpotensi mendukung pencegahan dan penanganan penyakit secara lebih efektif. Sebagai contoh, penerapan algoritma klasifikasi dapat digunakan untuk menganalisis data medis dalam skala luas sehingga dapat mengungkap faktor risiko penyakit kronis dengan tingkat ketepatan yang lebih baik (Tarimana et al., 2024).

Algoritma *Machine Learning* telah memberikan kontribusi besar dalam mendukung proses diagnosis maupun prediksi penyakit. Beberapa algoritma yang sering digunakan antara lain regresi, *Decision Tree*, serta jaringan saraf tiruan. Masing-masing algoritma tersebut memiliki karakteristik yang berbeda. Regresi,

misalnya, lebih sesuai diterapkan pada data numerik untuk tujuan prediksi atau identifikasi hubungan antarvariabel. Sementara itu, *Decision Tree* kerap dipilih karena mampu memberikan interpretasi yang jelas terhadap proses pengambilan keputusan, sehingga lebih mudah dipahami oleh praktisi medis. Di sisi lain, jaringan saraf tiruan memiliki keunggulan dalam menganalisis data dengan dimensi yang tinggi, khususnya citra medis, meskipun memerlukan komputasi yang lebih besar (Moreno-Ibarra et al., 2021).

2.4 *XGBoost Classifier*

Extreme Gradient Boosting (XGBoost) adalah algoritma *ensemble learning* yang sangat efisien dan skalabel, yang didasarkan pada kerangka kerja *Gradient Boosting Machine* (GBM). Algoritma ini dirancang untuk mengatasi kelemahan GBM tradisional, terutama dalam hal kecepatan komputasi dan pengendalian *overfitting*. XGBoost pertama kali diperkenalkan oleh Chen dan Guestrin (2016), dan sejak itu menjadi salah satu model dominan untuk data tabular dalam kompetisi *machine learning* dan aplikasi industri (Arif Ali et al., 2023). Cara kerjanya adalah dengan membangun model secara sekuensial (aditif), di mana setiap model baru berupa *decision tree* dilatih untuk memperbaiki kesalahan atau *residual* dari gabungan model-model sebelumnya. Keunggulan utamanya terletak pada efisiensi pemrosesan data, kemampuannya menangani data yang hilang (*missing values*), dan mekanisme regularisasi internal yang kuat untuk mencegah *overfitting* (Darmawan et al., 2024).

XGBoost terdiri dari kumpulan *decision trees* (pohon keputusan) yang dikenal sebagai *weak learners*. Berbeda dengan metode *bagging* seperti *Random*

Forest yang membangun pohon secara paralel, XGBoost membangun pohon secara aditif. Model akhir merupakan penjumlahan dari prediksi semua pohon (Yu et al., 2025). Pada setiap langkah iterasi (t), model menambahkan pohon baru f_t yang secara spesifik dilatih untuk memprediksi *residual* (selisih antara nilai aktual dan prediksi) dari gabungan pohon hingga iterasi ke *t*–1. Kemampuan untuk mengoptimalkan *loss function* secara langsung menggunakan turunan kedua (Hessian) dari *loss function* inilah yang membedakannya dari *gradient boosting* tradisional, memungkinkan konvergensi yang lebih cepat dan akurat (Zhou & Xie, 2025).

Konsep utama XGBoost terletak pada fungsi objektif yang dioptimalkannya pada setiap langkah. Fungsi ini dirancang untuk menyeimbangkan dua aspek krusial yaitu, *Loss Function* (fungsi kerugian) dan *Regularization Term* (fungsi regularisasi). *Loss function* (*l*) mengukur seberapa baik model memprediksi data (misalnya, *log loss* untuk klasifikasi), sementara *regularization term* (**Ω**) menghukum kompleksitas model (misalnya, jumlah daun atau besarnya bobot pada daun) untuk mencegah *overfitting* (Iriananda et al., 2025).

Proses pelatihan XGBoost secara matematis adalah upaya untuk menemukan pohon baru f_t yang meminimalkan fungsi objektif ini. Penjelasan berikut memaparkan fungsi dan formulasi matematis utama yang digunakan dalam XGBoost yang terbagi menjadi dua bagian (Yulianti et al., 2022):

A. Proses Pelatihan

1. Model Prediksi Aditif

Prediksi akhir dari model XGBoost untuk sebuah data x_i adalah hasil penjumlahan dari prediksi setiap pohon K (jumlah total pohon) yang telah dibangun. Model ini dibangun secara aditif, di mana prediksi pada iterasi ke-t $\hat{y}_i^{(t)}$ adalah prediksi dari iterasi sebelumnya $\hat{y}_i^{(t-1)}$ ditambah dengan prediksi dari pohon baru ($f_t(x_i)$) disebutkan pada Persamaan 2.1 dan Persamaan 2.2:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i) \tag{2.1}$$

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{2.2}$$

Keterangan :
$\widehat{y_i}$: Prediksi akhir untuk data ke-i.
K : Jumlah total pohon.
f_k : Fungsi yang merepresentasikan pohon keputusan ke-k.
x_i : Vektor fitur data ke-i.
$\hat{y}_i^{(t)}$: Prediksi pada iterasi ke-t.
$f_t(x_i)$: Prediksi dari pohon baru yang ditambahkan pada iterasi ke-t.

2. Fungsi Objektif

XGBoost meminimalkan Fungsi Objektif ($Obj^{(t)}$) untuk menemukan pohon f_t terbaik pada setiap iterasi. Fungsi ini memanfaatkan ekspansi *Taylor orde* kedua (menggunakan *gradien* dan *hessian*) untuk menyederhanakan optimasi *loss function* umum disebutkan pada Persamaan 2.3:

$$\mathrm{Obj}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.3}$$

Keterangan :
$Obj^{(t)}$: Fungsi objektif pada iterasi ke-t.
N : Jumlah data training.
g_i : Gradien (turunan pertama) dari *loss function* terhadap prediksi sebelumnya $\hat{y}_i^{(t-1)}$.

h_i : Hessian (turunan kedua) dari *loss function* terhadap prediksi sebelumnya $\hat{y}_i^{(t-1)}$.
$\Omega(f_t)$: Fungsi regularisasi untuk pohon f_t.

Nilai gradien $g_i$dan hessian $h_i$merupakan statistik penting yang dihitung berdasarkan kesalahan prediksi pada iterasi sebelumnya. Kedua nilai ini berfungsi sebagai dasar untuk mengukur seberapa besar kontribusi sebuah pohon baru dalam mengurangi error keseluruhan. Dengan memanfaatkan dua statistik tersebut, XGBoost dapat menilai perubahan loss secara lebih akurat dibanding metode boosting konvensional yang hanya menggunakan turunan pertama. Selain itu, penggunaan regularisasi melalui $\Omega(f_t)$ membantu mengontrol kompleksitas struktur pohon sehingga model tetap stabil dan terhindar dari overfitting meskipun dilakukan secara iteratif.

3. Regularisasi

Fungsi regularisasi Ω merupakan komponen penting dalam XGBoost yang berperan untuk mengontrol kompleksitas model dan mengurangi risiko overfitting. Tidak seperti algoritma boosting tradisional yang cenderung menghasilkan model sangat kompleks, XGBoost menambahkan penalti terhadap struktur pohon agar pembelajaran berjalan lebih stabil dan terukur. Regularisasi ini diterapkan dengan menambahkan biaya pada jumlah daun dalam pohon serta pada besar kecilnya bobot setiap daun sehingga pohon yang terlalu rumit atau memiliki bobot ekstrem akan dihukum lebih besar. Mekanisme ini memungkinkan model menjaga keseimbangan antara kemampuan mempelajari pola data dan kemampuan untuk melakukan

generalisasi terhadap data baru. Secara matematis, fungsi regularisasi tersebut dirumuskan pada Persamaan 2.4:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (2.4)$$

Keterangan :
$\Omega(f_t)$: Fungsi regularisasi untuk pohon f_t.
T : Jumlah daun (*leaves*) pada pohon.
γ (gamma) : Parameter penalti untuk kompleksitas pohon (penalti per daun).
w_j : Skor atau bobot (*weight*) pada daun ke-j.
λ (lambda) : Parameter regularisasi L2 pada bobot daun.

Parameter γ(gamma) mengendalikan keputusan apakah sebuah pemisahan node layak dilakukan; semakin tinggi nilainya, semakin sulit pohon untuk memperluas dirinya, sehingga hanya pemisahan yang benar-benar memberikan penurunan loss signifikan yang akan dipertahankan. Sementara itu, parameter λ berfungsi menstabilkan bobot daun dengan cara mencegah bobot yang terlalu besar, sehingga model menjadi lebih halus dan tidak sensitif terhadap noise pada data. Kedua parameter ini bekerja sama untuk menjaga struktur pohon tetap sederhana namun tetap informatif, memastikan model belajar secara efektif tanpa kehilangan kemampuan generalisasi.

Semakin besar nilai γdan λ, model akan menjadi semakin konservatif dan cenderung menghasilkan pohon yang lebih kecil serta bobot yang lebih terkontrol. Konsekuensinya, risiko overfitting dapat ditekan secara signifikan, terutama saat bekerja dengan data berukuran kecil atau data yang memiliki variasi tinggi. Dengan demikian, keberadaan regularisasi dalam XGBoost bukan sekadar tambahan, tetapi merupakan bagian fundamental yang

membedakan algoritma ini dari teknik boosting lainnya serta menjadikannya sangat kuat untuk berbagai skenario klasifikasi.

4. Penentuan Pemisahan (*Split Finding Gain*)

Tahap penting dalam pembentukan pohon keputusan pada XGBoost adalah penentuan pemisahan (*split finding*) yang optimal. Keputusan untuk memecah sebuah *node* didasarkan pada perhitungan nilai *Gain* (keuntungan informasi). Secara sistematis, XGBoost akan mengevaluasi setiap potensi pemisahan dan hanya mengeksekusinya jika menghasilkan Gain positif. Perhitungan ini mengandalkan akumulasi nilai gradien (g_i) dan hessian (h_i) dari *node* induk terhadap calon *node* anak (kiri dan kanan), sebagaimana dirumuskan dalam Persamaan 2.5:

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \qquad (2.5)$$

Keterangan :
Gain : Keuntungan informasi dari melakukan pemisahan.
I : Kumpulan data pada *node* induk (sebelum *split*).
I_L, I_R : Kumpulan data pada *node* anak kiri dan kanan (setelah *split*).
g_i , h_i : *Gradien* dan *Hessian* untuk data ke-i.
λ , γ : Parameter regularisasi.

Formula di atas menunjukkan bahwa *Gain* diperoleh dari selisih antara total skor kesamaan (*similarity score*) pada *node* anak dengan skor kesamaan pada *node* induk. Parameter γ di sini berperan sebagai ambang batas minimal; jika Gain maksimal yang ditemukan tidak melampaui nilai γ, maka proses pemisahan dihentikan. Mekanisme inilah yang secara otomatis menjalankan fungsi pemangkasan (*pruning*) untuk mencegah model menjadi terlalu kompleks (*overfitting*).

5. Perhitungan Bobot Daun (*Leaf Weight*)

Penentuan skor mentah (*raw score*) atau bobot (w_j^*) yang akan ditempatkan pada setiap daun baru (j) adalah langkah selanjutnya setelah pemisahan terbia ditemukan menggunakan skor *Gain*. Bobot ini adalah nilai optimal yang meminimalkan fungsi objektif untuk data yang berada di daun tersebut. Perhitungan bobot daun ditunjukkan pada Persamaan 2.6:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{2.6}$$

Keterangan :
w_j^* : Bobot (skor) optimal untuk daun ke-j.
I_j : Himpunan data (indeks) yang berada di daun ke-j.
Σ_{g_i} : Jumlah total gradien dari semua data di daun ke-j.
Σ_{h_i} : Jumlah total hessian dari semua data di daun ke-j.
λ : Parameter regularisasi L2.

Nilai w_j^* inilah yang menjadi keluaran $f_t(x_i)$dari pohon tersebut, yang kemudian ditambahkan ke prediksi model yang terdapat pada Persamaan 2.2. Proses ini bekerja secara berurutan. Model dimulai dengan prediksi awal, kemudian pada setiap iterasi, nilai gradien (g_i) dan hessian (h_i)dihitung berdasarkan *error* dari prediksi model saat ini. Selanjutnya, algoritma membangun pohon baru f_t dengan terlebih dahulu mencari struktur pohon (pemisahan *node*) terbaik dengan memaksimalkan formula *Gain* (Persamaan 2.5). Setelah *split* terbaik ditentukan, algoritma menghitung bobot optimal (w_j^*)untuk setiap daun (*leaf*) baru menggunakan rumus Bobot Daun (Persamaan 2.6). Pohon baru ini yang pada dasarnya adalah kumpulan nilai bobot daun, kemudian ditambahkan ke model untuk memperbarui prediksi. Mekanisme regularisasi (λ dan γ), yang digunakan baik dalam perhitungan *Gain* maupun

Bobot Daun, secara otomatis mengontrol kompleksitas. Proses ini diulang hingga jumlah pohon yang ditentukan tercapai, menjadikan XGBoost sebagai model yang sangat akurat dan tangguh terhadap *overfitting* (Sihombing et al., 2025).

Penjelasan Model Aditif, Fungsi Objektif, Regularisasi, *Split Finding Gain*, dan Perhitungan Bobot Daun adalah proses pelatihan (*training*), yaitu cara model XGBoost dibangun menggunakan data latih. Langkah selanjutnya adalah proses klasifikasi (*inference*), yaitu cara menggunakan model yang sudah jadi untuk mengklasifikasikan data baru yang belum pernah dilihatnya (Kurniawan et al., 2025). Untuk klasifikasi biner (dua kelas), berikut adalah langkah-langkah rinci setelah model selesai dilatih:

B. Proses Klasifikasi (*Inference*) untuk Klasifikasi Biner

Proses klasifikasi menjelaskan bagaimana model yang telah dilatih (terdiri dari *K* pohon), digunakan untuk mengklasifikasikan data baru. Proses ini terdiri dari tiga tahap: agregasi skor mentah, transformasi probabilitas, dan penentuan kelas akhir.

1. Agregasi Skor Mentah (*Raw Score Aggregation*)

Satu data baru (x_i) ke dalam model XGBoost yang sudah terlatih akan dijalankan melewati semua pohon (K) yang telah dibangun selama proses pelatihan. Setiap pohon (f_k) akan memberikan prediksi skor mentah. Skor ini adalah nilai bobot (w) yang ada di *leaf* (daun) tempat data x_i tersebut berakhir. Sesuai dengan model aditif, output akhir dari model adalah penjumlahan dari

skor mentah dari setiap pohon. Skor gabungan ini disebut skor mentah (*raw score*) atau *logit* ($\widehat{y_i}$). Untuk persamaannya terdapat pada Persamaan 2.7:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i) \tag{2.7}$$

Keterangan :
Nilai $\widehat{y_i}$ ini bukan probabilitas. Nilainya bisa positif (misal: 2.5), negatif (misal: -1.3), atau nol. Nilai ini merepresentasikan *log-odds* dari data tersebut untuk masuk ke Kelas 1.

2. Transformasi Probabilitas (Fungsi Sigmoid)

Loss function yang dioptimalkan pada Fungsi Objektif untuk klasifikasi biner biasanya adalah *Log Loss* atau *Binary Cross-Entropy*. Untuk mengubah skor mentah/logit ($\widehat{y_i}$) tadi menjadi probabilitas (nilai antara 0 dan 1) yang dapat diinterpretasikan, jadi harus melewatkannya ke Fungsi Sigmoid (atau Fungsi Logistik). Probabilitas data x_i untuk masuk ke Kelas 1 (p_i) dihitung menggunakan Persamaan 2.8:

$$\mathrm{p_i} = \sigma(\widehat{\mathrm{y_1}}) = \frac{1}{1 + \mathrm{e}^{-\widehat{y_1}}} \tag{2.8}$$

Keterangan :
p_i : Probabilitas data x_i termasuk dalam Kelas 1 (misal: 'Alzheimer').
σ : Fungsi Sigmoid.
$\widehat{y_i}$: Skor mentah (logit) dari langkah agregasi skor mentah.

Contoh:

a. Jika skor mentah $\widehat{y_i}$ = 0, maka p_i = 1 / (1 + e^0) = 1 / (1 + 1) = 0.5. (Probabilitas 50%).
b. Jika skor mentah $\widehat{y_i}$ = 1.5 (positif kuat), maka p_i = 1 / (1 + $e^{-1.5}$) ≈ 0.817. (Probabilitas 81.7%).
c. Jika skor mentah $\widehat{y_i}$ = -2.0 (negatif kuat), maka p_i = 1 / (1 + $e^{-(-2.0)}$) ≈ 0.119. (Probabilitas 11.9%).

Output dari langkah ini adalah probabilitas prediksi.

3. Penentuan Kelas (*Decision Threshold*)

Perubahan probabilitas (p_i) dari langkah transformasi probabilitas menjadi keputusan kelas (Kelas 0 atau Kelas 1) adalah langkah terakhir. Biasanya disebut prediksi kelas akhir sebagai $\widehat{y_i}$. Transformasi ini dilakukan dengan menerapkan fungsi keputusan (*decision function*) yang didasarkan pada nilai ambang batas (*threshold*), yang disimbolkan dengan τ (tau). Dalam klasifikasi biner standar, nilai ambang batas yang umum digunakan adalah τ = 0.5. Secara matematis, prediksi kelas akhir $\widehat{y_i}$ditentukan oleh Persamaan 2.9 *piecewise* berikut:

$$\widehat{y_i} = \begin{cases} 1 \; jika \; p_i \; > \; \tau \\ 0 \; jika \; p_i \; \leq \; \tau \end{cases} \tag{2.9}$$

Dengan $\tau \; = \; 0.5$, maka:

$$\widehat{y_i} = \begin{cases} 1 \; jika \; p_i \; > \; 0.5 \\ 0 \; jika \; p_i \; \leq \; 0.5 \end{cases}$$

Keterangan :
$\widehat{y_i}$: Prediksi kelas akhir untuk data ke-i (1=Alzheimer, 0=Tidak Alzheimer).
p_i : Probabilitas data x_i termasuk dalam Kelas 1 (misal: 'Alzheimer').
τ : Nilai ambang batas keputusan (*threshold*), umumnya 0.5.

Output akhir yang dihasilkan oleh *XGBoost Classifier* berupa prediksi kelas biner, yaitu 0 atau 1, yang merepresentasikan kategori “Tidak Alzheimer” dan “Alzheimer”. Penentuan kelas ini dilakukan dengan membandingkan nilai probabilitas yang dihasilkan oleh fungsi Sigmoid pada tahap sebelumnya dengan ambang batas standar, yaitu 0.5. Jika probabilitas tersebut melebihi

nilai ambang, maka model akan memberikan label kelas 1, sedangkan jika nilainya berada di bawah ambang, maka model mengklasifikasikan data sebagai kelas 0. Mekanisme ini memastikan bahwa keputusan akhir model bersifat konsisten dan mengikuti prinsip dasar klasifikasi biner yang umum digunakan dalam berbagai penelitian *machine learning*.

Alur kerja algoritma *XGBoost Classifier* mulai dari proses pelatihan (*training*) hingga menghasilkan keputusan klasifikasi akhir (*inference*), telah dijelaskan. Untuk dapat menerapkan model tersebut secara efektif dalam penelitian ini, diperlukan sebuah kerangka kerja metodologi yang terstruktur. Kerangka kerja ini mencakup keseluruhan proses, mulai dari akuisisi data mentah, persiapan data (*preprocessing*), hingga implementasi dan evaluasi model. Adapun langkah-langkah implementasi yang dilakukan dalam penelitian ini adalah sebagai berikut (Sausan et al., 2025):

1. Pengumpulan Data

   Pengumpulan data merupakan tahap awal dalam penelitian, yaitu memperoleh data yang akan digunakan untuk proses analisis. Pada penelitian ini, data yang digunakan bersumber dari dataset sekunder berjudul *"Alzheimer's Disease Dataset"* yang tersedia secara terbuka di platform *Kaggle*.

2. *Preprocessing* Data

   Tahap *preprocessing* bertujuan untuk menyiapkan data agar siap diproses oleh algoritma machine learning. Data mentah sering kali mengandung kesalahan, ketidakkonsistenan, atau ketidakseimbangan distribusi

sehingga perlu dibersihkan dan diolah terlebih dahulu (Putra & Nugroho, 2022). Adapun langkah-langkah utama *preprocessing* antara lain:

a. Data *Cleaning*

Data *cleaning* adalah proses pembersihan data dari nilai yang hilang, atribut tidak relevan, atau duplikasi. Tujuannya agar dataset lebih rapi, konsisten, dan dapat meningkatkan kualitas hasil analisis (Rokhman et al., 2020).

b. Normalisasi Data

Normalisasi data adalah proses penyamaan skala antar fitur sehingga semua fitur memiliki kontribusi yang seimbang dalam perhitungan. Normalisasi penting untuk mencegah fitur bernilai besar mendominasi proses pelatihan model (Allorerung et al., 2024).

3. *Feature Selection*

Feature selection adalah proses memilih subset fitur / variabel yang paling relevan terhadap variabel target, dengan tujuan mengurangi fitur yang redundan atau tidak informatif sehingga model mesin-pelajarannya menjadi lebih efisien, lebih cepat dilatih, dan berpotensi memiliki performa yang lebih baik (Leni et al., 2023).

4. *Split* Data

Split data adalah proses membagi dataset menjadi dua bagian utama, yaitu data pelatihan (*training set*) dan data pengujian (*testing set*). Hal ini bertujuan agar model dapat belajar dari sebagian data, lalu diuji pada data lain yang belum pernah dilihat (Oktafiani et al., 2023).

5. *Balancing* Data

   Balancing data adalah upaya untuk mengatasi permasalahan ketidakseimbangan kelas pada dataset, di mana jumlah data antar kelas berbeda jauh. Ketidakseimbangan ini dapat menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga perlu dilakukan balancing agar distribusi data lebih seimbang dan proses pembelajaran model menjadi lebih optimal (Budi Utomo et al., 2024).

6. *Hyperparameter Tuning*

   Hyperparameter Tuning adalah proses penting untuk menyempurnakan dan mengoptimalkan kinerja model *machine learning*. Berbeda dengan *parameter* seperti bobot yang nilainya dipelajari oleh model selama proses pelatihan, *hyperparameter* adalah pengaturan yang ditetapkan oleh peneliti *sebelum* proses pelatihan dimulai (A Ilemobayo et al., 2024).

7. Implementasi Metode

   Pada tahap ini dilakukan penerapan algoritma XGBoost untuk proses klasifikasi penyakit Alzheimer. Data yang telah melalui tahap *preprocessing* selanjutnya digunakan pada proses pelatihan (*training*) untuk membangun model. Algoritma XGBoost dipilih karena memiliki kemampuan dalam menangani data dengan jumlah fitur yang banyak serta dikenal unggul dalam menghasilkan prediksi dengan tingkat akurasi tinggi. Setelah model dilatih, langkah berikutnya adalah melakukan pengujian (*testing*) untuk mengetahui kemampuan model dalam mengenali pola pada data baru yang sebelumnya tidak pernah dilihat.

8. Evaluasi Sistem

   Evaluasi dilakukan untuk menilai kinerja model XGBoost yang telah dibangun. Pada tahap ini, digunakan beberapa metrik evaluasi seperti akurasi, presisi, *recall*, dan f1-*score* untuk mengukur seberapa baik model dapat mengklasifikasikan data dengan benar. Evaluasi ini bertujuan untuk memastikan bahwa model tidak hanya bekerja dengan baik pada data pelatihan, tetapi juga mampu melakukan generalisasi terhadap data uji (Daffaa et al., 2025). Hasil evaluasi selanjutnya dijadikan dasar untuk menarik kesimpulan terkait efektivitas metode yang digunakan dalam mendeteksi penyakit Alzheimer.

2.5 *Bayesian Optimization*

Bayesian optimization merupakan metode optimasi berbasis model probabilistik yang dirancang untuk mencari nilai optimum dari suatu fungsi objektif yang sulit dihitung secara langsung (Shahriari et al., 2016). Berbeda dengan pendekatan pencarian tradisional, metode ini tidak melakukan evaluasi parameter secara menyeluruh, melainkan memanfaatkan informasi dari evaluasi sebelumnya untuk mengarahkan proses pencarian (Frazier, 2018). Pendekatan ini membuat *bayesian optimization* mampu bekerja secara efisien pada fungsi objektif yang membutuhkan biaya komputasi tinggi untuk dievaluasi, memiliki jumlah parameter yang tidak terlalu sedikit maupun terlalu banyak, serta tidak memiliki bentuk persamaan yang dapat dianalisis secara langsung.

Dalam prosesnya, *bayesian optimization* menggunakan *surrogate model* untuk memprediksi bentuk fungsi objektif berdasarkan data evaluasi sebelumnya.

Model penduga yang sering digunakan adalah *gaussian process* karena mampu memberikan estimasi nilai fungsi sekaligus tingkat ketidakpastian prediksinya. Prediksi tersebut kemudian digunakan oleh *acquisition function* untuk menentukan titik parameter yang akan diuji pada iterasi berikutnya. Fungsi ini secara eksplisit menyeimbangkan eksplorasi wilayah parameter yang belum pernah diuji dengan eksploitasi nilai parameter yang telah menunjukkan hasil baik, sehingga proses optimasi dapat berlangsung secara adaptif dan efisien (Rainforth et al., 2017).

Gaussian process merupakan model probabilistik yang banyak digunakan sebagai *surrogate model* dalam *bayesian optimization* karena mampu memetakan hubungan antara kombinasi parameter dan nilai fungsi objektif secara fleksibel tanpa memerlukan bentuk persamaan eksplisit. Model ini tidak hanya memberikan estimasi terhadap nilai fungsi, tetapi juga mengukur tingkat ketidakpastian dari prediksi tersebut, sehingga informasi yang dihasilkan lebih kaya dibandingkan pendekatan regresi konvensional (Elshawi et al., 2019). Kemampuan untuk memodelkan ketidakpastian ini menjadi alasan utama penggunaan *gaussian process*, karena informasi tersebut memungkinkan algoritma *bayesian optimization* menyeimbangkan proses eksplorasi pada wilayah parameter yang belum banyak diketahui dengan eksploitasi pada area yang telah menunjukkan hasil baik (Bischl et al., 2023). Dengan demikian, *gaussian process* membantu meningkatkan efektivitas pencarian nilai parameter yang optimal, terutama pada fungsi yang tidak dapat dirumuskan secara langsung dan memiliki pola yang sulit diprediksi secara sederhana.

Efisiensi *bayesian optimization* dibandingkan metode pencarian konvensional seperti *grid search* dan *random search* disebabkan oleh kemampuannya memanfaatkan informasi dari evaluasi sebelumnya secara sistematis. Pada *grid search*, seluruh kombinasi parameter harus diuji tanpa mempertimbangkan performa kombinasi lain, sehingga jumlah evaluasi meningkat secara drastis seiring bertambahnya dimensi parameter. Sementara itu, *random search* memilih titik secara acak tanpa arah pencarian yang jelas, membuat performa sangat bergantung pada keberuntungan *sampling*. Berbeda dari kedua metode tersebut, *bayesian optimization* membangun model penduga yang diperbarui secara iteratif sehingga proses pencarian dapat difokuskan pada area parameter yang paling berpotensi memberikan nilai optimum. Pendekatan berbasis model ini terbukti mampu mengurangi jumlah evaluasi secara signifikan dan mencapai performa yang lebih baik dalam jumlah iterasi yang lebih sedikit (Snoek et al., 2015).

Keunggulan metode ini juga telah dibuktikan secara nyata dalam berbagai penelitian. Sebagai contoh, penelitian oleh Awal dkk. (2021)menerapkan *bayesian optimization* untuk meningkatkan akurasi deteksi COVID-19 pada data fasilitas rawat inap. Hasilnya menunjukkan bahwa metode ini mampu menghasilkan model diagnosis yang mencapai akurasi hingga 97,5%, sehingga metode ini dapat diandalkan untuk pengambilan keputusan medis yang cepat. Penelitian lainnya ang dilakukan oleh Turner dkk. (2021) menganalisis hasil dari kompetisi *black-box optimization challenge* di NeurIPS 2020, sebuah kompetisi yang menguji berbagai algoritma optimasi pada masalah yang struktur datanya tidak diketahui (*black-box*).

Temuan mereka menyimpulkan bahwa *bayesian optimization* secara konsisten mengalahkan *random search*, terutama karena kemampuannya untuk belajar dan beradaptasi tanpa campur tangan manusia. Hal ini menjadikannya pilihan yang jauh lebih efektif untuk penyetelan *hyperparameter* pada model *machine learning* yang rumit.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Data yang dipakai dalam penelitian ini bersumber dari data sekunder. Dataset yang digunakan berjudul "*Alzheimer's Disease Dataset*" dan dapat diakses secara bebas melalui https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset/data. Dataset tersebut berisi 2.149 sampel pasien dengan total 35 atribut. Dari keseluruhan atribut, penelitian ini hanya memanfaatkan 32 fitur, di mana variabel output yang digunakan adalah Diagnosis, sementara kolom *PatientID* dan dikecualikan karena tidak memiliki pengaruh terhadap proses klasifikasi, sedangkan kolom *DoctorInCharge* dihapus karena kolom tersebut berisi informasi rahasia tentang dokter yang bertanggung jawab terhadap pasien.

Fitur-fitur yang terdapat dalam dataset memuat beragam informasi penting yang dibutuhkan untuk menganalisis risiko terjadinya penyakit Alzheimer. Setiap fitur mewakili karakteristik tertentu yang relevan, mulai dari aspek demografis, hasil pemeriksaan klinis, hingga parameter biologis yang dapat memengaruhi proses terjadinya penyakit. Seluruh fitur tersebut disajikan secara sistematis pada Tabel 3.1. untuk memberikan gambaran menyeluruh mengenai variabel apa saja yang digunakan dalam penelitian ini.

Tabel 3.1 Penjelasan Fitur

No	Fitur	Keterangan
1.	*PatientID*	Identifikasi unik untuk setiap pasien (4751-6900)
2.	*Age*	Usia pasien berkisar antara 60-90 (tahun)
3.	*Gender*	Jenis kelamin pasien (0 = laki-laki, 1 = Perempuan)
4.	*Ethnicity*	Etnis pasien (0 = kaukasia, 1 = afrika amerika, 2 = asia, 3 = lainnya)

No	Fitur	Keterangan
5.	*EducationLevel*	Tingkat pendidikan pasien (0 = tidak ada, 1 = sekolah menengah atas, 2 = sarjana, 3 = lebih tinggi)
6.	*BMI*	Indeks massa tubuh pasien, berkisar antara 15 – 40
7.	*Smoking*	Status merokok (0 = tidak, 1 = ya)
8.	*AlcoholConsumption*	Konsumsi alkohol mingguan dalam satuan berkisar dari 0 – 20
9.	*PhysicalActivity*	Aktivitas fisik mingguan dalam jam, berkisar dari 0 – 10
10.	*DietQuality*	Skor kualitas diet, berkisar dari 0 – 10
11.	*SleepQuality*	Skor kualitas tidur, berkisar dari 4 – 10
12.	*FamilyHistoryAlzheimers*	Riwayat keluarga penyakit Alzheimer, (0 = tidak, 1 = ya)
13.	*CardiovascularDisease*	Adanya penyakit kardiovaskular (0 = tidak, 1 = ya)
14.	*Diabetes*	Adanya penyakit diabetes (0 = tidak, 1 = ya)
15.	*Depression*	Adanya depresi (0 = tidak, 1 = ya)
16.	*HeadInjury*	Riwayat cedera kepala (0 = tidak, 1 = ya)
17.	*Hypertension*	Adanya hipertensi (0 = tidak, 1 = ya)
18.	*SystolicBP*	Tekanan darah sistolik, berkisar antara 90 – 180 mmHG
19.	*DiastolicBP*	Tekanan darah diastolic, berkisar antara 60 – 120 mmHG
20.	*CholesterolTotal*	Kadar kolesterol total, berkisar antara 150 – 300 mg/dL
21.	*CholesterolLDL*	Kadar kolesterol lipoprotein densitas rendah, berkisar antara 50 – 200 mg/dL
22.	*CholesterolHDL*	Kadar kolesterol lipoprotein densitas tinggi, berkisar antara 20 – 100 mg/dL
23.	*CholesterolTriglycerides*	Kadar trigliserida berkisar antara 50 – 400 mg/dL
24.	*MMSE*	Skor Mini-Mental State Examination, berkisar antara 0 – 30. (Skor yang lebih rendah menunjukkan gangguan kognitif)
25.	*FunctionalAssessment*	Skor penilaian fungsional, berkisar antara 0 – 10. (Skor yang lebih rendah menunjukkan gangguan yang lebih besar
26.	*MemoryComplaints*	Adanya keluhan memori (0 = tidak, 1 = ya)
27.	*BehavioralProblems*	Adanya masalah perilaku (0 = tidak, 1 = ya)
28.	*ADL*	Skor aktivitas kehidupan sehari-hari, berkisar antara 0 – 10. (Skor yang lebih rendah menunjukkan gangguan yang lebih besar
29.	*Confusion*	Adanya kebingungan (0 = tidak, 1 = ya)
30.	*Disorientation*	Adanya disorientasi (0 = tidak, 1 = ya)
31.	*PersonalityChanges*	Adanya perubahan kepribadian (0 = tidak, 1 = ya)
32.	*DifficultyCompletingTasks*	Adanya kesulitan dalam menyelesaikan tugas (0 = tidak, 1 = ya)
33.	*Forgetfulness*	Adanya kelupaan (0 = tidak, 1 = ya)
34.	*Diagnosis*	Status diagnosis untuk penyakit Alzheimer (0 = tidak, 1 = ya)
35.	*DoctorInCharge*	Informasi rahasia tentang dokter yang bertanggung jawab, XXXConfid sebagai nilai untuk semua pasien

Setelah memahami penjelasan dari setiap variabel tersebut, langkah selanjutnya adalah meninjau implementasi fitur-fitur tersebut ke dalam struktur data yang sesungguhnya. Mengingat total data yang digunakan dalam penelitian ini cukup besar, yaitu mencapai 2.149 rekam medis pasien, penyajian keseluruhan data secara utuh dalam satu halaman naskah tentu tidak efisien. Sebagai solusinya, Tabel 3.2 berikut ini menampilkan gambaran visual dari dataset penyakit Alzheimer dengan format yang lebih ringkas namun tetap mencerminkan karakteristik secara utuh. Tabel tersebut memuat cuplikan data yang diambil dari posisi awal atau urutan pertama, bagian tengah, serta posisi akhir dataset guna memberikan wawasan komprehensif mengenai pola sebaran nilai serta konsistensi format input yang akan menjadi landasan utama dalam tahapan pemrosesan selanjutnya.

Tabel 3.2 Dataset Alzheimer

No	***PatientID***	***Age***	**…**	***SystolicBP***	***DiastolicBP***	**…**	***Diagnosis***	***DoctorInCharge***
1	4751	73	…	142	72	…	0	XXXConfid
2	4752	89	…	115	64	…	0	XXXConfid
…	…	…	…	…	…	…	…	…
1074	5824	72	…	167	69	…	0	XXXConfid
1075	5825	60	…	154	98	…	0	XXXConfid
…	…	…	…	…	…	…	…	…
2148	6898	78	…	103	96	…	1	XXXConfid
2149	6899	72	…	166	78	…	0	XXXConfid

Berbeda dengan Tabel 3.2 yang menggambarkan dataset secara keseluruhan, Tabel 3.3 menampilkan fokus spesifik pada sepuluh sampel data yang dipilih secara berurutan mulai dari *PatientID* ke-4764. Pengambilan sampel terbatas ini bukan bertujuan untuk merepresentasikan distribusi data secara global, melainkan ditujukan secara khusus sebagai bahan studi kasus untuk simulasi perhitungan manual algoritma XGBoost yang akan dibahas pada sub-bab

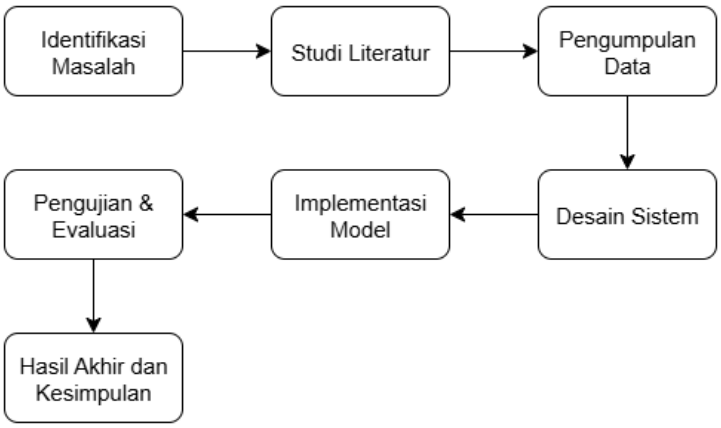
berikutnya. Dengan menggunakan dataset berskala kecil ini, tahapan kompleks dalam pembentukan *tree*, perhitungan *gain*, hingga penentuan prediksi akhir dapat dijabarkan secara rinci dan transparan, sehingga memudahkan pemahaman mengenai logika matematis yang bekerja di balik model klasifikasi.

Tabel 3.3 Sampel 10 Dataset

No	***PatientID***	***Age***	**…**	***SystolicBP***	***DiastolicBP***	**…**	***Diagnosis***	***DoctorInCharge***
1	4764	78	…	137	82	…	1	XXXConfid
2	4765	64	…	94	98	…	0	XXXConfid
3	4766	69	…	124	72	…	1	XXXConfid
4	4767	63	…	148	116	…	1	XXXConfid
5	4768	65	…	154	61	…	1	XXXConfid
6	4769	72	…	132	107	…	0	XXXConfid
7	4770	68	…	144	98	…	1	XXXConfid
8	4771	82	…	120	93	…	1	XXXConfid
9	4772	65	…	178	89	…	0	XXXConfid
10	4773	82	…	120	71	…	0	XXXConfid

3.2 Desain Penelitian

Desain penelitian dibuat untuk memberikan arah yang jelas dalam pelaksanaan penelitian, mulai dari perumusan masalah hingga pada tahap penarikan kesimpulan. Dengan adanya rancangan yang tersusun secara sistematis, setiap tahapan penelitian dapat dilakukan lebih teratur, sehingga hasil yang dicapai memiliki tingkat ketepatan yang lebih baik serta sesuai dengan tujuan yang ingin dicapai.



Gambar 3.1 Tahapan Penelitian

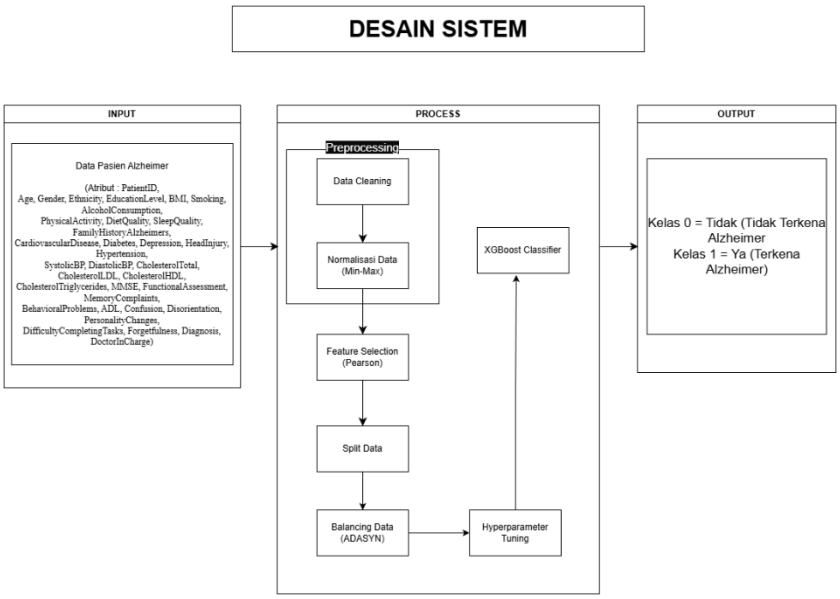
Pada Gambar 3.1 dijelaskan tahapan-tahapan dalam penelitian ini. Proses penelitian diawali dengan identifikasi masalah, di mana penulis berusaha merumuskan permasalahan yang berkaitan dengan deteksi penyakit Alzheimer. Setelah permasalahan ditentukan, langkah selanjutnya adalah melakukan studi literatur untuk mempelajari penelitian-penelitian sebelumnya yang relevan, khususnya yang membahas mengenai penerapan *machine learning* dalam klasifikasi penyakit Alzheimer.

Tahap pengumpulan data menggunakan dataset sekunder berjudul *"Alzheimer's Disease Dataset"* yang tersedia secara terbuka di *platform Kaggle*. Data yang diperoleh kemudian diproses dalam tahap perancangan sistem, yang mencakup persiapan data dan penyusunan alur pemodelan yang akan digunakan. Selanjutnya dilakukan implementasi model, yaitu penerapan algoritma XGBoost untuk klasifikasi Alzheimer sesuai rancangan yang telah dibuat. Model yang sudah diimplementasikan kemudian melalui tahap pengujian dan evaluasi untuk menilai kinerja serta akurasinya. Tahap akhir penelitian adalah analisis hasil, yang

digunakan untuk menarik kesimpulan terhadap keseluruhan proses yang telah dilakukan.

3.3 Desain Sistem

Berdasarkan Gambar 3.2, alur sistem dalam penelitian ini dibagi menjadi tiga bagian besar, yaitu input, proses, dan output. Pada bagian input, sistem menerima data pasien yang berisi berbagai atribut terkait kondisi pasien, seperti informasi demografis, gaya hidup, riwayat kesehatan, hingga hasil pemeriksaan yang berhubungan dengan Alzheimer. Data tersebut kemudian masuk ke tahap proses, dimulai dari data cleaning untuk memastikan data siap digunakan, lalu dilakukan normalisasi *Min–Max* agar nilai setiap fitur berada pada skala yang sama. Setelah itu sistem melakukan seleksi fitur dengan Pearson untuk memilih fitur-fitur yang paling relevan terhadap diagnosis, kemudian data dibagi menjadi data latih dan data uji. Pada skenario tertentu, data latih juga melalui tahap penyeimbangan kelas menggunakan ADASYN supaya model tidak bias terhadap kelas mayoritas. Selanjutnya, model dibangun menggunakan *XGBoost Classifier* dan dilakukan tuning hyperparameter agar konfigurasi model yang digunakan benar-benar optimal. Hasil akhirnya ditampilkan pada bagian output, yaitu prediksi kelas 0 yang tidak terindikasi Alzheimer atau 1 yang terindikasi Alzheimer sesuai dengan kondisi pasien.



Gambar 3.2 Desain sistem

Setiap tahapan dalam desain sistem akan dijelaskan secara terperinci pada bagian berikutnya. Penjabaran ini disusun untuk memberikan pemahaman yang lebih komprehensif mengenai prosedur yang diterapkan dalam proses perancangan sistem secara keseluruhan. Dengan penjelasan yang runtut dan sistematis, pembaca diharapkan dapat mengikuti alur kerja sistem dengan lebih mudah serta memahami dasar logika di balik setiap langkah yang dilakukan. Selain itu, uraian yang mendalam pada tiap tahap juga dapat membantu memastikan bahwa proses implementasi sistem dilakukan secara konsisten sesuai metodologi yang telah direncanakan.

3.3.1 Data *Preprocessing*

Data *preprocessing* merupakan tahap awal yang sangat penting dalam penelitian ini untuk mempersiapkan dataset Alzheimer sebelum digunakan oleh algoritma XGBoost. Proses ini meliputi serangkaian langkah yang bertujuan untuk

membersihkan, menata, dan menyesuaikan data agar lebih terstruktur sehingga dapat diolah secara optimal oleh model klasifikasi. Pada pembahasan ini akan dilakukan contoh proses penerapan *preprocessing* dengan menggunakan 10 sampel data yang ada pada Tabel 3.3.

3.3.1.1 Data *Cleaning*

Proses pembersihan data atau data *cleaning* untuk memastikan dataset dalam kondisi rapi dan siap diproses lebih lanjut. Langkah pertama yaitu menjadikan atribut *PatientID* sebagai indeks agar setiap data pasien memiliki penanda unik sehingga memudahkan pengolahan data. Informasi tersebut bisa dilihat di Tabel 3.4.

Proses pembersihan data atau *data cleaning* merupakan tahapan penting untuk memastikan dataset berada dalam kondisi terstruktur dan siap untuk diproses lebih lanjut. Langkah pertama yang dilakukan adalah menetapkan atribut *PatientID* sebagai indeks, bukan sebagai fitur, maka diberi *bold*. Hal ini bertujuan agar setiap data pasien memiliki penanda unik yang memudahkan pengolahan, sekaligus memastikan bahwa nomor identitas tersebut tidak ikut terhitung dalam operasi matematis model. Selanjutnya, kolom *DoctorInCharge* dihapus dari dataset karena teridentifikasi tidak memberikan kontribusi yang berarti terhadap proses klasifikasi. Atribut tersebut hanya berfungsi sebagai informasi administratif yang bersifat konstan, sehingga tidak memiliki hubungan korelasi dengan variabel yang mempengaruhi prediksi penyakit Alzheimer. Jika fitur yang tidak relevan seperti ini tetap dipertahankan, model berpotensi menerima gangguan atau *noise* yang justru dapat mendistorsi pola pembelajaran dan mengurangi akurasi. Oleh karena

itu, penghapusan kolom ini menjadi langkah penting untuk menjamin bahwa proses pemodelan hanya berfokus pada fitur-fitur klinis yang benar-benar valid secara akademik dalam mendukung kinerja algoritma. Hasil akhir dari penyesuaian struktur data ini dapat dilihat secara rinci pada Tabel 3.4.

Tabel 3.4 Contoh Hasil Data *Cleaning*

PatientID	***Age***	**…**	***SystolicBP***	***DiastolicBP***	**…**	***Forgetfulness***	***Diagnosis***
4764	78	…	137	82	…	0	1
4765	64	…	94	98	…	0	0
4766	69	…	124	72	…	1	1
4767	63	…	148	116	…	0	1
4768	65	…	154	61	…	0	1
4769	72	…	132	107	…	1	0
4770	68	…	144	98	…	1	1
4771	82	…	120	93	…	1	1
4772	65	…	178	89	…	0	0
4773	82	…	120	71	…	0	0

Kualitas dataset menjadi lebih baik serta meminimalkan kemungkinan adanya atribut yang tidak relevan yang memengaruhi hasil analisis setelah dilakukannya pembersihan data (Zega et al., 2022). Proses pembersihan ini juga memastikan bahwa setiap fitur yang digunakan benar-benar memiliki kontribusi terhadap proses klasifikasi sehingga model dapat bekerja secara lebih efisien. Selain itu, penghapusan data yang tidak diperlukan dapat mengurangi potensi bias dan meningkatkan stabilitas model ketika dilakukan pelatihan. Dengan demikian, tahapan pembersihan data menjadi langkah fundamental dalam membangun model yang akurat dan dapat diandalkan.

3.3.1.2 Normalisasi Data

Normalisasi data adalah proses untuk menyamakan skala antar fitur dalam dataset agar tidak ada satu fitur yang nilainya terlalu besar atau terlalu kecil dibandingkan fitur lainnya. Dengan normalisasi, semua fitur bisa berkontribusi

secara seimbang dalam proses pelatihan model *machine learning*. Hal ini penting karena perbedaan skala antar fitur dapat membuat hasil prediksi kurang akurat. Dalam penelitian ini menggunakan *Min-Max Normalization* (Permana & Salisah, 2022).

Min-Max Normalization adalah metode normalisasi yang digunakan untuk mengubah skala data sehingga nilai setiap atribut berada pada rentang 0 hingga 1. Teknik ini bertujuan untuk memastikan bahwa seluruh fitur memiliki skala yang seragam, sehingga algoritma pembelajaran mesin tidak memberikan bobot yang berlebihan pada atribut dengan nilai besar. Selain itu, normalisasi ini juga membantu meningkatkan stabilitas model dan mempercepat proses pelatihan, terutama pada algoritma yang sensitif terhadap skala seperti *gradient boosting*. Normalisasi data menggunakan *Min-Max Normalization* dapat menggunakan rumus pada Persamaan 3.1 berikut :

$$x' = \frac{x_i - min(x)}{max(x) - min(x)} \tag{3.1}$$

Keterangan :
x' : Nilai hasil normalisasi.
x_i : Nilai asli data
$min(x)$: Nilai terkecil dari suatu atribut.
$max(x)$: Nilai terbesar dari suatu aribut.

Contoh perhitungan *manual* menggunakan sebagian sampel data untuk menunjukkan bagaimana proses *Min-Max Normalization* diterapkan secara bertahap. Sebelum melakukan perhitungan, terlebih dahulu ditentukan atribut mana yang akan digunakan sebagai contoh:

1. Dari data Age {78, 64, 69, 63, 65, 72, 68, 82, 65, 82}
   x' : ?
   x_i : 78
   $min(x)$: 63

$max(x)$: 82

2. Proses normalisasi dilakukan dengan menerapkan rumus *Min-Max* pada setiap atribut. Dengan menggunakan Persamaan 3.1, didapat hasil perhitungan normalisasi sampel data seperti berikut:

$$x' = \frac{78 - 63}{82 - 63} = \frac{15}{19} = 0.789$$

Setelah dilakukan perhitungan manual tiap datanya, hasil akan ditampilkan pada Tabel 3.5 yang merupakan contoh hasil normalisasi *Min–Max* menggunakan 10 data sampel, sehingga dapat terlihat bahwa nilai pada setiap kolom fitur sudah berada pada skala yang sama, sementara kolom Diagnosis tetap merepresentasikan label kelas yaitu 0 dan 1 sebagai target klasifikasi.

Tabel 3.5 Hasil Contoh Normalisasi

PatientID	***Age***	**…**	***SystolicBP***	***DiastolicBP***	**…**	***Forgetfulness***	***Diagnosis***
4764	0.789474	…	0.511905	0.381818	…	0.0	1
4765	0.052632	…	0.000000	0.672727	…	0.0	0
4766	0.315789	…	0.357143	0.200000	…	1.0	1
4767	0.000000	…	0.642857	1.000000	…	0.0	1
4768	0.105263	…	0.714286	0.000000	…	0.0	1
4769	0.473684	…	0.452381	0.836364	…	1.0	0
4770	0.263158	…	0.595238	0.672727	…	1.0	1
4771	1.000000	…	0.309524	0.581818	…	1.0	1
4772	0.105263	…	1.000000	0.509091	…	0.0	0
4773	1.000000	…	0.309524	0.181818	…	0.0	0

Normalisasi pada Tabel 3.6 dilakukan dilakukan agar setiap fitur dalam dataset memiliki skala yang seimbang sehingga tidak ada atribut dengan rentang nilai yang terlalu besar mendominasi proses perhitungan dalam model. Langkah ini menjadi penting karena algoritma seperti XGBoost sangat sensitif terhadap variasi skala fitur, terutama ketika nilai antaratribut berbeda jauh. Dengan melakukan normalisasi, setiap fitur diproyeksikan ke rentang yang sama sehingga kontribusi

masing-masing fitur menjadi lebih proporsional. Pendekatan ini juga membantu model belajar pola secara lebih stabil dan mengurangi potensi bias akibat perbedaan skala antarfitur. Dengan demikian, normalisasi *Min-Max* berperan dalam meningkatkan efektivitas proses pelatihan model secara keseluruhan.

3.3.2 *Feature Selection*

Tahapan *feature selection* dilakukan untuk memilih fitur-fitur yang memiliki pengaruh paling kuat terhadap variabel target (*Diagnosis*). Pemilihan fitur ini bertujuan untuk meningkatkan kinerja model dengan mengurangi atribut yang kurang relevan, sehingga proses pelatihan menjadi lebih efisien tanpa kehilangan informasi penting. Pada penelitian ini, metode yang digunakan adalah korelasi *Pearson*, yaitu pendekatan statistik yang mengukur kekuatan hubungan linear antara masing-masing fitur dengan variabel target (Kumar & Rani, 2024).

Nilai korelasi *Pearson* berkisar antara -1 hingga 1. Nilai mendekati 1 atau -1 menunjukkan hubungan yang kuat, sedangkan nilai mendekati 0 menunjukkan hubungan yang lemah. Dalam penelitian ini, pemilihan *threshold* korelasi *pearson* dijadikan sebagai salah satu variabel eksperimen untuk menguji dampaknya terhadap performa model. Penelitian ini akan menguji tiga level *threshold* secara sistematis yaitu, 0.0, 0.15, dan 0.25. Tujuan dari pengujian *threshold* ini adalah untuk mengetahui secara nyata apakah kinerja model XGBoost lebih optimal saat dilatih dengan kumpulan fitur yang lengkap dimana akan digunakan *threshold* 0.0, atau saat dilatih dengan kumpulan fitur yang lebih sedikit namun memiliki korelasi sangat kuat dengan menggunakan *threshold* 0.25. Nilai 0.15 akan digunakan sebagai acuan pembanding dalam analisis ini. Hasil perhitungan korelasi *pearson*

terhadap 10 sampel data pasien, diperoleh nilai korelasi masing-masing fitur terhadap variabel target Diagnosis *(Alzheimer)* seperti ditunjukkan pada daftar hasil. Dari perhitungan tersebut, fitur yang memiliki nilai korelasi absolut di atas ambang batas 0,15 dipertahankan untuk proses pelatihan model XGBoost. Berdasarkan pada 10 sampel dataset yang ada pada Tabel 3.3, akan dilakukan perhitungan pada fitur *PhysicalActivity*.

Tahapan *feature selection* dilakukan menggunakan metode korelasi *pearson*, yang mengukur kekuatan hubungan linear antara setiap fitur (X) dengan variabel target Diagnosis (Y). Dalam konteks penelitian ini, di mana fitur (X) bersifat kontinu dan variabel target (Y) bersifat biner (0/1), perhitungan *pearson* ini secara matematis identik dengan koefisien korelasi *point-biserial*. Rumus korelasi *pearson* ditunjukkan pada Persamaan 3.2 berikut :

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \times \sum(Y_i - \bar{Y})^2}} \tag{3.2}$$

Keterangan:
X_i= nilai fitur ke–i
Y_i= nilai target diagnosis (0 = Non-Alzheimer, 1 = Alzheimer)
$\bar{X}$= rata-rata nilai fitur
$\bar{Y}$= rata-rata nilai target

Contoh perhitungan untuk fitur *PhysicalActivity:*
X = nilai *PhysicalActivity* : { 0.000000, 1.000000, 0.255988, 0.721282, 0.074317, 0.715232, 0.377554, 0.633055, 0.935208, 0.775633 }.
Y = Diagnosis: { 1, 0, 1, 1, 1, 0, 1, 1, 0, 0 }
n = 10

Maka untuk perhitungannya seperti di bawah ini:

a. Hitung rata-rata

$$\bar{X} = \frac{\sum X_i}{10} = \frac{5.488269}{10} = 0.5488269$$

$$\bar{Y} = \frac{\sum Y_i}{10} = \frac{6}{10} = 0.6$$

b. Hitung komponen perbaris

Secara umum, cara menghitung komponen baris dapat dinyatakan dalam Persamaan 3.3 berikut:

$$(X_i - \bar{X}), \quad (Y_i - \bar{Y}), \quad (X_i - \bar{X})(Y_i - \bar{Y}), \quad (X_i - \bar{X})^2, \quad (Y_i - \bar{Y})^2 \qquad (3.3)$$

Tabel 3.6 Contoh Proses Perhitungan Korelasi *Pearson* Untuk Fitur *PhysicalActivity*

i	X_i	Y_i	$X_i - \bar{X}$	$Y_i - \bar{Y}$	$(X-\bar{X})(Y-\bar{Y})$	$(X-\bar{X})^2$	$(Y-\bar{Y})^2$
1	0.000000	1	-0.5488269	0.4	-0.21953076	0.30119200	0.16
2	1.000000	0	0.4511731	-0.6	-0.27070386	0.20355700	0.36
3	0.255988	1	-0.2928389	0.4	-0.11713556	0.08575400	0.16
4	0.721282	1	0.1724551	0.4	0.06898204	0.02974200	0.16
5	0.074317	1	-0.4745099	0.4	-0.18980396	0.22516400	0.16
6	0.715232	0	0.1664051	-0.6	-0.09984306	0.02770100	0.36
7	0.377554	1	-0.1712729	0.4	-0.06850916	0.02933300	0.16
8	0.633055	1	0.0842281	0.4	0.03369124	0.00709400	0.16
9	0.935208	0	0.3863811	-0.6	-0.23182866	0.14928500	0.36
10	0.775633	0	0.2268061	-0.6	-0.13608366	0.05144000	0.36

Setelah komponen-komponen pada setiap baris dihitung seperti pada Tabel 3.7, langkah berikutnya adalah menjumlahkan hasil perhitungan di tiap kolom yang sudah dibuat yaitu pada kolom hasil perkalian selisih, serta kolom kuadrat selisih. Dari hasil penjumlahan tersebut, diperoleh satu nilai korelasi *pearson* untuk fitur *PhysicalActivity*, yaitu angka yang mewakili seberapa kuat hubungan fitur tersebut dengan label diagnosis Alzheimer. Nilai korelasi ini kemudian diinterpretasikan arahnya antara positif atau negatif dan kekuatannya yaitu kuat, sedang, atau lemah. Hasil akhirnya digunakan sebagai dasar seleksi fitur: jika nilai korelasi memenuhi ambang *threshold* yang ditentukan, maka fitur dipertahankan; jika tidak, fitur dapat dieliminasi karena dianggap kurang berkontribusi terhadap prediksi.

c. Jumlahkan komponen

$$\sum(X_i - \bar{X})(Y_i - \bar{Y}) = -1.23076540$$

$$\sum(X_i - \bar{X})^2 = 1.11026200$$

$$\sum(Y_i - \bar{Y})^2 = 6 \times (0.4)^2 + 4 \times ((-0.6)^2) = 6(0.16) + 4(0.36)$$

$$= 0.96 + 1.44$$

$$= 2.40$$

d. Hitung korelasi *pearson*

Menggunakan Persamaan 3.2:

$$r = \frac{-1.2307654}{\sqrt{1.1102620 \times 2.40}}$$

$$r = \frac{-1.2307654}{\sqrt{2.6646288}} = \frac{-1.2307654}{1.632368} = -0.7540$$

e. Interpretasi

Nilai korelasi *pearson* untuk fitur *PhysicalActivity* terhadap Diagnosis adalah:

$$r = -0.7540 \quad \text{atau} \quad |r| = 0.7540$$

Nilai korelasi Pearson untuk fitur *PhysicalActivity* terhadap fitur diagnosis diperoleh sebesar -0.7540. Tanda negatif pada angka ini menunjukkan adanya hubungan yang berlawanan arah, yang berarti semakin rutin seseorang melakukan aktivitas fisik, semakin kecil kemungkinannya terdiagnosis Alzheimer pada sampel data ini. Pola ini mengindikasikan bahwa olahraga atau gerak fisik bisa menjadi faktor pelindung terhadap risiko penyakit tersebut.

Namun, perlu ditekankan bahwa dalam proses seleksi fitur, fokus utamanya adalah seberapa kuat pengaruh suatu fitur, tidak peduli apakah pengaruhnya positif yaitu searah atau negatif yaitu berlawanan. Oleh karena itu, nilai korelasi ini diperlakukan sebagai nilai mutlak (absolut), sehingga -0.7540 dianggap sebagai 0.7540. Karena angka 0.7540 ini jauh lebih besar daripada ambang batas (*threshold*) 0,15 yang telah ditetapkan, maka fitur *PhysicalActivity* dinilai memiliki pengaruh yang signifikan dan dinyatakan lolos untuk digunakan dalam pelatihan model XGBoost.

3.3.3 *Split* Data

Proses *split* data membagi dataset menjadi dua bagian utama yaitu, data pelatihan (*training set*) dan data pengujian (*testing set*). Data pelatihan digunakan untuk membangun dan melatih model *machine learning*, sedangkan data pengujian berfungsi untuk mengevaluasi seberapa baik model bekerja pada data baru yang belum pernah dilihat sebelumnya. Dalam penelitian ini data pelatihan dan pengujian akan dibagi menjadi dua kelompok, yaitu menggunakan rasio pembagian 60:40 serta 80:20.

Tabel 3.7 Skenario Pembagian Data Latih dan Data Uji

Rasio Pembagian Data	Data Pelatihan	Data Pengujian
60 : 40	60 %	40 %
80 : 20	80 %	20 %

Pada Tabel 3.8 menampilkan rasio pembagian data, yaitu 60:40 dan 80:20. Pembagian data seperti ini membantu memastikan bahwa model memperoleh cukup informasi pada tahap pelatihan sehingga dapat mempelajari pola secara optimal. Pada saat yang sama, keberadaan data uji yang terpisah memungkinkan

evaluasi performa model dilakukan secara objektif tanpa bias dari proses pelatihan. Dengan demikian, hasil evaluasi dapat mencerminkan kemampuan model yang sesungguhnya dalam melakukan prediksi terhadap data baru (Aftha Harianto et al., 2025).Pendekatan ini juga memastikan bahwa model tidak hanya bekerja baik pada data latih, tetapi memiliki kemampuan generalisasi yang memadai.

3.3.4 *Balancing* Data

Penelitian ini menerapkan metode ADASYN (*Adaptive Synthetic Sampling*) untuk mengatasi permasalahan ketidakseimbangan data yang dapat memengaruhi performa model klasifikasi. Ketidakseimbangan terjadi ketika jumlah data pada kelas *non-Alzheimer* (kelas 0) jauh lebih banyak dibandingkan kelas *Alzheimer* (kelas 1). Kondisi ini berpotensi menyebabkan model cenderung bias terhadap kelas mayoritas dan mengabaikan pola pada kelas minoritas (Haibo He et al., 2008).

Pada penelitian ini digunakan metode ADASYN (*Adaptive Synthetic Sampling*) sebagai teknik *oversampling* pada data latih. Secara konsep, ADASYN bekerja mirip dengan SMOTE, yaitu menghasilkan sampel sintetis baru di sekitar data minoritas. Namun, ADASYN bersifat adaptif karena lebih banyak menambahkan sampel sintetis pada area di mana data minoritas sulit dipelajari (misalnya berada di dekat batas keputusan dengan kelas mayoritas). Dengan cara ini, proses pembelajaran model menjadi lebih fokus pada wilayah yang kompleks, sehingga kemampuan model untuk mengenali pola pada kelas Alzheimer dapat ditingkatkan tanpa harus melakukan duplikasi data secara langsung.

Tabel 3.8 Hasil ADASYN

Rasio *Split*	Diagnosis	Kelas	Jumlah Sebelum ADASYN	Jumlah Sesudah ADASYN

80 : 20	Tidak Alzheimer	0	1111	1111
	Alzheimer	1	608	1180
60 : 40	Tidak Alzheimer	0	833	833
	Alzheimer	1	456	884

Penerapan ADASYN dalam penelitian ini dilakukan setelah proses pembagian data menjadi *train* dan *test,* dan hanya diaplikasikan pada data latih. Tabel 3.9 menunjukkan bahwa pada rasio *train–test split* 80:20, data latih sebelum ADASYN terdiri dari 1.111 sampel kelas 0 dan 608 sampel kelas 1. Setelah ADASYN, jumlah kelas mayoritas tetap 1.111, sedangkan kelas Alzheimer meningkat menjadi 1.180 sampel. Pada rasio 60:40, data latih awal berisi 833 sampel kelas 0 dan 456 sampel kelas 1; setelah ADASYN, jumlah kelas 0 tetap 833, sedangkan kelas 1 bertambah menjadi 884 sampel. Dengan demikian, ADASYN tidak mengubah jumlah sampel pada kelas mayoritas, tetapi secara signifikan menambah representasi kelas Alzheimer sehingga distribusi data latih menjadi lebih seimbang. Kondisi ini diharapkan dapat membantu model XGBoost mempelajari karakteristik pasien Alzheimer dengan lebih baik dan mengurangi bias terhadap kelas tidak Alzheimer.

3.3.5 *Hyperparameter Tuning*

Hyperparameter tuning adalah proses penting untuk menemukan konfigurasi model terbaik guna memaksimalkan performa prediktif. Berbeda dengan parameter internal (seperti bobot dalam pohon) yang dipelajari model selama pelatihan, *hyperparameter* adalah pengaturan eksternal yang ditetapkan oleh peneliti sebelum proses pelatihan dimulai (misalnya, laju pembelajaran atau

jumlah pohon). Nilai tetap dari *hyperparameter* jarang sekali optimal untuk dataset yang spesifik. Oleh karena itu, *hyperparameter tuning* adalah proses sistematis untuk mencari dan menemukan kombinasi nilai *hyperparameter* terbaik yang menghasilkan model dengan kinerja evaluasi tertinggi pada data.

Penelitian ini menggunakan metode *tuning* yang lebih cerdas, yaitu *bayesian optimization*. Metode ini bekerja secara terinformasi (*informed search*), alih-alih mencoba kombinasi secara acak, metode ini membangun model probabilitas yang disebut *surrogate model* yang memetakan *hyperparameter* ke skor kinerja. Berdasarkan hasil evaluasi sebelumnya, metode ini secara cerdas memutuskan kombinasi mana yang paling menjanjikan untuk dievaluasi selanjutnya, sehingga proses pencarian menjadi jauh lebih efisien.

Fokus proses *tuning* dalam penelitian ini akan diarahkan pada lima *hyperparameter* utama XGBoost yang memiliki pengaruh paling signifikan terhadap performa model. *Hyperparameter* tersebut dipilih karena secara langsung memengaruhi kompleksitas pohon, tingkat pembelajaran, serta kemampuan model dalam menangani variasi data. Untuk memperoleh kombinasi parameter yang optimal, setiap *hyperparameter* diberikan rentang pencarian (*search space*) tertentu yang telah disesuaikan dengan praktik terbaik dalam pemodelan berbasis *boosting*. Rentang ini memungkinkan proses optimasi mengeksplorasi berbagai kemungkinan konfigurasi model secara lebih efektif, sehingga peluang menemukan parameter terbaik menjadi lebih besar. Dengan demikian, *tuning hyperparameter* diharapkan dapat meningkatkan akurasi dan stabilitas model secara keseluruhan. Seperti pada Tabel 3.10, penelitian ini menetapkan rentang nilai untuk setiap

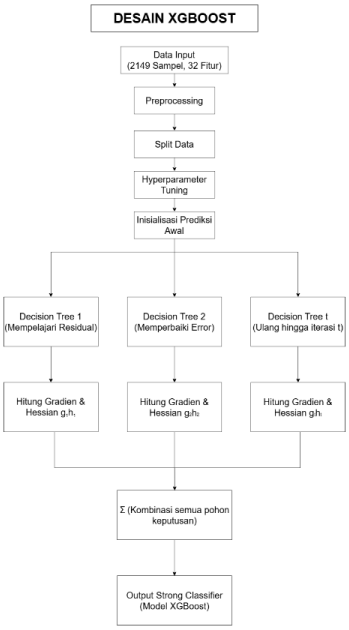
hyperparameter utama XGBoost yaitu *n_estimators, max_depth, learning_rate, subsample, dan colsample_bytree* sebagai ruang pencarian dalam proses *tuning*, sehingga model dapat menemukan kombinasi parameter yang paling optimal.

Tabel 3.9 Ruang Pencarian untuk *Hyperparameter Tuning*

Parameter	Rentang Nilai	Deskripsi
n_estimators	100 – 500	Jumlah total pohon *decision tree* yang akan dibangun.
max_depth	3 – 10	Kedalaman maksimum setiap pohon. Mengontrol kompleksitas model.
learning_rate (eta)	0.01 – 0.2	Mengatur laju pembelajaran; seberapa besar kontribusi setiap pohon baru.
subsample	0.7 – 1.0	Fraksi sampel data (baris) yang digunakan untuk melatih setiap pohon.
colsampe_bytree	0.7 – 1.0	Fraksi fitur (kolom) yang digunakan saat membangun setiap pohon.

3.3.6 Implementasi Metode *XGBoost*

Sebelum masuk ke tahap perhitungan dan simulasi manual, perlu dijelaskan terlebih dahulu rancangan implementasi metode XGBoost yang digunakan pada penelitian ini. Rancangan ini dibuat untuk menunjukkan alur kerja XGBoost secara utuh, mulai dari data hasil *preprocessing* yang menjadi masukan, hingga proses pembentukan model melalui mekanisme *boosting* berbasis pohon keputusan. alur setiap tahapan penting dapat dipahami dengan lebih jelas, seperti pembentukan pohon secara bertahap, proses perbaikan kesalahan prediksi pada iterasi berikutnya, serta hubungan antara input fitur dan keluaran prediksi, sebelum melihat penjabaran perhitungan pada satu iterasi pertama. Desain implementasi XGBoost pada penelitian ini ditampilkan pada Gambar 3.3 berikut.



Gambar 3.3 Desain Model XGBoost

Gambar 3.3 menampilkan desain model XGBoost yang digunakan dalam penelitian ini. XGBoost (*Extreme Gradient Boosting*) merupakan algoritma *ensemble* berbasis pohon keputusan yang menggunakan prinsip *boosting* dengan optimasi *gradien* untuk meningkatkan akurasi prediksi. Proses pelatihan dilakukan secara bertahap (sekuensial), di mana setiap pohon baru dibangun untuk memperbaiki kesalahan (*residual error*) dari pohon sebelumnya.

Dataset yang telah melalui tahap *preprocessing,* selanjutnya digunakan sebagai input dalam proses pelatihan model. Setiap fitur pada dataset diproses untuk membangun pohon keputusan secara bertahap sesuai dengan mekanisme *boosting* yang dimiliki XGBoost. Pada sub-bab 2.4 sebelumnya telah menguraikan komponen teoretis dari bagian proses pelatihan, yang mencakup model aditif, Fungsi objektif, *gain*, dan bobot daun. Untuk memberikan gambaran rinci mengenai proses pelatihan model XGBoost, berikut disajikan simulasi perhitungan

manual untuk satu iterasi (pohon) pertama menggunakan 10 sampel data. Perlu ditekankan bahwa perhitungan ini adalah penyederhanaan ilustratif untuk mendemonstrasikan mekanisme inti algoritma, dan bukan representasi model akhir yang kompleks. Studi kasus ini akan menggunakan 10 sampel data (6 pasien Alzheimer dan 4 *Non*-Alzheimer) yang diambil dari Tabel 3.3. Alur perhitungannya akan mengikuti langkah-langkah logis dari proses pelatihan yang telah dijelaskan secara teoretis:

1. Inisialisasi prediksi awal ($p^{(0)}$):

   Menentukan probabilitas awal ($p^{(0)}$) sebelum ada pohon yang dibangun merupakan langkah pertama yang dilakukan. Ini adalah gambaran terbaik model berdasarkan distribusi data.

   a. Jumlah data total (n) = 10

   b. Jumlah pasien Alzheimer (Y=1 = 1) = 6

   c. Jumlah pasien Tidak Alzheimer (Y=0) = 4

   Probabilitas awal dihitung menggunakan Persamaan 3.4 sebagai berikut:

$$p^{(0)} = \frac{\text{Jumlah Y} = 1}{Total\ Data} = \frac{6}{10} = 0{,}6 \tag{3.4}$$

   Setiap pasien, terlepas dari fiturnya, akan diberikan probabilitas awal 60% menderita Alzheimer. Dalam XGBoost, perhitungan sebenarnya didasarkan pada *margin* (skor mentah/logit). Margin awal ($m^{(0)}$) dihitung dari log-odds dengan Persamaan 3.5:

$$tp^{(0)} = m^{(0)} = \ln\left(\frac{p^{(0)}}{1-p^{(0)}}\right) = \ln\left(\frac{0.6}{1-0.6}\right) = \ln(1.5) \approx 0.405 \tag{3.5}$$

2. Menghitung *Gradien* (g_i) dan *Hessian* (h_i)

Error dari prediksi awal kemudian dihitung. Ini diukur oleh Gradien (turunan pertama) dan Hessian (turunan kedua) dari *log-loss*.

a. *Gradien* (g_i) menggunakan Persamaan 3.6:

$$g_i = p^{(0)} - y_i \quad \text{(Prediksi - Aktual)} \tag{3.6}$$

Maka,

- Jika pasien Alzheimer (Y=1):

$$g_i = 0.6 - 1 = -0.4$$

- Jika pasien Non-Alzheimer (Y=0):

$$g_i = 0.6 - 0 = 0.6$$

b. Hessian (h_i) menggunakan Persamaan 3.7:

$$h_i = p^{(0)} \times \left(1 - p^{(0)}\right) \tag{3.7}$$

Maka,

- Untuk semua pasien:

$$h_i = 0.6 \times (1 - 0.6) = 0.24$$

Tabel 3.11 di bawah ini merangkum data, prediksi awal, serta *gradien* dan *hessian* untuk setiap pasien.

Tabel 3.10 Data Sampel dan Contoh Perhitungan *Gradien*/*Hessian* Iterasi t=1

i	**Y (Aktual)**	**X (*PhysicalActivity*)**	***P*$^{(0)}$ (Prediksi Awal)**	***Gradien*** **(g_i)**	***Hessian*** **(h_i)**
1	1	0.000	0.6	-0.4	0.24
2	0	1.000	0.6	0.6	0.24
3	1	0.255	0.6	-0.4	0.24
4	1	0.721	0.6	-0.4	0.24
5	1	0.074	0.6	-0.4	0.24
6	0	0.715	0.6	0.6	0.24
7	1	0.377	0.6	-0.4	0.24
8	1	0.633	0.6	-0.4	0.24
9	0	0.935	0.6	0.6	0.24
10	0	0.775	0.6	0.6	0.24
Total				G = 0.0	H= 2.4

3. Membangun Pohon (f_1) – Mencari Split Terbaik

Pohon pertama (f_1) dibangun dengan mencari pemisahan (*split*) pada fitur *PhysicalActivity* yang memberikan *Gain* pada Persamaan 2.5 tertinggi. Untuk melakukannya, data diurutkan berdasarkan *PhysicalActivity* pada Tabel 3.12 dan algoritma akan menguji *split* di antara setiap nilai.

Tabel 3.11 Contoh Data Urut Berdasarkan Fitur *PhysicalActivity*

i	Y (Aktual)	X (*PhysicalActivity*)	Gradien (g_i)	*Hessian* (h_i)
1	1	0.000	-0.4	0.24
5	0	0.074	-0.4	0.24
3	1	0.255	-0.4	0.24
7	1	0.377	-0.4	0.24
8	1	0.633	-0.4	0.24
6	0	0.715	0.6	0.24
4	1	0.721	-0.4	0.24
10	1	0.775	0.6	0.24
9	0	0.935	0.6	0.24
2	0	1.000	0.6	0.24

Demonstrasi selanjutnya menguji *split* pada nilai $X \leq 0.633$.

a. Node Kiri (*Left*): $X \leq 0.633$

   Data (i): {1, 5, 3, 7, 8} (Total 5 data)

   G_L (Jumlah *Gradien* Kiri): 5 × (-0.4) = -2.0

   H_L (Jumlah *Hessian* Kiri): 5 × (0.24) = 1.2

b. Node Kanan (*Right*): $X > 0.633$

   - Data (i): {6, 4, 10, 9, 2} (Total 5 data)
   - G_R (Jumlah *Gradien* Kanan): (0.6 × 4) + (-0.4 × 1) = 2.4 - 0.4 = 2.0
   - H_R (Jumlah *Hessian* Kanan): 5 × (0.24) = 1.2

Menghitung *Gain* dengan menggunakan Persamaan 2.5 dengan parameter λ=1 (regularisasi) dan γ = 0 (penalty kompleksitas).

$$\text{Gain} = \frac{1}{2}\left[\frac{(-2.0)^2}{1.2+1} + \frac{(2.0)^2}{1.2+1} - \frac{(-2.0+2.0)^2}{(1.2+1.2)+1}\right] - 0$$

$$\text{Gain} = \frac{1}{2}\left[\frac{4}{2.2} + \frac{4}{2.2} - \frac{0}{3.4}\right]$$

$$\text{Gain} = \frac{1}{2}[1.818 + 1.818 - 0] \; = 1.818$$

Karena *Gain* > 0, split ini diterima.

4. Menghitung Bobot Daun (w_j^*)

   Nilai skor mentah (bobot) untuk setiap daun baru dihitung menggunakan Persamaan 2.6:

   - Bobot Daun Kiri (w_L):

$$w_L = -\frac{G_L}{H_L + \lambda} = -\frac{(-2.0)}{1.2+1} = \frac{2.0}{2.2} \approx 0.909$$

   - Bobot Daun Kanan (w_R):

$$w_R = -\frac{G_R}{H_R + \lambda} = -\frac{(2.0)}{1.2+1} = \frac{-2.0}{2.2} \approx -0.909$$

   Daun kiri (aktivitas fisik rendah) memberikan skor positif (menaikkan probabilitas Alzheimer), sementara daun kanan (aktivitas fisik tinggi) memberikan skor negatif (menurunkan probabilitas Alzheimer).

5. Memperbarui Prediksi (Iterasi $t = 1$)

   Perbarui prediksi probabilitas ($p^{(1)}$) untuk setiap pasien. Dapat menggunakan *learning rate* $\eta = 0.1$ untuk memperlambat proses belajar. Prediksi baru dihitung dengan memperbarui *margin* awal dengan Persamaan 3.8 dan 3.9:

$$m^{(1)} = m^{(0)} + \eta \times w_j \tag{3.8}$$

$$p^{(1)} = \text{sigmoid}\left(m^{(1)}\right) = \frac{1}{1 + e^{-m^{(1)}}} \tag{3.9}$$

- Pasien di Daun Kiri (*Node* L) menggunakan Persamaan 3.8 dan 3.9:

$$m_L^{(1)} = 0.405 + (0.1 \times 0.909) = 0.405 + 0.0909 = 0.4959$$

$$p_L^{(1)} = \frac{1}{1 + e^{-0.4959}} \approx 0.6216$$

- Pasien di Daun Kanan (*Node* R) menggunakan Persamaan 3.8 dan 3.9:

$$m_R^{(1)} = 0.405 + (0.1 \times -0.909) = 0.405 - 0.0909 = 0.3141$$

$$p_R^{(1)} = \frac{1}{1 + e^{-0.3141}} \approx 0.5779$$

Tabel 3.12 menampilkan contoh hasil akhir dari iterasi pertama.

Tabel 3.12 Contoh Hasil Akhir Iterasi Pertama

i	**Y (Aktual)**	**X (*PhysicalActivity*)**	***Node***	***p*$^{(0)}$ (Awal)**	***p*$^{(1)}$ (Baru)**	**Perubahan**
1	1	0.000	Kiri	0.6	0.6216	Naik
2	0	1.000	Kanan	0.6	0.5779	Turun
3	1	0.255	Kiri	0.6	0.6216	Naik
4	1	0.721	Kanan	0.6	0.5779	Turun
5	1	0.074	Kiri	0.6	0.6216	Naik
6	0	0.715	Kanan	0.6	0.5779	Turun
7	1	0.377	Kiri	0.6	0.6216	Naik
8	1	0.633	Kiri	0.6	0.6216	Naik
9	0	0.935	Kanan	0.6	0.5779	Turun
10	0	0.775	Kanan	0.6	0.5779	Turun

6. Kesimpulan Iterasi

Hasil setelah satu iterasi menunjukkan model telah berhasil memperbarui prediksinya. Pasien di *Node* Kiri dimana 5/5 adalah Alzheimer, probabilitasnya meningkat dari 60% menjadi 62.16%. Pasien di *Node* Kanan dimana 4/5 adalah Non-Alzheimer, probabilitasnya menurun dari 60% menjadi 57.79%.

Proses prediksi akhir yang sesungguhnya adalah melanjutkan proses ini. Nilai $p^{(1)}$ (probabilitas baru) ini akan menjadi dasar untuk menghitung *gradien* dan *hessian* baru pada iterasi kedua (t = 2), yang kemudian akan membangun pohon kedua (f_2) untuk memperbaiki *error* yang masih tersisa, dan begitu seterusnya.

Perhitungan manual tersebut mengilustrasikan satu iterasi dari proses Pelatihan (Bagian A) Dalam implementasi penuh, proses ini diulang puluhan atau ratusan kali (sesuai jumlah *estimators*) untuk membangun *ensemble* model atau kumpulan pohon yang utuh. Setelah model *ensemble* ini selesai dilatih menggunakan data latih, model tersebut siap digunakan untuk Proses Klasifikasi (Bagian B). Pada tahap ini, data baru, khususnya data uji yang telah disisihkan, akan dimasukkan ke seluruh pohon dalam *ensemble* untuk mendapatkan agregasi skor mentah. Skor ini kemudian diubah menjadi probabilitas, dan akhirnya ditetapkan menjadi prediksi kelas (0 atau 1).

Model yang telah dibangun harus diukur kinerjanya untuk mengetahui seberapa akurat dan andal kemampuannya dalam mengklasifikasikan data yang belum pernah dilihat sebelumnya. Oleh karena itu, langkah penting berikutnya setelah model selesai dilatih adalah melakukan evaluasi model.

3.4 Evaluasi Model

Evaluasi model dilakukan untuk mengetahui sejauh mana kinerja algoritma dalam melakukan prediksi. Pada penelitian ini, metode evaluasi yang digunakan adalah *confusion matrix*. *Confusion matrix* menyajikan ringkasan hasil prediksi model dalam bentuk tabel, yang memperlihatkan jumlah data uji yang

diklasifikasikan dengan benar maupun yang salah. Tabel 3.14 ini terdiri dari empat komponen penting: *True Positive* (TP) yang menunjukkan jumlah data positif yang berhasil diprediksi dengan benar, *True Negative* (TN) yang menunjukkan jumlah data negatif yang diprediksi benar, *False Positive* (FP) yaitu data negatif yang keliru diprediksi sebagai positif, dan *False Negative* (FN) yaitu data positif yang salah diprediksi sebagai negatif (Suryadewiansyah & Tju, 2022).

Tabel 3.13 *Confusion matrix*

Kelas Aktual	**Kelas Prediksi**	
	Prediksi Positif	**Prediksi Negatif**
Positif	*True Positive (TP)*	*False Negative (FN)*
Negatif	*False Positive (FP)*	*True Negative (TN)*

Kinerja model tidak hanya diukur berdasarkan jumlah prediksi yang benar dan salah, tetapi juga melalui beberapa metrik evaluasi yang lebih spesifik. Metrik yang digunakan meliputi akurasi, yang menunjukkan proporsi prediksi benar terhadap seluruh data, presisi, yang menggambarkan ketepatan model dalam memprediksi kelas positif, *recall*, yang menilai kemampuan model dalam menemukan seluruh data pada kelas positif, serta *f1-score*, yang merupakan rata-rata harmonis antara presisi dan *recall*. Perhitungan masing-masing metrik ditunjukkan pada Persamaan 3.10 hingga 3.13 berikut (Ulfah et al., 2023):

$$Akurasi(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{3.10}$$

$$Presisi(\%) = \frac{TP}{TP + FP} \times 100\% \tag{3.11}$$

$$Recall(\%) = \frac{TP}{TP + FN} \times 100\% \tag{3.12}$$

$$F1 - Score = 2 \times \frac{presisi \times recall}{presisi + recall} \tag{3.13}$$

Pemilihan metrik ini didasarkan pada kebutuhan spesifik klasifikasi medis, di mana akurasi saja seringkali tidak cukup dan bisa menyesatkan, terutama jika dataset tidak seimbang (*imbalanced*). Model yang hanya menebak kelas mayoritas bisa memiliki akurasi tinggi, namun gagal total mendeteksi pasien yang sakit. Oleh karena itu, *confusion matrix* yang gambarannya terdapat pada Tabel 3.14, digunakan sebagai dasar analisis yang lebih mendalam untuk membedah tipe kesalahan. Dalam konteks diagnosis Alzheimer, kesalahan *False Negative* (FN) yaitu pasien sakit yang diprediksi sehat adalah konsekuensi paling serius karena berarti kegagalan deteksi dini. *recall* secara spesifik mengukur kemampuan model untuk meminimalkan kesalahan fatal ini dan menemukan sebanyak mungkin kasus positif. Di sisi lain, kesalahan *False Positive* (FP) yaitu pasien sehat diprediksi sakit, sehingga dapat menyebabkan stres psikologis dan biaya tes lanjutan yang tidak perlu. presisi bertugas mengukur seberapa akurat prediksi positif model untuk meminimalkan kesalahan ini. Karena sering terjadi pertukaran antara presisi dan *recall*, *f*1-*score* digunakan sebagai metrik penyeimbang yang menghitung rata-rata harmonis keduanya. Kombinasi keempat metrik ini memberikan gambaran kinerja yang jauh lebih menyeluruh dan jujur tentang kemampuan model dalam aplikasi kritis seperti deteksi penyakit.

3.5 Skenario Pengujian

Pada bagian ini disusun skenario pengujian sebagai rancangan eksperimen untuk memastikan proses evaluasi model dilakukan secara sistematis dan terukur. Penyusunan skenario dilakukan dengan mengombinasikan beberapa tahapan *preprocessing* yang menjadi fokus penelitian, yaitu seleksi fitur berbasis korelasi

pearson dengan *threshold* 0,0; 0,15; dan 0,25, variasi rasio pembagian data latih dan data uji yaitu 80:20 dan 60:40, serta penggunaan *balancing* data menggunakan ADASYN dan tanpa *balancing* (*non*-ADASYN). Kombinasi tersebut menghasilkan beberapa konfigurasi pengujian yang setara, sehingga perbandingan performa XGBoost dapat dilakukan secara adil dan objektif pada kondisi data yang berbeda. Adapun alur pembentukan skenario pengujian secara keseluruhan ditunjukkan pada Gambar 3.4 berikut.

Gambar 3.4 Skenario Pengujian

Gambar 3.4 menunjukkan alur lengkap skenario pengujian yang digunakan dalam penelitian ini. Proses dimulai dari data mentah yang terlebih dahulu melalui tahap *data cleaning* dan normalisasi menggunakan metode *min–max* sehingga seluruh fitur berada pada rentang nilai yang seragam. Setelah itu, data yang telah

dinormalisasi dikenai proses *feature selection* berbasis korelasi pearson dengan tiga variasi *threshold*, yaitu 0,0; 0,15; dan 0,25, sehingga dihasilkan beberapa himpunan fitur terpilih. Setiap himpunan fitur tersebut kemudian dibagi menjadi data latih dan data uji dengan dua skema pembagian, yakni rasio 80:20 dan 60:40. Pada cabang skenario dengan ADASYN, data latih yang telah terbagi selanjutnya *dibalancing* dengan menambah sampel sintetis pada kelas minoritas, sedangkan pada cabang *Non*-ADASYN data latih digunakan apa adanya tanpa proses *balancing*. Seluruh kombinasi data latih dari kedua cabang tersebut kemudian digunakan dalam tahap *hyperparameter tuning* dan pelatihan model XGBoost, sehingga secara keseluruhan terbentuk 12 skenario pengujian yang dievaluasi kinerjanya pada data uji.

Penelitian ini menyusun beberapa skenario pengujian untuk mencari konfigurasi model terbaik dengan memvariasikan *preprocessing* yaitu ADASYN dan *non*-ADASYN, rasio pembagian data dengan rentang 80:20 dan 60:40, serta seleksi fitur *pearson* dengan *threshold* 0,0; 0,15; dan 0,25. Variasi ini sengaja dirancang karena tiga komponen tersebut yaitu *balancing*, rasio *split*, dan seleksi fitur merupakan faktor yang paling sering memengaruhi performa model klasifikasi pada data medis, *balancing* berpengaruh pada bias kelas, rasio *split* menentukan banyaknya data yang dipelajari model, sedangkan *threshold pearson* menentukan seberapa banyak informasi fitur yang dipertahankan. Dengan menguji beberapa nilai *threshold*, penelitian dapat melihat apakah pengurangan fitur secara bertahap masih mempertahankan pola penting atau justru menyebabkan hilangnya informasi yang penting.

Uji coba ini bertujuan untuk memahami pengaruh kombinasi *preprocessing* yaitu *balancing* dan *non-balancing* apakah berpengaruh pada kemampuan model mendeteksi kelas minoritas. *Feature selection* dengan tiga pembagian *threshold* untuk menguji apakah model XGBoost bekerja lebih baik jika diberi lebih banyak fitur (kuantitas), atau jika diberi lebih sedikit fitur yang korelasinya sangat kuat (kualitas). Lalu ada penggunaan rasio pembagian data untuk menguji stabilitas dan keandalan model, skenario 80:20 fokus pada performa dengan data latih yang lebih banyak, sementara skenario 60:40 fokus pada keandalan evaluasi dengan data uji yang lebih besar. Selanjutnya penggunaan *hyperparameter tuning* untuk menemukan konfigurasi parameter XGBoost terkuat untuk masing-masing jalur seperti yang dijelaskan pada Tabel 3.9, sehingga perbandingan akhir yaitu akurasi, presisi, *recall*, *f1-score* adalah perbandingan antara model-model yang sudah dioptimalkan secara penuh.

Pada setiap skenario, dilakukan optimasi *hyperparameter* XGBoost dalam rentang yang sama agar perbandingan hasil lebih objektif. Artinya, semua skenario mendapatkan kesepatan yang setara untuk menemukan konfigurasi terbaiknya, sehingga perbedaan performa akhir lebih dapat dikaitkan dengan perubahan variabel *preprocessing*, bukan karena perbedaan ruang pencarian *hyperparameter*. Rentang *hyperparameter* yang digunakan yang terdiri dari *n_estimators, max_depth, learning_rate, subsample*, dan *colsample_bytree* dibuat konsisten untuk seluruh skenario, sehingga evaluasi benar-benar membandingkan dampak ADASYN vs *non*-ADASYN, rasio 80:20 vs 60:40, serta *threshold* 0,0 vs 0,15 vs 0,25 pada kondisi optimasi yang sebanding.

Tabel 3.14 Skenario Pengujian

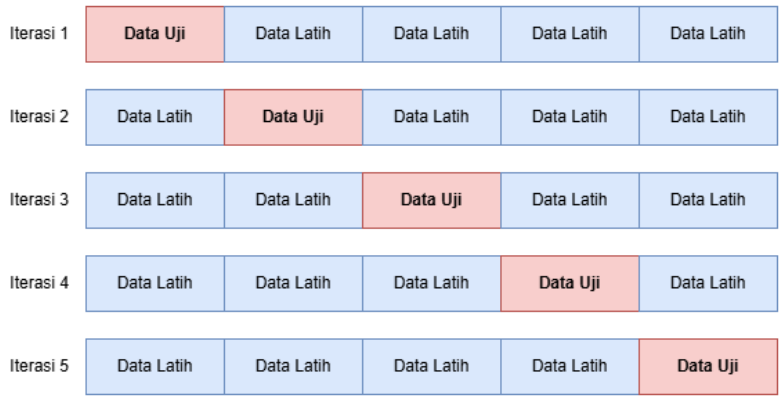
Uji Coba	*Preprocessing*	Rasio Pembagian Data	*Feature Selection (Threshold Pearson)*	*Hyperparameter XGBoost*
1	ADASYN	80:20	0.0	*n_estimators*: 100–500 *max_depth*: 3–10 *learning_rate*: 0,01–0,2 *subsample*: 0,7–1,0 *colsample_bytree*: 0,7–1,0
2	Non-ADASYN			
3	ADASYN		0.15	
4	Non-ADASYN			
5	ADASYN		0.25	
6	Non-ADASYN			
7	ADASYN	60:40	0.0	
8	Non-ADASYN			
9	ADASYN		0.15	
10	Non-ADASYN			
11	ADASYN		0.25	
12	Non-ADASYN			

Tabel 3.14 adalah menggambarkan 12 skenario penelitian untuk membandingkan performa algoritma XGBoost dalam konfigurasi *preprocessing* yang berbeda. Seluruh 12 model menggunakan algoritma XGBoost yang sama, namun dilatih dan diuji pada data yang disiapkan secara berbeda. Setiap skenario diuji dengan kombinasi tiga variabel metodologi yaitu *balancing* ADASYN vs *non*-ADASYN, *threshold feature selection* dengan nilai 0.0, 0.15, 0.25, dan rasio pembagian data dengan rentang 80:20 vs 60:40. Penelitian ini juga menerapkan *hyperparameter tuning* menggunakan *bayesian optimization* pada semua skenario untuk mencari konfigurasi terbaiknya secara otomatis, sesuai ruang pencarian (*search space*). Variasi ini dirancang untuk menentukan konfigurasi *preprocessing* optimal dalam meningkatkan kinerja klasifikasi Alzheimer.

3.5.1 *K-Fold Cross Validation*

Penelitian ini menerapkan metode *K-Fold Cross Validation* dengan nilai k=5 untuk menguji konsistensi kinerja model secara lebih objektif. Dataset dibagi menjadi lima bagian atau lipatan dengan ukuran yang sama besar, kemudian proses

pelatihan dan validasi dilakukan berulang sebanyak lima kali putaran. Pada setiap iterasi, satu bagian data digunakan khusus sebagai data validasi untuk menguji model, sedangkan empat bagian sisanya digabungkan menjadi data latih. Hasil evaluasi dari kelima putaran tersebut kemudian dirata-rata untuk mendapatkan gambaran performa model yang lebih stabil dan tidak bergantung pada satu kali pembagian data saja. Ilustrasi pembagian data menggunakan metode ini dapat dilihat pada Gambar 3.5 berikut.



Gambar 3.5 Ilustrasi 5-*Fold Cross Validation*

Gambar 3.5 menampilkan alur kerja 5-*Fold Cross Validation* yang diterapkan dalam proses pelatihan dan pengujian model ini. Setiap iterasi menunjukkan mekanisme rotasi di mana satu bagian data berperan sebagai data uji, sementara empat bagian lainnya berfungsi sebagai data latih. Proses ini terus dilakukan secara bergantian hingga seluruh bagian data pernah mendapatkan peran yang sama sebagai data uji tepat satu kali. Pendekatan ini memastikan bahwa seluruh data dimanfaatkan secara maksimal baik untuk proses pembelajaran maupun pengujian, sehingga hasil evaluasi akhir menjadi lebih akurat dan meminimalkan bias pada pembagian data tertentu.

BAB IV

UJI COBA DAN PEMBAHASAN

Bab ini menampilkan hasil pengujian model untuk klasifikasi penyakit Alzheimer menggunakan metode XGBoost. Hasil ini mencakup visualisasi proses *training internal* model. Selain itu, bab ini akan merinci hasil uji coba dari 12 skenario dan analisis pembahasan terhadap temuan tersebut.

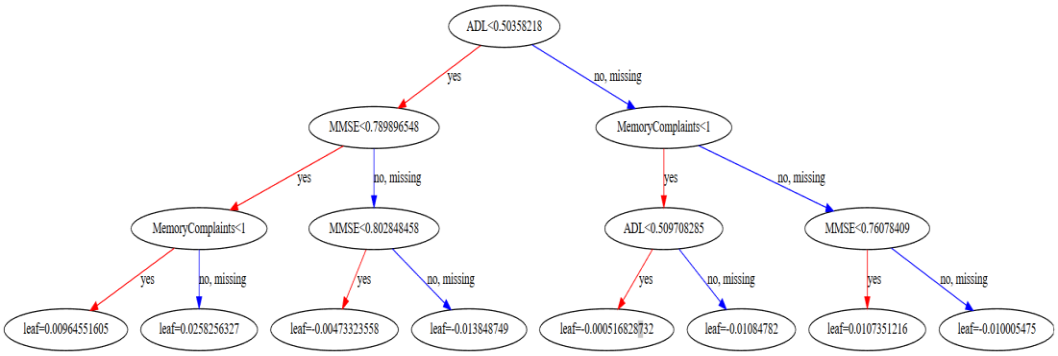
4.1 Hasil *Training*

Pada tahap ini dilakukan proses pelatihan model klasifikasi Alzheimer menggunakan algoritma XGBoost untuk dua kelas, yaitu kelas 0 (Non-Alzheimer) dan kelas 1 (Alzheimer). Sebelum model dilatih, data terlebih dahulu melalui tahapan normalisasi *min–max*, pemilihan fitur menggunakan korelasi Pearson, serta penyeimbangan kelas dengan metode ADASYN sesuai dengan variasi skenario yang telah dirancang. Setiap skenario kemudian dilatih menggunakan XGBoost dengan kombinasi *hyperparameter* yang diperoleh melalui proses pencarian berbasis *Bayesian* (BayesSearchCV) dengan fungsi objektif *F1-Score*, sehingga proses optimasi berorientasi pada keseimbangan antara presisi dan *recall*.

Selama *training*, XGBoost membangun sejumlah pohon keputusan (*weak learners*) secara bertahap, di mana setiap pohon baru ditambahkan untuk mengoreksi kesalahan prediksi dari pohon-pohon sebelumnya. Hasil pelatihan dari berbagai skenario ini kemudian menjadi dasar bagi tahap berikutnya, yaitu pengujian dan evaluasi model menggunakan metrik akurasi, presisi, *recall*, dan *F1-Score* pada data uji. Sebelum masuk ke pembahasan hasil pengujian, subbab berikut

memvisualisasikan beberapa komponen penting dari proses pelatihan, yaitu salah satu pohon keputusan, salah satu *decision stump*, serta *learning curve* yang menggambarkan perkembangan nilai *loss* selama iterasi *boosting*.

4.1.1 Visualisasi Salah Satu Pohon Keputusan

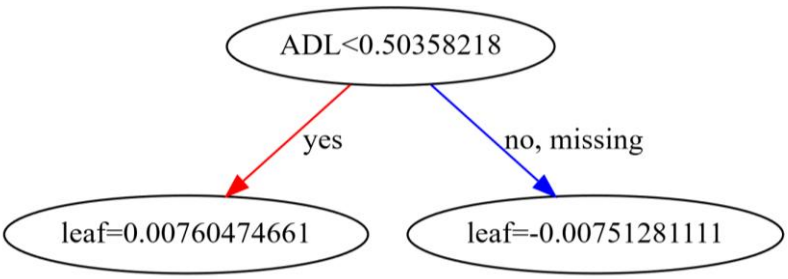


Gambar 4.1 Visualisasi Salah Satu Hasil Pembangunan Pohon Keputusan

Gambar 4.1 menunjukkan visualisasi dari salah satu pohon keputusan yang dibangun oleh algoritma XGBoost selama proses pelatihan model pada skenario 10. Pohon ini merepresentasikan satu *weak learner* yang menyusun serangkaian aturan berbasis fitur-fitur yang dianggap paling informatif terhadap status Alzheimer pasien. Pada simpul akar terlihat bahwa fitur *Activities of Daily Living* (ADL) digunakan sebagai pemisah pertama dengan suatu nilai ambang tertentu, sehingga kemampuan pasien dalam melakukan aktivitas sehari-hari menjadi faktor awal yang sangat menentukan arah keputusan model. Pada level berikutnya muncul fitur *Mini Mental State Examination* (MMSE) dan *MemoryComplaints*, yang menggambarkan kondisi fungsi kognitif dan keluhan memori pasien, sehingga ketiga variabel ini dapat dianggap memberikan kontribusi besar dalam memisahkan kelas 0 dan kelas 1.

Model memeriksa nilai fitur pada setiap simpul dan membandingkannya dengan *threshold* yang tertera pada *node*. Jika kondisi terpenuhi (misalnya ADL < *threshold*), sampel diarahkan ke cabang *yes*, sedangkan jika tidak terpenuhi sampel diarahkan ke cabang *no/missing*. Proses ini berlanjut secara bertingkat hingga mencapai simpul daun (*leaf*) yang berisi nilai skor prediksi mentah. Skor pada simpul daun tidak langsung menunjukkan kelas 0 atau 1, tetapi merupakan kontribusi logit yang kemudian dijumlahkan dengan skor dari pohon-pohon lain dalam *ensemble* XGBoost. Dengan demikian, pohon keputusan ini dapat dipandang sebagai salah satu aturan keputusan yang menyumbang bagian tertentu terhadap prediksi akhir model, sekaligus memberikan gambaran yang lebih mudah dipahami mengenai bagaimana model memanfaatkan ADL, MMSE, dan keluhan memori untuk membedakan pasien Non-Alzheimer dan Alzheimer.

4.1.2 Visualisasi Salah Satu *Decision Stump*



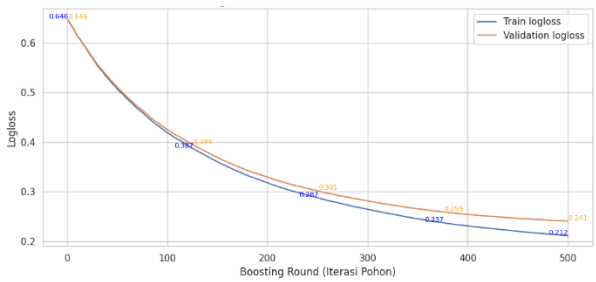
Gambar 4.2 Visualisasi *Decision Stump* pada XGBoost

Gambar 4.2 menampilkan visualisasi salah satu *decision stump* yang terbentuk pada proses pelatihan XGBoost di skenario 10. *Decision stump* adalah bentuk pohon keputusan paling sederhana karena hanya memiliki satu simpul pemisah dan dua simpul daun. Pada stump ini, pemisahan dilakukan menggunakan

fitur ADL dengan ambang 0,50358218. Jika nilai ADL lebih kecil dari ambang tersebut (*yes*), data diarahkan ke daun kiri dengan nilai *leaf* = 0,00760474661 (kontribusi positif). Sebaliknya, jika nilai ADL tidak lebih kecil dari ambang (*no*) atau bernilai hilang (*missing*), data diarahkan ke daun kanan dengan nilai *leaf* = −0,00751281111 (kontribusi negatif). Ini menunjukkan bahwa ADL menjadi indikator awal yang mampu memisahkan sampel secara kasar melalui satu aturan sederhana.

Meskipun sangat sederhana, *stump* ini menggambarkan mekanisme dasar pembelajaran XGBoost. Setiap aturan pemisahan menghasilkan kontribusi skor pada daun untuk menggeser prediksi model. Nilai *leaf* yang positif cenderung mendorong prediksi ke kelas target (Alzheimer), sedangkan nilai *leaf* yang negatif mendorong ke kelas sebaliknya (*non*-Alzheimer). Dalam praktiknya, XGBoost tidak berhenti pada satu *stump*, tetapi membangun banyak *stump* secara bertahap; *stump* berikutnya dibuat untuk memperbaiki kesalahan dari *stump* sebelumnya. Akumulasi kontribusi dari banyak *stump* dengan fitur dan ambang yang berbeda inilah yang membentuk model *ensemble* yang lebih stabil dan akurat dibandingkan satu *stump* tunggal.

4.1.3 Visualisasi Kurva *Error*



Gambar 4.3 Visualisasi Kurva Pembelajaran *Error* Model XGBoost

Gambar 4.3 menunjukkan salah satu *learning curve* model XGBoost pada skenario 10 yang menggambarkan perubahan nilai *logloss* pada data latih dan data validasi selama proses *boosting*. Pada awal pelatihan (iterasi ke-1), nilai logloss pada data latih berada di sekitar 0,6460 dan pada data validasi sekitar 0,6459, yang menunjukkan bahwa model masih berada pada tahap awal pembelajaran dan belum mampu membedakan dua kelas dengan baik. Seiring bertambahnya jumlah pohon yang ditambahkan ke dalam model, kedua kurva logloss menurun secara konsisten. Misalnya, pada iterasi ke-125 logloss latih turun menjadi sekitar 0,387 dan logloss validasi menjadi sekitar 0,394, kemudian pada iterasi ke-250 kembali menurun menjadi sekitar 0,288 untuk data latih dan 0,301 untuk data validasi.

Learning curve model XGBoost terbaik tanpa *balancing* ADASYN yang menggambarkan perubahan nilai logloss pada data latih dan data validasi selama proses boosting. Pada awal pelatihan (iterasi ke-1), nilai logloss pada data latih berada di sekitar 0,6460 dan pada data validasi sekitar 0,6459, yang menunjukkan bahwa model masih berada pada tahap awal pembelajaran dan belum mampu membedakan dua kelas dengan baik. Seiring bertambahnya jumlah pohon yang ditambahkan ke dalam model, kedua kurva logloss menurun secara konsisten. Misalnya, pada iterasi ke-125 logloss latih turun menjadi sekitar 0,387 dan logloss validasi menjadi sekitar 0,394, kemudian pada iterasi ke-250 kembali menurun menjadi sekitar 0,288 untuk data latih dan 0,301 untuk data validasi.

4.2 Hasil Uji Coba

Hasil dari 12 skenario pengujian yang telah dirancang akan dijelaskan pada bagian ini. Setiap skenario menggunakan algoritma XGBoost yang dilatih

berdasarkan tiga variabel utama yaitu, rasio pembagian data dengan 80:20 atau 60:40, lalu *balancing* data menggunakan ADASYN atau tanpa *balancing* data(80:20 atau 60:40), dan *threshold feature selection pearson* dengan 0.0, 0.15, atau 0.25. Data yang telah melalui proses normalisasi dan *feature selection* kemudian digunakan dalam proses pelatihan model *XGBoost*.

Untuk setiap skenario, proses *hyperparameter tuning* dilakukan terlebih dahulu pada data latih dengan pendekatan *Bayesian optimization* menggunakan *BayesSearchCV*. Pencarian ini bertujuan untuk menemukan kombinasi parameter terbaik, yaitu *n_estimators*, *max_depth*, *learning_rate*, *subsample*, dan *colsample_bytree*, dalam rentang nilai yang telah ditentukan. Setelah konfigurasi *hyperparameter* terbaik diperoleh, model *final* dari masing-masing skenario dilatih ulang pada data latih dan kemudian dievaluasi menggunakan data uji. Kinerja model diukur menggunakan metrik akurasi, presisi, *recall*, dan *f1-score*.

4.2.1 Model 1

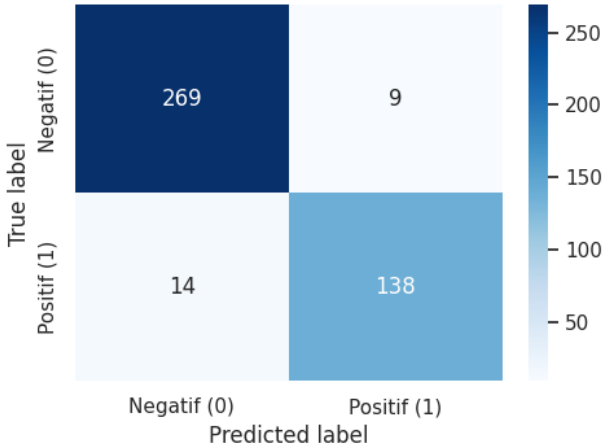
Model 1 dibangun menggunakan algoritma XGBoost dengan rancangan *preprocessing* yang telah ditetapkan. Pada skenario ini, data dibagi menjadi set pelatihan dan pengujian dengan rasio 80:20 sehingga diperoleh 1.719 data latih dan 430 data uji. Data latih kemudian diseimbangkan menggunakan metode ADASYN untuk mengatasi ketimpangan distribusi kelas, di mana jumlah kelas minoritas ditingkatkan dari 608 menjadi 1.139 sampel. Dengan demikian, total data latih bertambah menjadi 2.250 sampel setelah proses penyeimbangan. Selain itu, seleksi fitur dilakukan menggunakan korelasi *pearson* dengan threshold 0,0 yang menyebabkan seluruh 32 fitur tetap digunakan dalam pelatihan model.

Data latih yang telah diproses melalui *preprocessing* tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci (Tabel 3.9) untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 1 ditampilkan pada Tabel 4.1:

Tabel 4.1 *Hyperparameter* Terbaik Pada Model 1

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	386
max_depth	[3 - 10]	7
learning_rate	[0.01 - 0.2]	0.03
subsample	[0.7 - 1.0]	0.97
colsample_bytree	[0.7 - 1.0]	0.93

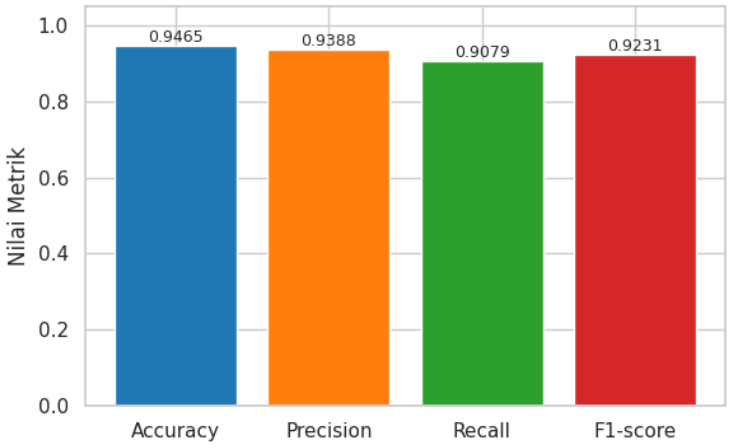
Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.1 setelah proses tuning selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji (430 data) yang telah disisihkan. Pengujian ini bertujuan untuk mengevaluasi sejauh mana model mampu melakukan prediksi terhadap data baru. Hasil *confusion matrix* pada model 1 untuk data *testing* ditampilkan pada Gambar 4.4.



Gambar 4.4 *Confusion Matrix* Model 1

Gambar 4.4 menampilkan *confusion matrix* yang merinci distribusi prediksi benar dan salah dari model 1. Berdasarkan *matrix* tersebut, model mampu memprediksi kelas dengan benar sebanyak 407 data, yang terdiri dari 269 *true negative* (pasien sehat terprediksi sehat) dan 138 *true positive* (pasien Alzheimer terprediksi Alzheimer). Di sisi lain, terdapat 23 data yang salah klasifikasi, yang terdiri dari 9 *false positive* (pasien sehat diprediksi Alzheimer) dan 14 *false negative* (pasien Alzheimer diprediksi sehat). Keberadaan *false negative* ini perlu diperhatikan karena dalam konteks medis, pasien Alzheimer yang tidak terdeteksi memiliki risiko terabaikan dalam penanganan.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.5 untuk menunjukkan analisis kinerja model 1 secara lebih jelas.



Gambar 4.5 Performa Model 1

Model ini menghasilkan akurasi sebesar 94,65%, presisi 93,88%, *recall* 90,79%, dan *f1-score* 92,31%. Visualisasi performa pada Gambar 4.5 menunjukkan bahwa model memiliki kemampuan prediksi yang baik, ditandai dengan presisi dan *recall* yang sama-sama tinggi. Hasil ini menunjukkan bahwa kombinasi dari penggunaan seluruh fitur melalui *threshold pearson* 0.0, dan *balancing* menggunakan ADASYN, serta optimasi *hyperparameter* berbasis *bayesian optimization* mampu menghasilkan performa klasifikasi yang kuat dalam mendeteksi penyakit Alzheimer pada data uji.

4.2.2 Model 2

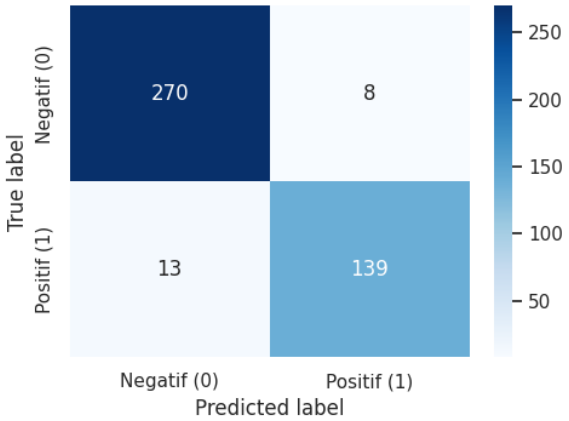
Uji coba model 2 menggunakan algoritma *XGBoost* dengan konfigurasi *preprocessing* yang serupa dengan model 1, namun tanpa penerapan *balancing ADASYN* pada data latih. Data tetap dibagi dengan rasio *train–test split* 80:20 sehingga diperoleh 1.719 data latih dan 430 data uji. Proses seleksi fitur menggunakan korelasi *pearson* dilakukan dengan *threshold* 0,0 sehingga seluruh 32 fitur tetap digunakan dalam pelatihan model. Tidak digunakannya ADASYN pada skenario ini menyebabkan distribusi kelas pada data latih mengikuti kondisi asli dataset, yaitu 1.111 sampel kelas *Non*-Alzheimer dan 608 sampel kelas Alzheimer.

Data latih yang tidak seimbang dan telah diseleksi fiturnya tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci yang ada pada Tabel 3.9 untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 2 ditampilkan pada Tabel 4.2.

Tabel 4.2 *Hyperparameter* Terbaik Pada Model 2

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	421
max_depth	[3 - 10]	7
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	0.85
colsample_bytree	[0.7 - 1.0]	0.94

Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.2 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji, terdiri dari 430 data yang telah disisihkan. Pengujian ini bertujuan untuk mengevaluasi sejauh mana model mampu melakukan prediksi terhadap data baru. Hasil *confusion matrix* pada model 2 untuk data *testing* ditampilkan pada Gambar 4.6.

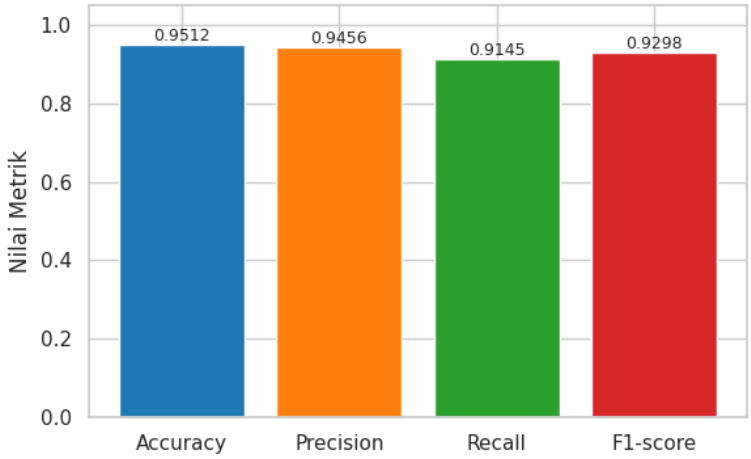


Gambar 4.6 *Confusion Matrix* Model 2

Gambar 4.6 menampilkan *confusion matrix* yang merinci hasil prediksi model 2 pada data uji yang tidak seimbang. Berdasarkan *matrix* tersebut, model mampu memprediksi kelas dengan benar sebanyak 409 data, yang terdiri dari 270 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 139 *true positive* (pasien Alzheimer terprediksi Alzheimer). Sementara itu, terdapat 21 data yang salah klasifikasi, yaitu 8 *false positive* (pasien *Non*-Alzheimer diprediksi

Alzheimer) dan 13 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Nilai *false negative* ini perlu menjadi perhatian, karena berkaitan dengan potensi keterlambatan diagnosis pada pasien Alzheimer.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.7 untuk menunjukkan analisis kinerja model 2 secara lebih jelas.



Gambar 4.7 Performa Model 2

Gambar 4.7 menampilkan performa dari model 2 yaitu menghasilkan akurasi sebesar 95,12%, presisi 94,56%, *recall* 91,45%, dan *f1-score* 92,98%. Visualisasi keempat metrik ini dalam bentuk diagram batang pada gambar di atas menunjukkan bahwa seluruh metrik berada pada rentang nilai tinggi, yang menandakan bahwa model 2 memiliki kemampuan klasifikasi yang sangat baik meskipun tanpa proses *balancing* data. Dibandingkan model 1, model 2

menghasilkan presisi dan *f1-score* yang sedikit lebih tinggi, menunjukkan bahwa tidak adanya ADASYN justru memungkinkan model menurunkan prediksi positif yang keliru. Meskipun *recall* sedikit lebih rendah karena jumlah *false negative* yang sedikit meningkat, model 2 tetap menunjukkan kinerja yang kuat dan seimbang dalam mengidentifikasi pasien Alzheimer, sehingga dapat dipertimbangkan sebagai pendekatan yang efektif pada data uji dengan distribusi kelas asli.

4.2.3 Model 3

Uji coba model 3 menggunakan algoritma *XGBoost* dengan konfigurasi *preprocessing* yang mirip dengan model 1, yaitu menerapkan *balancing ADASYN* dan rasio *train–test split* 80:20. Perbedaan utama dibanding model 1 terletak pada nilai *threshold* korelasi *pearson* yang digunakan untuk *feature selection*, yaitu sebesar 0,15. Proses seleksi fitur meloloskan 5 fitur, yaitu *MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems,* dan *ADL*. Setelah normalisasi dan *oversampling* dengan *ADASYN* selesai, data latih yang semula berjumlah 1.719 sampel meningkat menjadi 2.265 sampel yang lebih seimbang antara kelas 0 dan kelas 1.

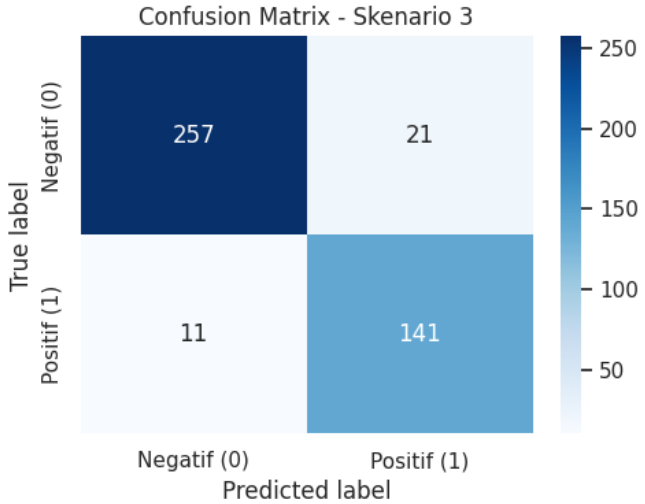
Data latih yang telah diproses melalui *balancing* ADASYN dan 5 fitur terpilih tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 3 ditampilkan pada Tabel 4.3.

Tabel 4.3 *Hyperparameter* Terbaik Pada Model 3

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	500
max_depth	[3 - 10]	10

Hyperparameter	Nilai *Bayesian*	Nilai Parameter Terbaik
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	1.0
colsample_bytree	[0.7 - 1.0]	1.0

Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.3 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji terdiri dari 430 data yang telah disisihkan. Pengujian ini bertujuan untuk mengevaluasi sejauh mana model mampu melakukan prediksi terhadap data baru. Hasil *confusion matrix* pada model 3 untuk data *testing* ditampilkan pada Gambar 4.8.

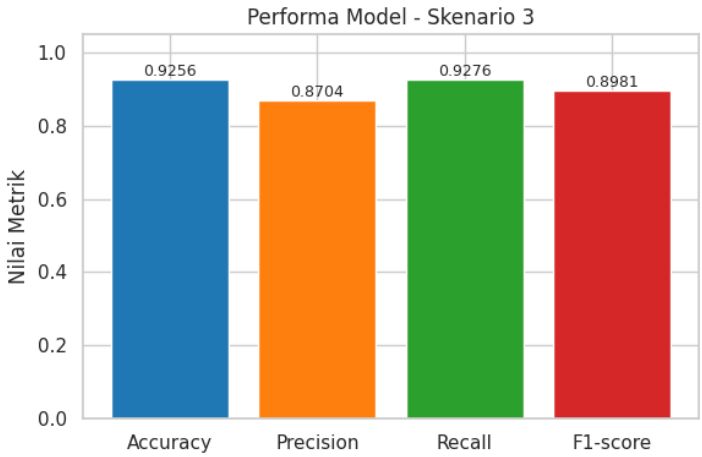


Gambar 4.8 *Confusion Matrix* Model 3

Gambar 4.8 menampilkan model 3 mampu memprediksi dengan benar sebanyak 398 data, yang terdiri dari 257 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 141 *true positive* (pasien Alzheimer terprediksi Alzheimer). Sementara itu, terdapat 32 data yang salah klasifikasi, yaitu 21 *false positive* (pasien *Non*-Alzheimer diprediksi Alzheimer) dan 11 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Keberadaan *alse positive*

menunjukkan bahwa beberapa sampel *Non*-Alzheimer masih terprediksi sebagai Alzheimer, sedangkan *false negative* menunjukkan adanya sejumlah kasus Alzheimer yang belum berhasil terdeteksi oleh model. Pola kesalahan ini menggambarkan bahwa meskipun model 3 memiliki kinerja yang baik secara keseluruhan, kemampuan pemisahan antarkelas masih memerlukan perhatian lebih lanjut, terutama untuk memastikan bahwa kedua kelas dapat dikenali secara konsisten.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.9 untuk menunjukkan analisis kinerja model 3 secara lebih jelas.



Gambar 4.9 Performa Model 3

Gambar 4.9 menampilkan hasil performa model 3 yaitu menghasilkan akurasi sebesar 92,56%, presisi 87,04%, *recall* 92,76%, dan *f1-score* 89,81%.

Keempat metrik tersebut menunjukkan bahwa model 3 memiliki kemampuan klasifikasi yang baik, dengan *recall* yang relatif tinggi yang mengindikasikan bahwa sebagian besar kasus Alzheimer berhasil terdeteksi. Namun, nilai presisi yang lebih rendah dibandingkan *recall* menunjukkan bahwa masih terdapat sejumlah prediksi positif yang salah. Meskipun demikian, nilai *f1-score* yang mendekati 0,90 memperlihatkan bahwa model 3 tetap mampu mencapai keseimbangan yang cukup baik antara presisi dan *recall*. Hasil ini menunjukkan bahwa pada konfigurasi dengan menggunakan 5 fitur dan penerapan ADASYN, model 3 dapat menghasilkan performa yang stabil dalam mendeteksi penyakit Alzheimer, meskipun tidak setinggi model sebelumnya.

4.2.4 Model 4

Uji coba model 4 menggunakan algoritma *XGBoost* dengan konfigurasi yang serupa dengan model 2, yaitu tanpa penerapan *balancing ADASYN* dan menggunakan rasio *train–test split* 80:20. Perbedaan utamanya adalah nilai *threshold* korelasi *pearson* untuk *feature selection* menggunakan 0,15. Namun, sama seperti pada model 3, perubahan *threshold* ini tidak mengubah himpunan fitur yang terpilih yaitu lima fitur yang digunakan tetap *MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems,* dan *ADL*. Dengan demikian, pada model 4 data latih tetap berjumlah 1.719 sampel tanpa *oversampling*, dan model belajar dari distribusi kelas asli yang sedikit tidak seimbang.

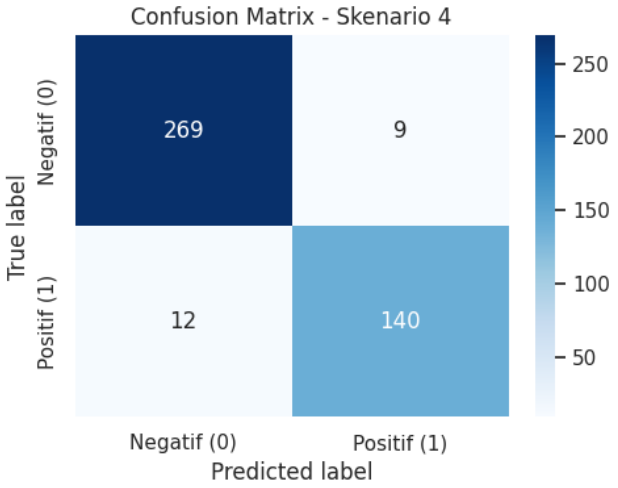
Data latih yang tidak seimbang dan telah diseleksi tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci yang

ada pada Tabel 3.9, untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 4 ditampilkan pada Tabel 4.4.

Tabel 4.4 *Hyperparameter* Terbaik Pada Model 4

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	500
max_depth	[3 - 10]	3
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	0.75
colsample_bytree	[0.7 - 1.0]	1.0

Model final dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.4 setelah proses tuning selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji (430 data) yang telah disisihkan. Hasil *confusion matrix* pada model 4 untuk data *testing* ditampilkan pada Gambar 4.10.

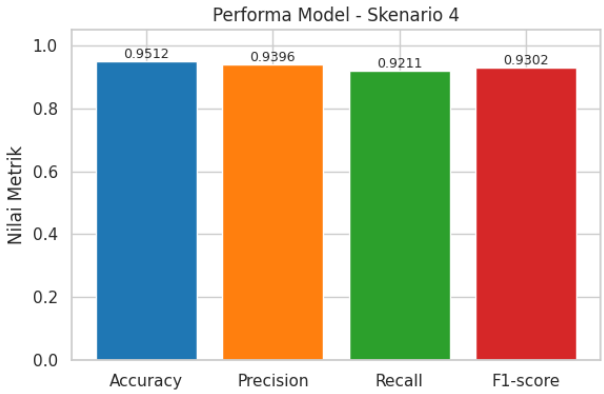


Gambar 4.10 *Confusion Matrix* Model 4

Gambar 4.10 menampilkan model 4 memprediksi dengan benar sebanyak 409 data, terdiri atas 269 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 140 *true positive* (pasien Alzheimer terprediksi Alzheimer). Sementara itu, terdapat 21 data yang salah klasifikasi, yaitu 9 *false positive* (pasien

Non-Alzheimer diprediksi Alzheimer) dan 12 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Nilai *false positive* mengindikasikan bahwa sebagian sampel *Non*-Alzheimer masih terprediksi sebagai Alzheimer, sedangkan keberadaan *false negative* menunjukkan bahwa beberapa kasus Alzheimer belum berhasil teridentifikasi oleh model. Pola kesalahan tersebut menggambarkan bahwa meskipun model 4 mampu mencapai jumlah prediksi benar yang tinggi, kemampuan pemisahan antarkelas masih perlu diperhatikan, terutama karena kesalahan pada kategori positif dapat berdampak pada proses deteksi dini penyakit Alzheimer. Secara keseluruhan, performa ini menunjukkan bahwa model 4 tetap bekerja dengan baik meskipun jumlah fitur dibatasi melalui *threshold pearson*.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.11 untuk menunjukkan analisis kinerja model 4 secara lebih jelas.



Gambar 4.11 Performa Model 4

Gambar 4.11 menampilkan performa model 4 yaitu menghasilkan akurasi sebesar 95,12%, presisi 93,96%, *recall* 92,11%, dan *f1-score* 93,02%. Keempat metrik tersebut menunjukkan bahwa model 4 memiliki kemampuan klasifikasi yang sangat baik, dengan seluruh nilai berada di atas 0,92. Nilai presisi yang tinggi mengindikasikan bahwa sebagian besar prediksi positif yang dihasilkan model merupakan kasus Alzheimer yang benar, sementara nilai *recall* yang tetap tinggi menunjukkan bahwa sebagian besar pasien Alzheimer berhasil teridentifikasi. *F1-score* yang melampaui 0,93 juga menegaskan bahwa model mampu menjaga keseimbangan yang baik antara presisi dan *recall*. Secara keseluruhan, hasil ini menunjukkan bahwa meskipun seleksi fitur menggunakan *threshold pearson* yang lebih ketat, model 4 tetap mampu mempertahankan kualitas prediksi yang kuat pada konfigurasi tanpa *balancing* ADASYN dan rasio *split* 80:20.

4.2.5 Model 5

Uji coba model 5 menggunakan algoritma *XGBoost* dengan konfigurasi *preprocessing* yang berbeda cukup signifikan dibanding empat model sebelumnya. Pada skenario ini, data tetap dibagi dengan rasio *train–test split* 80:20, namun proses *feature selection pearson* menggunakan *threshold* yang lebih tinggi, yaitu 0,25. Penerapan *threshold* yang lebih ketat ini hanya meloloskan 3 fitur utama sehingga ruang fitur yang digunakan model menjadi lebih sederhana. Fitur-fitur yang terpilih adalah *FunctionalAssessment, MemoryComplaints, dan ADL*. Fitur *MMSE* dan *BehavioralProblems,* yang sebelumnya lolos di *threshold* 0.15, kini tereliminasi. Seperti halnya model 1 dan 3, pada model 5 juga diterapkan *balancing ADASYN* pada data latih, jumlah sampel latih meningkat dari 1.719 menjadi 2.101

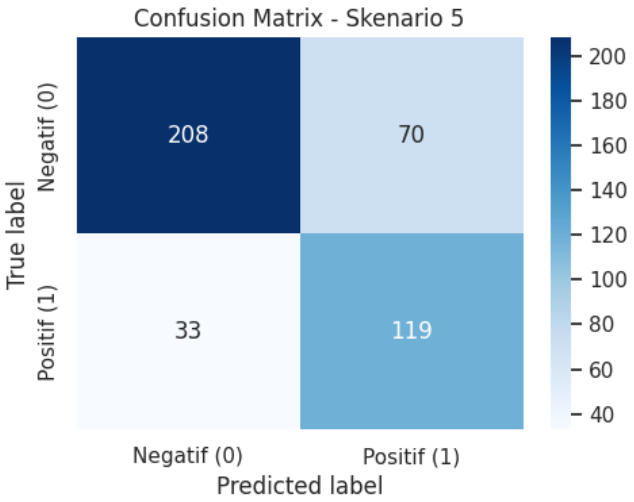
setelah proses *oversampling* sehingga distribusi kelas pada data latih menjadi lebih seimbang.

Data latih yang telah *dipreprocessing tersebut* kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 5 ditampilkan pada Tabel 4.5.

Tabel 4.5 *Hyperparameter* Terbaik Pada Model 5

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	487
max_depth	[3 - 10]	6
learning_rate	[0.01 - 0.2]	0.031
subsample	[0.7 - 1.0]	0.70
colsample_bytree	[0.7 - 1.0]	1.00

Kedalaman pohon yang menengah yaitu *max_depth* = 6, dan nilai *learning_rate* yang sedikit lebih besar menunjukkan bahwa model mencoba mengompensasi berkurangnya jumlah fitur dengan membangun pohon-pohon yang relatif lebih ekspresif, namun ruang informasi yang tersisa tetap terbatas. Setelah konfigurasi ini diperoleh, model *final* dilatih menggunakan 3 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.5 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji terdiri dari 430 data yang telah disisihkan. Hasil *confusion matrix* pada model 5 untuk data *testing* ditampilkan pada Gambar 4.12.

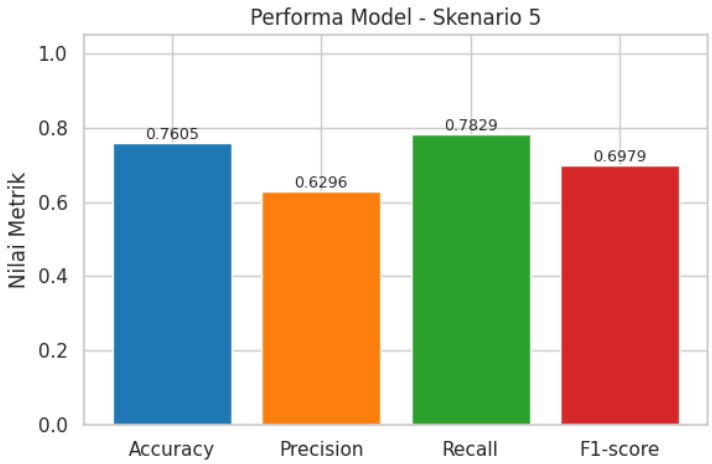


Gambar 4.12 *Confusion Matrix* Model 5

Gambar 4.12 menampilkan model 5 memprediksi dengan benar sebanyak 327 data, terdiri atas 208 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 119 *true positive* (pasien Alzheimer terprediksi Alzheimer). Di sisi lain, terdapat 103 data yang salah klasifikasi, yaitu 70 *false positive* (pasien *Non*-Alzheimer diprediksi Alzheimer) dan 33 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Jumlah *false positive* yang cukup besar menunjukkan bahwa model cenderung terlalu sering memberikan prediksi positif, sedangkan peningkatan *false negative* dibandingkan skenario sebelumnya menunjukkan penurunan kemampuan model dalam mendeteksi seluruh pasien Alzheimer.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai

tersebut kemudian ditampilkan secara visual pada Gambar 4.13 untuk menunjukkan analisis kinerja model 5 secara lebih jelas.



Gambar 4.13 Performa Model 5

Gambar 4.13 menampilkan performa model 5 yaitu menghasilkan akurasi sebesar 76,05%, presisi 62,96%, *recall* 78,29%, dan *f1-score* 69,79%. Keempat metrik tersebut menunjukkan nilai yang jauh lebih rendah dibandingkan model 1–4. Dapat dilihat bahwa kombinasi *threshold pearson* yang tinggi yaitu 0,25, dengan hanya 3 fitur dan *balancing ADASYN* tidak cukup untuk menghasilkan model dengan kemampuan generalisasi yang baik. Presisi yang hanya 62,96% mengindikasikan bahwa lebih dari sepertiga pasien yang diprediksi Alzheimer sebenarnya merupakan pasien *Non*-Alzheimer, sehingga berpotensi menimbulkan banyak *false alarm* dalam praktik klinis. Walaupun *recall* masih tergolong cukup baik dengan nilai yang diperoleh 78,29% dan menunjukkan bahwa sebagian besar pasien Alzheimer tetap dapat terdeteksi, penurunan presisi dan *f1-score* yang cukup tajam membuat model 5 kurang layak dipertimbangkan sebagai kandidat model terbaik dalam penelitian ini.

4.2.6 Model 6

Uji coba model 6 menggunakan algoritma *XGBoost* dengan konfigurasi yang sejajar dengan model 5 dari sisi pemilihan fitur, namun tanpa penerapan *balancing ADASYN* pada data latih. Data tetap dibagi menggunakan rasio *train–test split* 80:20 sehingga diperoleh 1.719 data latih dan 430 data uji. Proses *feature selection pearson* menggunakan *threshold* yang sama tinggi, yaitu 0,25, sehingga hanya tiga fitur utama yang diloloskan untuk membangun model. Berbeda dengan model 5, data latih tidak melalui proses *oversampling*, sehingga distribusi kelas pada data latih tetap mengikuti kondisi asli dataset yang tidak seimbang.

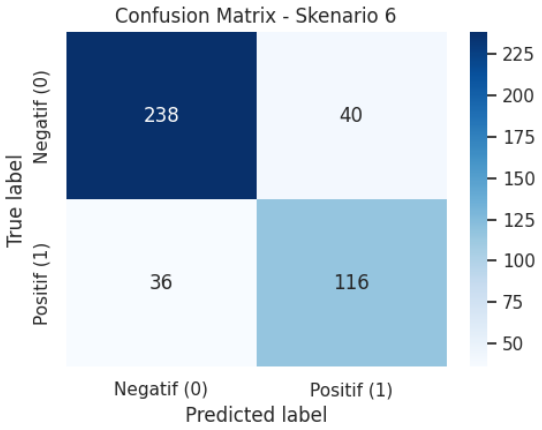
Data latih yang tidak seimbang dan telah diseleksi 3 fitur tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 6 ditampilkan pada Tabel 4.6.

Tabel 4.6 Hyperparameter Terbaik Pada Model 6

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	391
max_depth	[3 - 10]	3
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	1.00
colsample_bytree	[0.7 - 1.0]	1.00

Kombinasi ini menggambarkan model dengan pohon-pohon yang relatif dangkal tetapi cukup banyak, sehingga berusaha mengimbangi keterbatasan jumlah fitur yang digunakan. Model *final* dilatih menggunakan 3 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.6 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 20% data uji terdiri

dari 430 data yang telah disisihkan. Hasil *confusion matrix* pada model 6 untuk data *testing* ditampilkan pada Gambar 4.14.

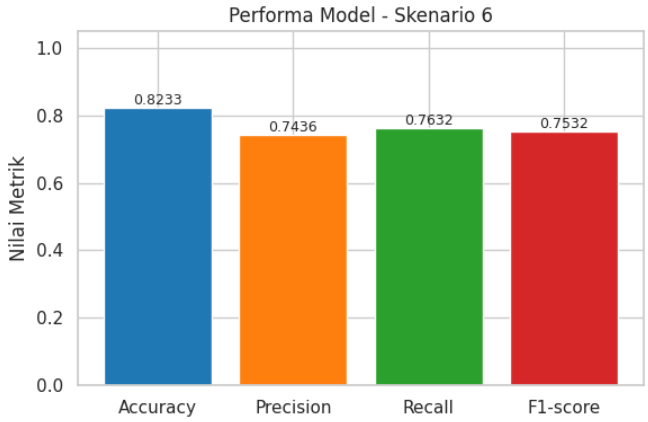


Gambar 4.14 *Confusion Matrix* Model 6

Gambar 4.14 menampilkan *confusion matrix* model 6 yang mampu mengklasifikasikan dengan benar sebanyak 354 data, yang terdiri dari 238 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 116 *true positive* (pasien Alzheimer terprediksi Alzheimer). Di sisi lain, terdapat 40 *false positive* (pasien *Non*-Alzheimer diprediksi Alzheimer) dan 36 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Jumlah *false negative* yang cukup tinggi dibandingkan beberapa skenario awal menunjukkan bahwa model 6 mulai mengalami penurunan kemampuan dalam menangkap seluruh kasus Alzheimer, sementara *false positive* yang masih cukup besar menandakan adanya risiko *false alarm* pada pasien yang sebenarnya sehat.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai

tersebut kemudian ditampilkan secara visual pada Gambar 4.15 untuk menunjukkan analisis kinerja model 6 secara lebih jelas.



Gambar 4.15 Performa Model 6

Gambar 4.15 menampilkan hasil performa model 6 yaitu menghasilkan menghasilkan akurasi sebesar 82,33%, presisi 74,36%, *recall* 76,32%, dan *f1-score* 75,32%. Dibandingkan dengan model 5, seluruh metrik mengalami peningkatan, yang menunjukkan bahwa penghilangan *balancing* ADASYN pada konfigurasi dengan tiga fitur justru memberikan perbaikan performa. Namun, jika dibandingkan dengan model-model yang menggunakan jumlah fitur lebih banyak, yaitu model 1 dan 2 dengan 32 fitur serta model 3 dan 4 dengan 5 fitur, kinerja model 6 masih berada pada tingkat yang lebih rendah. Hal ini menunjukkan bahwa penggunaan hanya tiga fitur berdasarkan *threshold pearson* 0,25 belum mampu menyediakan informasi yang cukup kuat untuk menghasilkan performa klasifikasi yang mendekati skenario dengan lebih banyak fitur. Dengan demikian, meskipun model 6 menunjukkan peningkatan dibandingkan model 5, jumlah fitur yang terbatas tetap menjadi faktor pembatas utama dalam mendeteksi penyakit Alzheimer secara lebih akurat.

4.2.7 Model 7

Uji coba model 7 dilakukan menggunakan algoritma XGBoost dengan rasio *train–test split* sebesar 60:40, sehingga diperoleh 1.289 data latih dan 860 data uji. Pada skenario ini diterapkan metode ADASYN untuk menyeimbangkan distribusi kelas pada data latih, yang meningkatkan jumlah sampel dari 1.289 menjadi 1.695 data dengan komposisi kelas yang lebih seimbang yaitu 833 sampel *Non*-Alzheimer dan 862 sampel Alzheimer. Proses seleksi fitur menggunakan korelasi *pearson* dengan *threshold* 0,0 sehingga seluruh 32 fitur tetap digunakan. Dengan demikian, model 7 mempelajari pola klasifikasi Alzheimer dari data yang telah dinormalisasi, diseleksi fiturnya tanpa pengurangan atribut, dan diseimbangkan melalui *oversampling*.

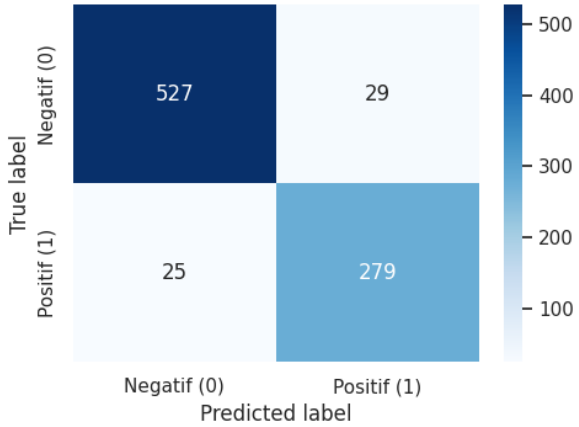
Data latih yang telah *dipreprocessing* tersebut, kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 7 ditampilkan pada Tabel 4.7.

Tabel 4.7 *Hyperparameter* Terbaik Pada Model 7

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	450
max_depth	[3 - 10]	6
learning_rate	[0.01 - 0.2]	0.1020
subsample	[0.7 - 1.0]	0.8778
colsample_bytree	[0.7 - 1.0]	0.8851

Konfigurasi ini menghasilkan model yang membangun jumlah pohon yang cukup besar dengan kedalaman menengah serta laju pembelajaran yang relatif tinggi. Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik

dari Tabel 4.7 setelah proses tuning selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 40% data uji terdiri dari 860 data yang telah disisihkan. Hasil *confusion matrix* pada model 7 untuk data *testing* ditampilkan pada Gambar 4.16.

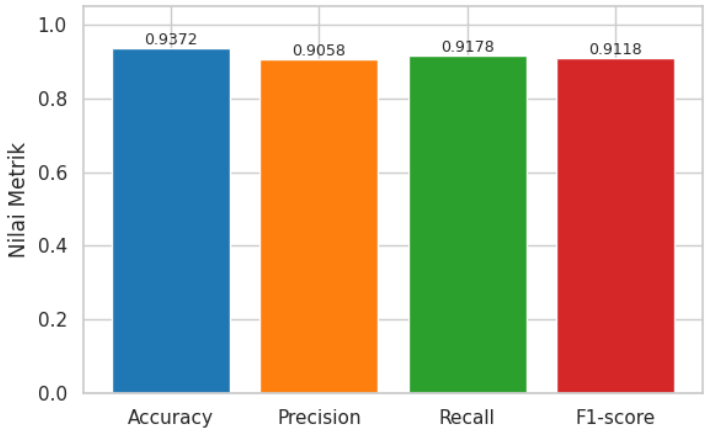


Gambar 4.16 *Confusion Matrix* Model 7

Gambar 4.16 menampilkan *confusion matrix* model 7 menghasilkan 527 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 279 *true Ppositive* (pasien Alzheimer terprediksi Alzheimer), sehingga total 806 data diklasifikasikan dengan benar. Sementara itu, terdapat 29 *false positive* dan 25 *false negative*. Nilai *false positive* menunjukkan bahwa sebagian sampel *Non*-Alzheimer masih terprediksi sebagai Alzheimer, sedangkan *false negative* menunjukkan adanya kasus Alzheimer yang belum terdeteksi. Meskipun demikian, jumlah *false negative* yang relatif kecil memperlihatkan bahwa model cukup efektif dalam menangkap sebagian besar kasus Alzheimer pada skenario ini.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*,

dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.17 untuk menunjukkan analisis kinerja model 7 secara lebih jelas.



Gambar 4.17 Performa Model 7

Gambar 4.17 menampilkan hasil performa model 7 yaitu menghasilkan akurasi sebesar 93,72%, presisi 90,58%, *recall* 91,78%, dan *f1-score* 91,18%. Keempat metrik ini menunjukkan bahwa model 7 memiliki performa yang kuat dan seimbang, dengan tingkat presisi dan *recall* yang sama-sama tinggi. Nilai *recall* di atas 91% menandakan kemampuan deteksi yang baik terhadap pasien Alzheimer, sementara nilai *f1-score* yang mendekati 0,92 memperlihatkan bahwa model mampu mempertahankan keseimbangan prediksi positif yang benar dan kesalahan prediksi positif. Secara keseluruhan, model 7 menunjukkan performa yang kompetitif pada skenario dengan rasio split 60:40, tanpa penurunan kinerja yang berarti meskipun proporsi data uji lebih besar.

4.2.8 Model 8

Model 8 dibangun dengan konfigurasi yang serupa dengan model 7, yaitu menggunakan rasio *train–test split* sebesar 60:40. Perbedaannya terletak pada tidak digunakannya proses *balancing* ADASYN, sehingga data latih tetap berjumlah 1.289 sampel dengan distribusi kelas asli, yaitu 833 sampel *Non*-Alzheimer dan 456 sampel Alzheimer. Pada skenario ini, seleksi fitur menggunakan korelasi *pearson* dengan *threshold* 0,0 yang menyebabkan seluruh 32 fitur tetap digunakan. Dengan demikian, model 8 mempelajari pola klasifikasi langsung dari data yang tidak seimbang namun dengan cakupan informasi fitur yang lengkap.

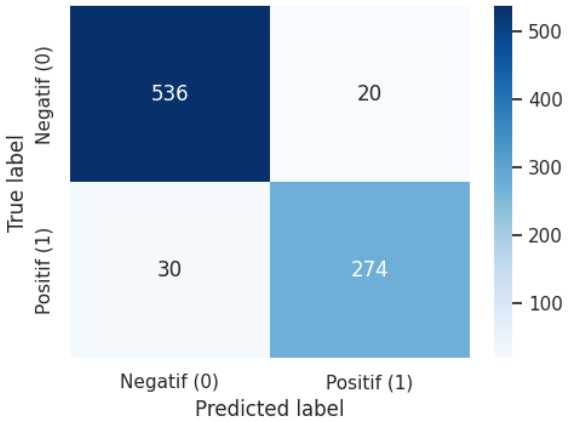
Data latih yang tidak seimbang dan telah diseleksi 5 fitur tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 8 ditampilkan pada Tabel 4.8.

Tabel 4.8 *Hyperparameter* Terbaik Pada Model 8

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	458
max_depth	[3 - 10]	10
learning_rate	[0.01 - 0.2]	0.0131
subsample	[0.7 - 1.0]	0.9871
colsample_bytree	[0.7 - 1.0]	0.9289

Konfigurasi ini membentuk model dengan kedalaman pohon maksimal dan jumlah *estimators* yang cukup besar, disertai laju pembelajaran yang rendah sehingga pembaruan bobot dilakukan secara lebih bertahap. Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.8 setelah proses tuning selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya

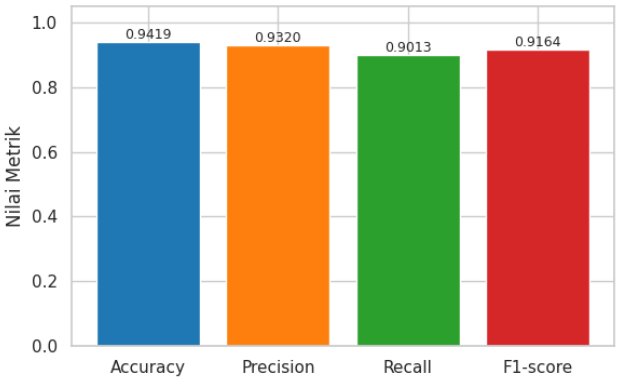
menggunakan 40% data uji (860 data) yang telah disisihkan. Hasil *confusion matrix* pada model 8 untuk data *testing* ditampilkan pada Gambar 4.18.



Gambar 4.18 *Confusion Matrix* Model 8

Gambar 4.18 menampilkan model 8 berhasil mengklasifikasikan dengan benar 810 sampel, yang terdiri dari 536 *true negative* dan 274 *true positive*. Kesalahan klasifikasi mencakup 20 *false positive* dan 30 *false negative*. *False positive* menunjukkan bahwa sebagian kecil sampel *Non*-Alzheimer masih terprediksi sebagai Alzheimer, sedangkan *false negative* mengindikasikan adanya sejumlah kasus Alzheimer yang tidak terdeteksi oleh model.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.19 untuk menunjukkan analisis kinerja model 8 secara lebih jelas.



Gambar 4.19 Performa Model 8

Gambar 4.19 menampilkan hasil performa model 8 yaitu akurasi sebesar 94,19%, presisi 93,20%, *recall* 90,13%, dan *f1-score* 91,64%. Keempat metrik ini menunjukkan bahwa model 8 memiliki kinerja yang kuat dan stabil meskipun dilatih tanpa *balancing* ADASYN. Presisi yang tinggi menunjukkan bahwa sebagian besar prediksi positif merupakan pasien Alzheimer yang benar-benar positif, sementara *recall* yang masih di atas 90% menandakan bahwa model mampu mendeteksi sebagian besar kasus Alzheimer. Nilai *f1-score* yang mendekati 0,92 juga memperlihatkan keseimbangan yang baik antara presisi dan *recall*, sehingga model 8 dapat dikategorikan sebagai model yang andal pada skenario dengan distribusi kelas asli.

4.2.9 Model 9

Model 9 dirancang dengan konfigurasi yang sama dengan model 7 dari sisi *preprocessing*, yaitu rasio *train–test split* 60:40 dan penggunaan *balancing ADASYN* pada data latih. Perbedaannya hanya terletak pada nilai *threshold* Pearson yang dinaikkan menjadi 0,15. Namun demikian, seperti yang terjadi pada beberapa skenario sebelumnya, perubahan *threshold* ini tidak mengubah himpunan fitur yang

terpilih; lima fitur yang digunakan tetap *MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems,* dan *ADL*. Setelah proses *oversampling* dengan *ADASYN*, jumlah data latih meningkat dan distribusi kelas menjadi lebih seimbang, sehingga model 9 mempelajari pola klasifikasi Alzheimer dari data yang telah dinormalisasi, dipilih fiturnya, dan diseimbangkan secara sistematis.

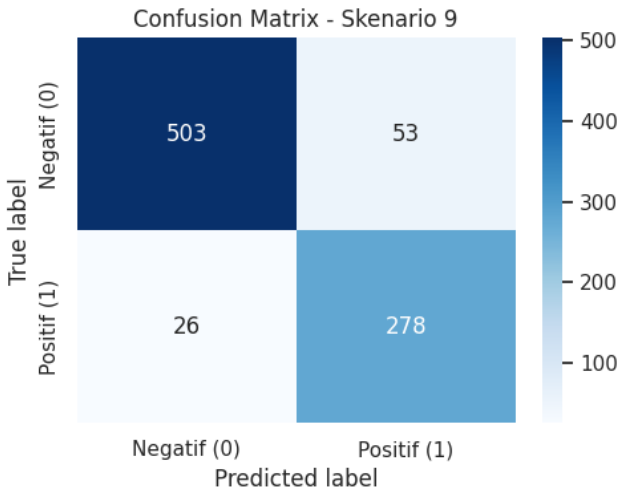
Data latih yang telah di*preprocessing* tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 9 ditampilkan pada Tabel 4.9.

Tabel 4.9 *Hyperparameter* Terbaik Pada Model 9

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	181
max_depth	[3 - 10]	10
learning_rate	[0.01 - 0.2]	0.071
subsample	[0.7 - 1.0]	0.93
colsample_bytree	[0.7 - 1.0]	1.0

Hasil *tuning* pada model 9 menunjukkan bahwa penggunaan *threshold pearson* sebesar 0,15 tidak mengubah komposisi fitur yang digunakan dalam pelatihan. Hal ini menunjukkan bahwa fitur-fitur yang memiliki korelasi kuat terhadap target tetap terpilih meskipun nilai *threshold* diperketat, sehingga proses optimasi cenderung memberikan preferensi yang konsisten terhadap kombinasi *hyperparameter* yang dihasilkan. Model *final* kemudian dilatih menggunakan lima fitur terpilih dan *hyperparameter* terbaik sebagaimana ditunjukkan pada Tabel 4.9. Setelah proses pelatihan selesai, model dievaluasi menggunakan 860 data uji atau 40% dari total dataset. Hasil evaluasi tersebut divisualisasikan pada Gambar 4.20

melalui *confusion matrix*, yang memberikan gambaran mengenai distribusi prediksi benar dan salah dari model 9.

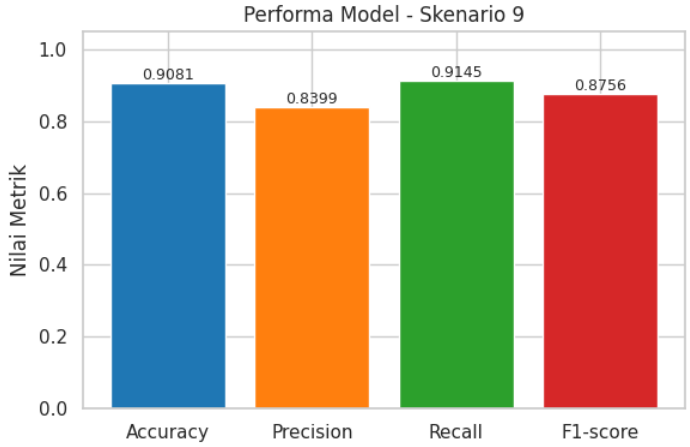


Gambar 4.20 *Confusion Matrix* Model 9

Gambar 4.20 menampilkan *confusion matrix* yang merinci hasil prediksi model 9 pada 860 data uji. Menariknya, *confusion matrix* yang dihasilkan identik dengan model 7, yaitu 503 *true negative*, 278 *true positive*, 53 *false positive*, dan 26 *false negative*. Dengan demikian, jumlah prediksi benar dan salah pada kedua model ini sama persis. Kondisi ini mengindikasikan bahwa, dalam konfigurasi dengan *ADASYN* dan lima fitur terpilih, perubahan kecil pada nilai *threshold* Pearson tidak memberikan pengaruh berarti terhadap perilaku model dalam mengklasifikasikan data uji.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai

tersebut kemudian ditampilkan secara visual pada Gambar 4.21 untuk menunjukkan analisis kinerja model 9 secara lebih jelas.



Gambar 4.21 Performa Model 9

Gambar 4.21 menampilkan hasil performa model 9 yang menghasilkan akurasi sebesar 90,81%, presisi 83,99%, *recall* 91,45%, dan *f1-score* 87,56%. Keempat metrik tersebut menunjukkan bahwa model 9 memiliki kemampuan klasifikasi yang cukup baik, terutama pada aspek *recall* yang tinggi, yang mengindikasikan bahwa sebagian besar kasus Alzheimer berhasil terdeteksi oleh model. Nilai presisi yang lebih rendah dibandingkan *recall* menunjukkan bahwa masih terdapat sejumlah prediksi positif yang tidak tepat, namun kinerja keseluruhannya tetap stabil. Hasil ini memperlihatkan bahwa penggunaan lima fitur terpilih melalui *threshold pearson* 0,15, dikombinasikan dengan *balancing* menggunakan ADASYN, mampu menghasilkan performa yang konsisten pada skenario dengan rasio *split* 60:40. Dengan demikian, model 9 tetap dapat menunjukkan kemampuan klasifikasi yang memadai meskipun jumlah fitur lebih sedikit dibandingkan skenario lain yang menggunakan himpunan fitur yang lebih lengkap.

4.2.10 Model 10

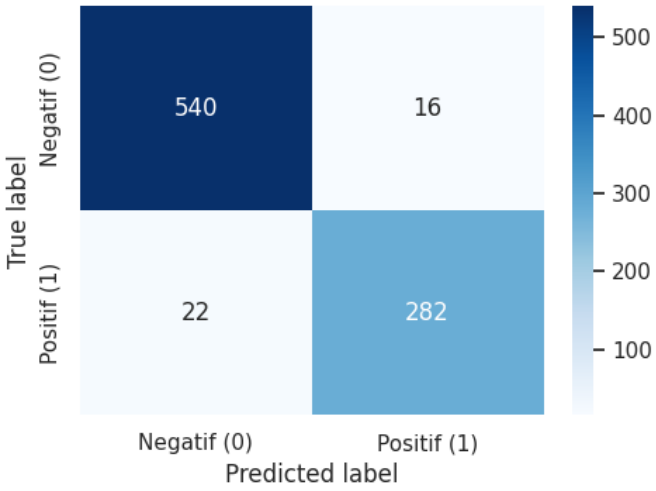
Uji coba model 10 menguji algoritma XGBoost pada data latih yang tidak seimbang yaitu *non*-ADASYN dengan rasio *split* 60:40. Skenario ini menggunakan *feature selection pearson* dengan *threshold* 0.15. Tujuan utamanya adalah untuk melihat performa model *imbalanced* saat dilatih dengan data latih yang lebih sedikit yaitu 60% dan dievaluasi pada data uji yang lebih besar yaitu 40%. Berdasarkan *threshold* 0.15, proses seleksi fitur yang dilakukan pada data yang tidak di-*balance* meloloskan 5 fitur untuk digunakan. Fitur-fitur yang terpilih adalah: *MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems,* dan *ADL*.

Data latih yang telah di*preprocessing* tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 10 ditampilkan pada Tabel 4.10.

Tabel 4.10 *Hyperparameter* Terbaik Pada Model 10

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	500
max_depth	[3 - 10]	3
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	0.92
colsample_bytree	[0.7 - 1.0]	0.7

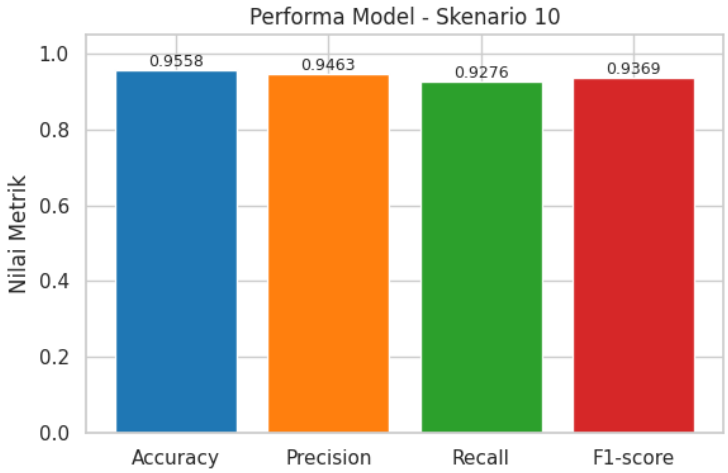
Model *final* dilatih menggunakan 5 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.10 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 40% data uji terdiri dari 860 data yang telah disisihkan. Hasil *confusion matrix* pada model 10 untuk data *testing* ditampilkan pada Gambar 4.22.



Gambar 4.22 *Confusion Matrix* Model 10

Gambar 4.22 menampilkan *confusion matrix* yang merinci hasil prediksi model 10 pada 860 data uji. Berdasarkan *matrix* tersebut, model mampu memprediksi kelas dengan benar sebanyak 822 data, yang terdiri dari 540 *true negative* (pasien sehat terprediksi sehat) dan 282 *true positive* (pasien alzheimer terprediksi alzheimer). Sementara itu, tercatat terdapat 38 data yang salah prediksi, yang terdiri dari 16 *false positive* (pasien sehat diprediksi alzheimer) dan 22 *false negative* (pasien alzheimer diprediksi sehat).

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.23 untuk menunjukkan analisis kinerja model 10 secara lebih jelas.



Gambar 4.23 Performa Model 10

Gambar 4.23 menampilkan hasil performa model 10 yaitu menghasilkan akurasi sebesar 95,58%, presisi 94,63%, *recall* 92,76%, dan *f1-score* 93,69%. Nilai metrik yang tinggi dan seimbang tersebut menunjukkan bahwa model 10 memiliki kemampuan klasifikasi yang kuat meskipun hanya menggunakan lima fitur terpilih berdasarkan *threshold pearson* 0,15. Nilai presisi yang mendekati 95% mengindikasikan bahwa sebagian besar prediksi positif merupakan kasus Alzheimer yang benar, sementara nilai *recall* yang berada di atas 92% menunjukkan bahwa model mampu mendeteksi sebagian besar pasien Alzheimer pada data uji. Secara keseluruhan, hasil ini memperlihatkan bahwa konfigurasi tanpa *balancing* ADASYN dan dengan jumlah fitur yang lebih sedikit tetap mampu menghasilkan performa yang kompetitif. Model 10 dapat dianggap sebagai salah satu model yang paling efektif dalam skenario rasio *split* 60:40, karena berhasil mencapai kombinasi akurasi, presisi, *recall*, dan *f1-score* yang termasuk tertinggi di antara model-model dengan jumlah fitur yang terbatas.

4.2.11 Model 11

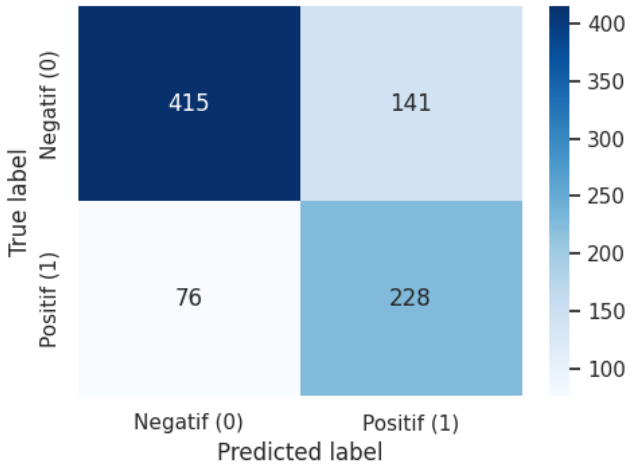
Uji coba model 11 menguji rasio *split* 60:40 dan *balancing* ADASYN, namun kali ini menggunakan *feature selection pearson* dengan *threshold* tinggi (0.25). Skenario ini bertujuan untuk melihat performa model ketika dilatih hanya pada fitur-fitur dengan korelasi terkuat (kualitas), dan diuji pada data uji yang besar yaitu 40%. Berdasarkan *threshold* 0.25, proses seleksi fitur yang dilakukan pada data latih yang sudah *dibalance* meloloskan 3 fitur untuk digunakan. Fitur-fitur yang terpilih adalah: *FunctionalAssessment, MemoryComplaints,* dan *ADL*.

Data latih yang telah diproses *balancing* ADASYN dan 3 fitur terpilih tersebut, kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 11 ditampilkan pada Tabel 4.11.

Tabel 4.11 *Hyperparameter* Terbaik Pada Model 11

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	455
max_depth	[3 - 10]	8
learning_rate	[0.01 - 0.2]	0.01
subsample	[0.7 - 1.0]	0.82
colsample_bytree	[0.7 - 1.0]	0.89

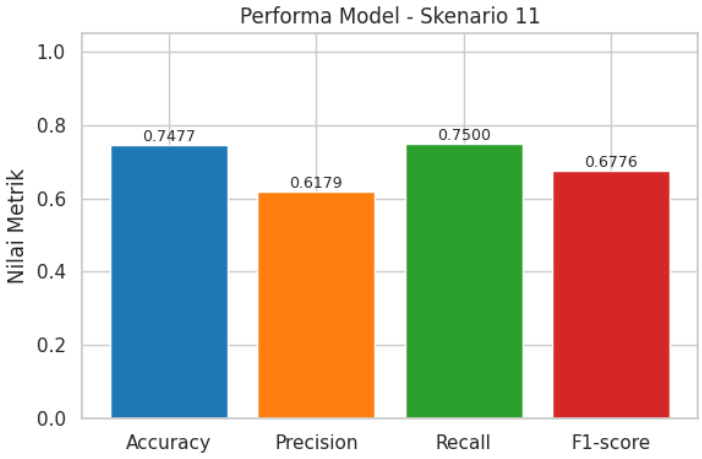
Model *final* dilatih menggunakan 3 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.11 setelah proses tuning selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 40% data uji terdiri dari 860 data yang telah disisihkan. Hasil *confusion matrix* pada model 11 untuk data *testing* ditampilkan pada Gambar 4.24.



Gambar 4.24 *Confusion Matrix* Model 11

Gambar 4.24 menampilkan *confusion matrix* yang merinci hasil prediksi model 11 pada 860 data uji. Berdasarkan *matrix* tersebut, model mampu memprediksi kelas dengan benar sebanyak 643 data, yang terdiri dari 415 *true negative* (pasien sehat terprediksi sehat) dan 228 *true positive* (pasien alzheimer terprediksi alzheimer). Namun, model ini menghasilkan 217 data yang salah prediksi, yang terdiri dari 141 *false positive* (pasien sehat diprediksi alzheimer) dan 76 *false negative* (pasien alzheimer diprediksi sehat). Jumlah kesalahan ini jauh lebih tinggi dibandingkan skenario-skenario sebelumnya.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai tersebut kemudian ditampilkan secara visual pada Gambar 4.25 untuk menunjukkan analisis kinerja model 11 secara lebih jelas.



Gambar 4.25 Performa Model 11

Gambar 4.25 menampilkan hasil performa model 11 yaitu menghasilkan akurasi sebesar 74,77%, presisi 61,79%, *recall* 75,00%, dan *f1-score* 67,76%. Nilai akurasi dan *f1-score* yang paling rendah di antara seluruh skenario menunjukkan bahwa kombinasi *threshold pearson* 0,25 dengan hanya tiga fitur terpilih dan *balancing ADASYN* tidak memberikan performa yang memadai. Rendahnya presisi mengindikasikan bahwa hampir empat dari sepuluh prediksi positif adalah keliru, sedangkan *recall* yang hanya 75% menunjukkan bahwa satu dari empat pasien Alzheimer masih berisiko tidak terdeteksi. Dengan demikian, model 11 dapat disimpulkan sebagai salah satu skenario dengan kinerja terburuk dalam penelitian ini dan kurang layak dipertimbangkan untuk implementasi pada sistem pendukung keputusan klinis.

4.2.12 Model 12

Model 12 menggunakan konfigurasi *preprocessing* yang sejajar dengan model 11 dari sisi pemilihan fitur, namun tanpa penerapan *balancing ADASYN*. Data dibagi menggunakan rasio *train–test split* 60:40 sehingga diperoleh 1.289 data

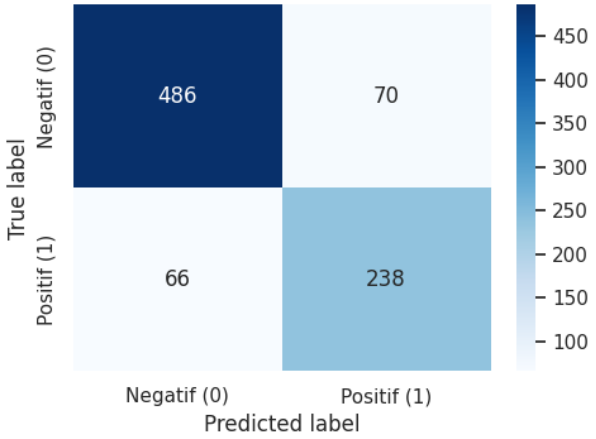
latih dan 860 data uji. Proses *feature selection pearson* dengan *threshold* 0,25 hanya meloloskan 3 fitur utama, sehingga model berlatih pada ruang fitur yang lebih sederhana dibanding skenario dengan *threshold* yang lebih rendah. Tidak digunakannya *ADASYN* membuat distribusi kelas pada data latih tetap mengikuti kondisi asli dataset, sehingga model belajar langsung dari pola ketidakseimbangan kelas yang ada.

Data latih yang tidak seimbang dan telah diseleksi 3 fitur tersebut kemudian digunakan untuk mencari *hyperparameter* optimal melalui *bayesian optimization*. Proses *hyperparameter tuning* ini mencari nilai terbaik dari 5 parameter kunci untuk memaksimalkan *f1-score*. Hasil *hyperparameter* terbaik yang ditemukan untuk model 12 ditampilkan pada Tabel 4.12.

Tabel 4.12 *Hyperparameter* Terbaik Pada Model 12

Hyperparameter	**Nilai *Bayesian***	**Nilai Parameter Terbaik**
n_estimators	[100 - 500]	100
max_depth	[3 - 10]	3
learning_rate	[0.01 - 0.2]	0.05
subsample	[0.7 - 1.0]	0.76
colsample_bytree	[0.7 - 1.0]	1.0

Kombinasi parameter ini menggambarkan model dengan pohon-pohon yang dangkal dan jumlah yang relatif sedikit, sehingga pembelajaran dilakukan secara lebih konservatif untuk menghindari *overfitting* pada jumlah fitur yang terbatas. Model *final* dilatih menggunakan 3 fitur terpilih dan *hyperparameter* terbaik dari Tabel 4.12 setelah proses *tuning* selesai. Model yang sudah dilatih ini kemudian dievaluasi kinerjanya menggunakan 40% data uji (860 data) yang telah disisihkan. Hasil *confusion matrix* pada model 12 untuk data *testing* ditampilkan pada Gambar 4.26.

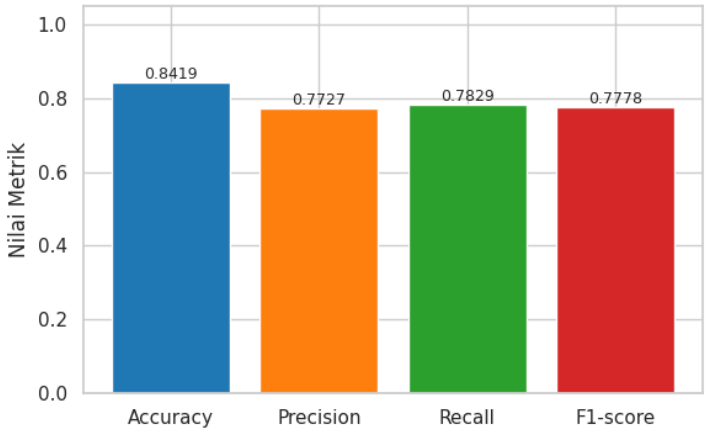


Gambar 4.26 *Confusion Matrix* Model 12

Gambar 4.26 menampilkan *confusion matrix* yang merinci hasil prediksi model 12 pada 860 data uji. Berdasarkan *matrix* tersebut, model mampu mengklasifikasikan dengan benar sebanyak 724 data, yang terdiri atas 486 *true negative* (pasien *Non*-Alzheimer terprediksi *Non*-Alzheimer) dan 238 *true positive* (pasien Alzheimer terprediksi Alzheimer). Di sisi lain, masih terdapat 70 *false positive* (pasien *Non*-Alzheimer diprediksi Alzheimer) dan 66 *false negative* (pasien Alzheimer diprediksi *Non*-Alzheimer). Dibandingkan model 11 yang menggunakan konfigurasi fitur yang sama namun dengan *balancing ADASYN*, jumlah kesalahan prediksi model 12 menurun cukup signifikan, baik pada *false positive* maupun *false negative*, yang menunjukkan bahwa dalam kondisi tiga fitur saja, *oversampling* justru tidak memberikan keuntungan.

Perhitungan metrik evaluasi Persamaan 3.10 hingga 3.13 yang berasal dari *confusion matrix* tersebut, menghasilkan nilai-nilai kuantitatif yang menggambarkan performa model. Nilai-nilai ini mencakup akurasi, presisi, *recall*, dan *f1-score*, yang penting untuk evaluasi komprehensif. Hasil dari nilai-nilai

tersebut kemudian ditampilkan secara visual pada Gambar 4.27 untuk menunjukkan analisis kinerja model 12 secara lebih jelas.



Gambar 4.27 Performa Model 12

Gambar 4.27 menampilkan hasil performa model 12 yaitu menghasilkan akurasi sebesar 84,19%, presisi 77,27%, *recall* 78,29%, dan *f1-score* 77,78%. Nilai-nilai ini menunjukkan adanya peningkatan performa yang cukup jelas dibanding model 11, terutama pada presisi dan *f1-score*, meskipun masih berada di bawah performa model-model lainnya. Presisi sebesar 77,27% mengindikasikan bahwa sekitar tiga perempat pasien yang diprediksi Alzheimer benar-benar positif, sementara *recall* sebesar 78,29% menunjukkan bahwa sekitar tiga perempat pasien Alzheimer berhasil dideteksi. Secara keseluruhan, model 12 memberikan performa menengah, lebih baik daripada skenario dengan tiga fitur dan *ADASYN* yang ada pada model 11) namun masih belum mampu bersaing dengan skenario yang memanfaatkan penggunaan 32 dan 5 fitur dengan konfigurasi *hyperparameter* yang lebih optimal.

4.3 Pembahasan

Bagian ini membahas pengaruh tiga komponen utama dalam rancangan skenario pengujian terhadap performa model klasifikasi Alzheimer, yaitu *balancing* data menggunakan *ADASYN*, *feature selection* berbasis korelasi *pearson*, serta variasi rasio pembagian data latih dan data uji. Analisis dilakukan dengan menggunakan nilai rata-rata akurasi, presisi, *recall*, dan *f1-score* dari keseluruhan 12 skenario pengujian. Gambar-gambar pada subbab ini menyajikan visualisasi berupa diagram batang, sedangkan tabel yang menyertainya merangkum nilai numerik masing-masing metrik. Melalui kombinasi grafik dan tabel tersebut, diperoleh gambaran yang lebih komprehensif mengenai konfigurasi mana yang paling menguntungkan, serta dampak yang muncul akibat perubahan setiap komponen.

4.3.1 Analisis Pengaruh *Balancing* Data Menggunakan ADASYN

Pada penelitian ini digunakan teknik *balancing* data ADASYN untuk mengatasi ketidakseimbangan kelas antara pasien Alzheimer dan *non*-Alzheimer. Ketidakseimbangan tersebut berpotensi membuat model lebih cenderung mengenali kelas mayoritas, sehingga kinerja pada kelas minoritas menjadi kurang optimal. Oleh karena itu, perlu dilakukan analisis khusus untuk melihat sejauh mana penggunaan ADASYN benar-benar berpengaruh terhadap kualitas prediksi model. ADASYN tidak sekadar menggandakan data yang sudah ada, melainkan menghasilkan sampel sintetis baru untuk kelas minoritas berdasarkan pola data kelas minoritas pada data latih. Data sintetis tersebut dibentuk dengan memanfaatkan kedekatan sampel minoritas terhadap tetangga terdekatnya,

sehingga nilai fitur pada data baru berada di sekitar pola minoritas yang sudah ada. Selain itu, ADASYN menambahkan lebih banyak sampel pada area yang dianggap sulit dipelajari, yaitu ketika sampel minoritas berada dekat dengan sampel kelas mayoritas. Pada penelitian ini, ADASYN diterapkan hanya pada data latih agar tidak terjadi kebocoran informasi dari data uji. Analisis ini dilakukan dengan membandingkan rata-rata nilai akurasi, presisi, *recall*, dan *f1-score* antara model yang dilatih dengan ADASYN dan model yang dilatih tanpa *balancing*. Ringkasan hasil perbandingan kedua konfigurasi tersebut ditampilkan pada Tabel 4.13.

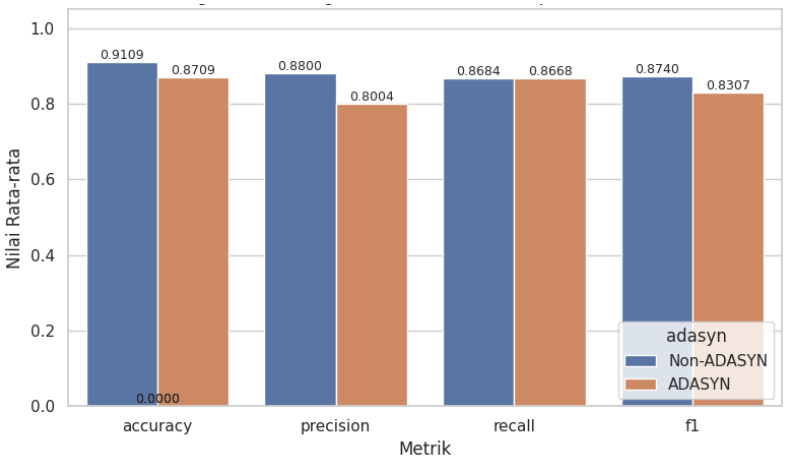
Tabel 4.13 Rata-Rata Metrik Evaluasi Berdasarkan Penggunaan *Balancing*

Balancing	**Akurasi**	**Presisi**	***Recall***	***F1-Score***
Non-ADASYN	0.910853	0.879962	0.868421	0.874048
ADASYN	0.870930	0.800397	0.866776	0.830672

Tabel 4.13 menunjukkan bahwa model tanpa *balancing* ADASYN memiliki performa yang lebih baik dibandingkan model yang menggunakan ADASYN pada seluruh metrik evaluasi. Pada konfigurasi *non*-ADASYN, rata-rata akurasi mencapai sekitar 91,09% dan nilai *f1-score* sekitar 87,40%. Sementara itu, pada model dengan ADASYN, akurasi turun menjadi kurang lebih 87,09% dan *f1-score* menjadi sekitar 83,07%. Penurunan yang paling jelas terlihat pada metrik presisi, dari sekitar 87,99% pada *non*-ADASYN menjadi hanya sekitar 80,04% saat ADASYN digunakan. Artinya, ketika ADASYN diterapkan, proporsi prediksi positif yang benar berkurang dan model lebih sering memberikan prediksi positif yang salah.

Nilai *recall* pada kedua konfigurasi relatif mendekati, yaitu sekitar 86,84% untuk *non*-ADASYN dan 86,68% untuk ADASYN. Hal ini menunjukkan bahwa

penggunaan ADASYN tidak memberikan peningkatan yang berarti terhadap kemampuan model dalam menemukan seluruh kasus Alzheimer. Kombinasi penurunan presisi yang cukup besar dan *recall* yang cenderung tetap mengindikasikan bahwa pada penelitian ini, penggunaan ADASYN membuat model menjadi lebih sensitif terhadap kelas positif Alzheimer. Akibatnya jumlah prediksi positif meningkat, tetapi peningkatan tersebut tidak diikuti kenaikan deteksi positif yang benar secara signifikan, sehingga prediksi positif yang salah menjadi lebih banyak dan presisi menurun. Dampak ini juga menjelaskan mengapa nilai *f1-score* ikut turun, karena *f1-score* dipengaruhi oleh keseimbangan antara presisi dan *recall*. Dengan kata lain, pada dataset ini ADASYN tidak memperbaiki kemampuan model dalam menangkap kasus Alzheimer, tetapi justru meningkatkan kesalahan prediksi positif.



Gambar 4. 28 Pengaruh *Balancing* Data Terhadap Metrik Evaluasi

Gambar 4.28 menampilkan diagram batang yang memvisualisasikan perbandingan rata-rata metrik evaluasi antara model *non*-ADASYN dan ADASYN. Pola yang terlihat pada grafik sejalan dengan Tabel 4.13, di mana batang untuk *non*-ADASYN selalu lebih tinggi pada setiap metrik, terutama pada presisi dan *f1-score*.

Visualisasi ini memperkuat kesimpulan bahwa *balancing* menggunakan ADASYN pada dataset ini tidak meningkatkan performa model, dan justru membuat hasil prediksi menjadi kurang akurat dibandingkan model yang dilatih pada distribusi data asli.

Berdasarkan hasil pengujian yang dirangkum pada Tabel 4.13, terlihat fenomena menarik di mana penerapan metode *balancing* ADASYN justru menghasilkan penurunan performa dibandingkan skenario tanpa *balancing* ADASYN. Secara spesifik, penggunaan ADASYN menyebabkan penurunan akurasi rata-rata dari 91,09% menjadi 87,09% dan penurunan presisi yang signifikan dari 87,99% menjadi 80,04%. Secara mekanisme, penurunan ini dapat terjadi karena ADASYN membuat sampel minoritas baru di area yang dekat dengan perbatasan kelas. Jika batas pemisah antar kelas pada data asli sebenarnya sudah cukup jelas, penambahan sampel sintetis di sekitar perbatasan dapat memperluas wilayah prediksi positif. Kondisi tersebut meningkatkan peluang model memprediksi Alzheimer pada data yang sebenarnya *non*-Alzheimer, sehingga kesalahan prediksi positif meningkat dan presisi menurun. Hal ini sejalan dengan pola metrik yang diperoleh, yaitu presisi turun cukup besar sementara *recall* tidak meningkat secara berarti.

Penerapan ADASYN pada kondisi data tersebut justru memicu terjadinya penyimpangan pola distribusi. ADASYN bekerja dengan membuat data sintetis secara adaptif di area yang dianggap sulit dipelajari, yang umumnya berada di dekat perbatasan kelas mayoritas dan minoritas. Pada dataset yang bersih, penambahan data sintetis ini justru menciptakan *noise* buatan yang mengaburkan batas pemisah

yang sebenarnya sudah jelas. Hal ini menyebabkan model melakukan generalisasi yang berlebihan, di mana area yang seharusnya menjadi wilayah kelas mayoritas (*non*-Alzheimer) ditempati oleh data sintetis kelas minoritas. Akibatnya, model menjadi bias dan cenderung melakukan kesalahan prediksi positif (*false positive*), sebagaimana ditunjukkan oleh penurunan nilai presisi yang drastis. Temuan ini sejalan dengan salah satu penelitian yang menyatakan bahwa pada dataset dengan kompleksitas tertentu, penggunaan teknik *oversampling* tidak selalu menjamin peningkatan performa dan justru berisiko merusak struktur data asli. Secara spesifik, hal ini dibuktikan dalam penelitian yang dilakukan oleh Haluska dkk (2022) yang menampilkan hasil penelitiannya pada lampiran B, khususnya Tabel VIII dan Tabel XVII, di mana terlihat bahwa penggunaan ADASYN justru menghasilkan nilai evaluasi yang lebih rendah dibandingkan dengan pendekatan *Baseline* tanpa penyeimbang data, dan bahkan pada Tabel XVII, ADASYN tertulis N/A. Ini artinya metode ADASYN gagal berjalan atau tidak menghasilkan output pada dataset *Click Prediction* V1.

4.3.2 Analisis Pengaruh *Feature Selection (Threshold Pearson)*

Pada tahap *feature selection*, nilai korelasi *pearson* digunakan sebagai acuan untuk menentukan fitur mana yang dipertahankan. Semakin tinggi nilai *threshold*, semakin sedikit fitur yang dianggap relevan dan digunakan dalam pelatihan model. Dalam penelitian ini diuji tiga nilai *threshold*, yaitu 0,0; 0,15; dan 0,25. Pengaruh masing-masing *threshold* terhadap rata-rata nilai *accuracy*, *precision*, *recall*, dan *f1-score* dirangkum pada Tabel 4.14.

Tabel 4.14 Analisis Pengaruh *Threshold Feature Selection*

***Threshold* Pearson**	**Akurasi**	**Presisi**	***Recall***	***F1-score***
0.0	0.944186	0.930543	0.910362	0.920249
0.15	0.935174	0.899039	0.922697	0.910197
0.25	0.793314	0.690958	0.769737	0.726634

Tabel 4.14 menunjukkan bahwa *threshold* 0,0 memberikan rata-rata nilai metrik tertinggi di antara ketiga skenario. Pada *threshold* ini, seluruh 32 fitur digunakan sehingga informasi yang tersedia bagi model masih sangat lengkap. Akurasi rata-rata mencapai sekitar 94,42% dengan *f1-score* sekitar 92,02%. Ketika *threshold* dinaikkan menjadi 0,15, jumlah fitur yang digunakan berkurang menjadi lima fitur utama. Nilai *accuracy* dan *f1-score* sedikit menurun menjadi sekitar 93,52% dan 91,02%, namun *recall* justru sedikit meningkat dibanding *threshold* 0,0 yang awalnya 91,03% menjadi 92,26%. Hal ini menunjukkan bahwa meskipun jumlah fitur lebih sedikit, fitur yang dipertahankan masih cukup representatif sehingga kinerja model tetap tinggi.

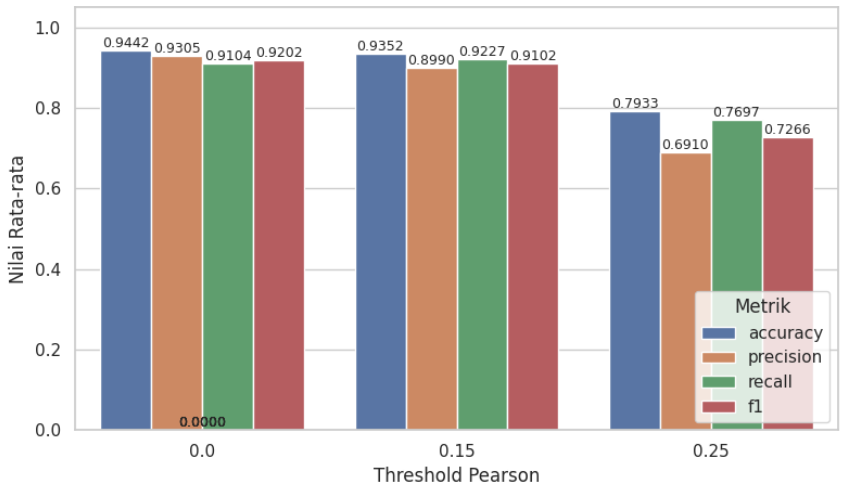
Tabel 4.15 Hasil *Feature Selection* Berdasarkan *Threshold Pearson*

***Threshold* Pearson**	**Jumlah Fitur**	**Fitur Terpilih**
0.0	32	*Age, Gender, Ethnicity, EducationLevel, BMI, Smoking, AlcoholConsumption, PhysicalActivity, DietQuality, SleepQuality, FamilyHistoryAlzheimers, CardiovascularDisease, Diabetes, Depression, HeadInjury, Hypertension, SystolicBP, DiastolicBP, CholesterolTotal, CholesterolLDL, CholesterolHDL, CholesterolTriglycerides, MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems, ADL, Confusion, Disorientation, PersonalityChanges, DifficultyCompletingTasks, Forgetfulness*
0.15	5	*MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems, ADL*
0.25	3	*FunctionalAssessment, MemoryComplaints, ADL*

Tabel 4.15 menunjukkan daftar fitur yang dipertahankan pada setiap nilai *threshold pearson*. Pada *threshold* 0,0 seluruh fitur digunakan sehingga model

menerima informasi yang paling lengkap, yaitu sebanyak 32 fitur. Ketika *threshold* dinaikkan menjadi 0,15, jumlah fitur berkurang menjadi lima fitur utama yang memiliki hubungan korelasi lebih kuat terhadap variabel Diagnosis, yaitu *MMSE, FunctionalAssessment, MemoryComplaints, BehavioralProblems*, dan *ADL*. Pada *threshold* 0,25 seleksi menjadi lebih ketat sehingga hanya tiga fitur yang tersisa, yaitu *FunctionalAssessment, MemoryComplaints*, dan *ADL*, yang menandakan bahwa ketiga fitur tersebut merupakan fitur dengan korelasi paling dominan terhadap target pada dataset ini.

Perubahan yang paling jelas terlihat ketika *threshold* dinaikkan menjadi 0,25, di mana hanya tiga fitur yang digunakan. Pada kondisi ini, seluruh metrik mengalami penurunan yang cukup besar. Akurasi turun dari sekitar 93–94% menjadi 79,33%, presisi turun menjadi 69,10%, *recall* menjadi 76,97%, dan *f1-score* menjadi 72,66%. Penurunan ini mengindikasikan bahwa *threshold* yang terlalu tinggi membuat beberapa fitur penting terbuang, sehingga informasi yang digunakan model menjadi kurang lengkap dan kemampuan klasifikasinya menurun.



Gambar 4. 29 Pengaruh *Threshold Pearson* Terhadap Metrik Evaluasi

Gambar 4.29 menampilkan diagram batang yang memvisualisasikan rata-rata *accuracy*, *precision*, *recall*, dan *f1-score* untuk setiap nilai *threshold pearson*. Pola pada grafik sejalan dengan Tabel 4.14, di mana batang untuk *threshold* 0,0 dan 0,15 tampak relatif berdekatan, sedangkan seluruh metrik menurun cukup tajam pada *threshold* 0,25. Visualisasi ini memperkuat kesimpulan bahwa pengurangan fitur yang terlalu agresif melalui *threshold* yang tinggi tidak menguntungkan, karena justru menurunkan kemampuan model dalam membedakan antara pasien Alzheimer dan *Non*-Alzheimer.

Keunggulan *threshold* 0.0 dibandingkan 0.15 dapat dijelaskan oleh karakteristik algoritma XGBoost itu sendiri. Metode korelasi *pearson* hanya mampu mengukur hubungan linear antar variabel, padahal data medis seringkali mengandung pola hubungan non-linear yang kompleks. Membatasi fitur berdasarkan filter linear berpotensi menghilangkan informasi tersembunyi yang sebenarnya dapat dimanfaatkan oleh XGBoost. Hal ini sejalan dengan penelitian Neyra dkk(2024), yang membuktikan bahwa kemampuan seleksi fitur internal atau *embedded feature selection* pada algoritma berbasis *tree* seperti XGBoost seringkali lebih efektif menangkap interaksi antar-fitur dibandingkan metode seleksi eksternal sederhana, sehingga penggunaan seluruh fitur cenderung memberikan hasil yang lebih optimal.

Sebaliknya, penurunan performa yang drastis pada *threshold* 0.25 disebabkan oleh terjadinya *Information Loss* atau hilangnya informasi klinis yang krusial. Penerapan *threshold* yang terlalu ketat menyebabkan tereliminasinya fitur-fitur penting, salah satunya adalah skor MMSE (*Mini-Mental State Examination*).

Padahal, dalam diagnosis medis, skor MMSE merupakan indikator standar emas untuk mendeteksi gangguan kognitif. Penghilangan fitur ini membuat model kehilangan variabel penting utamanya, sehingga kemampuan klasifikasinya menurun tajam. Penemuan ini didukung oleh studi medis yang menegaskan bahwa skor MMSE memiliki korelasi diagnostik yang sangat kuat terhadap demensia, sehingga keberadaannya dalam dataset sangat menentukan akurasi model prediksi (Jung et al., 2025). Oleh karena itu, mempertahankan kelengkapan fitur terbukti lebih menguntungkan daripada melakukan pengurangan fitur yang terlalu agresif.

4.3.3 Analisis Pengaruh Rasio *Split* Data

Pada proses pelatihan model, pemilihan rasio pemisahan data latih dan data uji berpengaruh terhadap jumlah data yang digunakan untuk belajar dan jumlah data yang digunakan untuk evaluasi. Rasio yang lebih besar pada data latih memungkinkan model mempelajari lebih banyak pola, sedangkan rasio data uji yang lebih besar memberikan gambaran evaluasi yang lebih luas. Dalam penelitian ini dibandingkan dua rasio pemisahan, yaitu 60:40 dan 80:20, untuk melihat pengaruhnya terhadap rata-rata nilai *accuracy*, *precision*, *recall*, dan *f1-Score*. Ringkasan hasil perbandingan kedua rasio tersebut ditunjukkan pada Tabel 4.16.

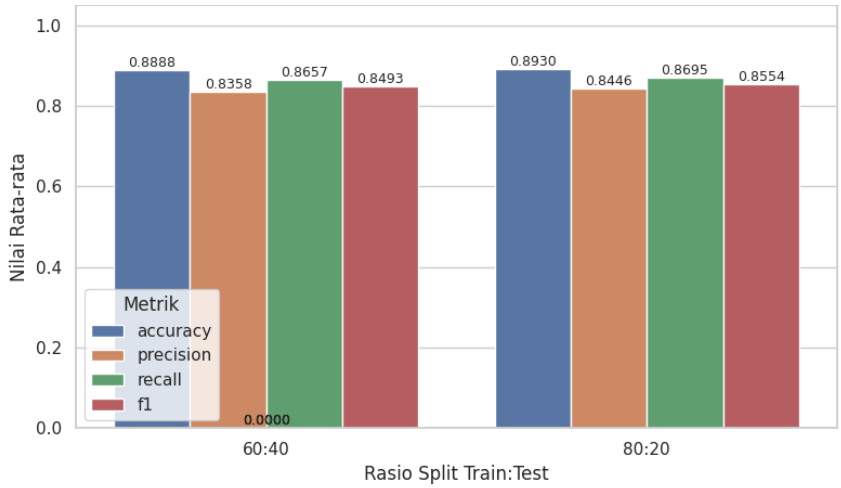
Tabel 4.16 Analisis Pengaruh Rasio *Split* Data

Rasio Split (Latih:Uji)	Akurasi	Presisi	*Recall*	*F1-score*
60:40	0.888760	0.83577	0.865680	0.849327
80:20	0.893023	0.84459	0.869518	0.855393

Hasil pada Tabel 4.16 menunjukkan bahwa rasio 80:20 memberikan nilai rata-rata metrik evaluasi yang sedikit lebih tinggi dibandingkan rasio 60:40. Pada rasio 60:40, akurasi rata-rata berada pada kisaran 88,88% dengan *f1-score* sekitar

84,93%. Sementara itu, pada rasio 80:20, akurasi meningkat menjadi sekitar 89,30% dan *f1-score* menjadi kurang lebih 85,54%. Hal yang sama terlihat pada presisi dan *recall*, yang masing-masing naik dari sekitar 83,58% menjadi 84,46% dan dari 86,57% menjadi 86,95% ketika digunakan rasio 80:20.

Kenaikan nilai metrik tersebut menunjukkan bahwa penambahan porsi data latih pada rasio 80:20 memberikan sedikit keuntungan bagi model. Dengan jumlah data latih yang lebih besar, model dapat mempelajari variasi pola yang lebih beragam sehingga menghasilkan prediksi yang lebih baik. Sebaliknya, pada rasio 60:40 jumlah data uji memang lebih banyak, tetapi pengurangan data latih menyebabkan performa model sedikit menurun. Perbedaan nilai metrik tidak terlalu besar, namun cukup untuk menunjukkan bahwa rasio 80:20 lebih menguntungkan bagi model pada dataset ini.



Gambar 4. 30 Pengaruh Rasio *Split* Data Terhadap Metrik Evaluasi

Gambar 4.30 menampilkan diagram batang yang memvisualisasikan rata-rata *accuracy*, *precision*, *recall*, dan *f1-Score* untuk rasio 60:40 dan 80:20. Pola pada grafik mendukung informasi pada Tabel 4.15, di mana batang untuk rasio

80:20 tampak sedikit lebih tinggi pada semua metrik dibandingkan rasio 60:40. Visualisasi ini menguatkan kesimpulan bahwa penggunaan rasio 80:20 memberikan kombinasi performa yang lebih baik, sehingga rasio tersebut dapat direkomendasikan sebagai pilihan yang lebih sesuai untuk penelitian ini.

Penelitian ini mendapatkan model terbaik diperoleh menggunakan rasio pembagian data 60:40. Keberhasilan model mencapai akurasi tertinggi yaitu 95,58% dengan hanya menggunakan 60% data latih membuktikan bahwa algoritma XGBoost memiliki efisiensi pembelajaran yang tinggi pada dataset ini. Jumlah sampel pada porsi 60% terbukti telah mencapai titik saturasi informasi, di mana model sudah berhasil menangkap seluruh pola esensial dari fitur-fitur terpilih tanpa memerlukan tambahan data latih.

Temuan ini menunjukkan berlakunya prinsip *diminishing returns* atau hasil yang semakin berkurang, di mana penambahan data latih menjadi 80% tidak lagi memberikan informasi baru yang signifikan untuk memperbaiki model, tetapi justru berpotensi meningkatkan risiko model menghafal *noise* atau *overfitting*. Hal ini sejalan dengan penelitian yang dilakukan oleh Barry-Straume dkk(2019) yang membuktikan bahwa akurasi validasi model tidak selalu berbanding lurus dengan besarnya data latih secara linear. Dalam studi mereka, penggunaan sebagian kecil data yaitu penggunaan 40% terbukti sudah cukup untuk mencapai tingkat akurasi yang tinggi dengan nilai 90%, dan penambahan data latih setelah titik optimal tersebut memberikan peningkatan performa yang sangat minim. Oleh karena itu, penggunaan rasio 80:20 dalam penelitian ini tidak memberikan dampak yang besar.

4.3.4 Analisis *Hyperparameter Tuning*

Pada penelitian ini, proses *hyperparameter tuning* dilakukan menggunakan pendekatan *bayesian optimization* untuk mencari kombinasi pengaturan terbaik pada setiap skenario. Lima *hyperparameter* utama yang diatur adalah jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), laju pembelajaran (*learning_rate*), proporsi sampel per-pohon (*subsample*), dan proporsi fitur per-pohon (*colsample_bytree*). Ringkasan nilai *hyperparameter* terbaik beserta akurasi yang diperoleh pada masing-masing skenario ditampilkan pada Tabel 4.16.

Berdasarkan Tabel 4.17, terlihat bahwa skenario dengan akurasi tertinggi adalah skenario 10 dengan nilai 95,58%, diikuti oleh skenario 2 dan 4 yang masing-masing mencapai 95,12%. Ketiga skenario tersebut menggunakan model tanpa *balancing* ADASYN dan sama-sama memanfaatkan *learning_rate* yang kecil (0,01) dengan jumlah pohon yang cukup banyak, yaitu antara 421 hingga 500 pohon. Pola ini menunjukkan bahwa penggunaan laju pembelajaran yang konservatif dan jumlah pohon yang besar membantu model belajar secara lebih bertahap, sehingga menghasilkan akurasi yang tinggi dan stabil. Skenario 1 dan 8 yang juga memiliki akurasi di atas 94% memperlihatkan kecenderungan serupa, meskipun menggunakan variasi kedalaman pohon dan nilai *subsample* yang sedikit berbeda.

Sebaliknya, skenario dengan akurasi terendah adalah skenario 5, 6, 11, dan 12 yang seluruhnya berada pada kelompok *threshold pearson* 0,25. Akurasi pada skenario-skenario tersebut hanya berkisar antara 74,77% sampai 84,19%, meskipun *hyperparameter* sudah dioptimasi. Hal ini mengindikasikan bahwa penurunan

jumlah fitur akibat *threshold* yang terlalu tinggi memiliki dampak yang lebih besar terhadap performa model dibandingkan variasi nilai *hyperparameter*. Dengan kata lain, konfigurasi *hyperparameter* yang baik tidak dapat sepenuhnya menutupi hilangnya informasi penting pada tahap seleksi fitur.

Jika dilihat dari sisi *balancing*, pasangan skenario dengan dan tanpa ADASYN yang memiliki konfigurasi lain serupa juga menunjukkan pola yang konsisten. Pada skenario 1 dan 2, 3 dan 4, 7 dan 8, serta 9 dan 10, model tanpa ADASYN selalu memiliki akurasi yang sedikit lebih tinggi dibandingkan model dengan ADASYN. Selain itu, skenario dengan ADASYN cenderung menggunakan kedalaman pohon yang lebih besar atau *learning_rate* yang lebih tinggi, seperti pada skenario 7 yaitu menggunakan *learning_rate* 0,1020 dan skenario 9 menggunakan *learning_rate* 0,0710, hal tersebut dilakukan untuk menyesuaikan diri dengan distribusi data sintetis yang lebih beragam. Meskipun demikian, penyesuaian ini belum mampu melampaui kinerja model *Non*-ADASYN yang menggunakan struktur pohon lebih sederhana namun dilatih pada data asli.

Hyperparameter tuning berperan penting sebagai langkah penyempurnaan konfigurasi model XGBoost. Penggunaan *learning_rate* yang kecil dan jumlah pohon yang memadai terbukti mendukung tercapainya akurasi yang tinggi, terutama pada skenario tanpa ADASYN dan dengan jumlah fitur yang lebih banyak. Namun demikian, kualitas data masukan, keseimbangan kelas, dan kelengkapan fitur tetap menjadi faktor utama yang menentukan performa model, sehingga *hyperparameter tuning* tidak dapat dilepaskan dari strategi pemilihan fitur dan *preprocessing* yang dirancang dengan baik.

Pada model 10, algoritma memilih strategi belajar yang sangat teliti dan hati-hati, atau bisa disebut sebagai *slow learning*. Hal ini terlihat dari kombinasi jumlah pohon yang dimaksimalkan hingga 500, namun dibarengi dengan laju pembelajaran (*learning rate*) yang sangat kecil, yaitu 0.01. Selain itu, model ini hanya menggunakan pohon yang dangkal (max_depth 3). Kombinasi ini mengindikasikan bahwa pola data asli sebenarnya sudah cukup jelas dan rapi. Model tidak perlu membuat aturan keputusan yang rumit untuk membedakan pasien sehat dan sakit, cukup dengan struktur sederhana yang dipelajari secara perlahan dan berulang-ulang, model justru bisa mencapai akurasi tertinggi 95,58% tanpa terjebak menghafal *noise*.

Kondisi ini berbanding terbalik dengan model 1. Di sini, algoritma seolah dipaksa bekerja lebih keras dan agresif. Model harus menggunakan struktur pohon yang jauh lebih dalam (*max_depth* 7) dan laju pembelajaran yang lebih besar 0.03). Peningkatan kompleksitas ini terjadi karena ADASYN menciptakan data buatan di area perbatasan kelas, yang membuat batas pemisah antara sehat dan sakit menjadi tidak beraturan. Untuk bisa memilah data yang sudah bercampur dengan data sintetis tersebut, XGBoost harus membangun logika percabangan yang rumit dan dalam. Sayangnya, karena model terlalu sibuk mempelajari kerumitan data buatan tersebut, kemampuannya dalam memprediksi data uji yang sebenarnya justru sedikit menurun menjadi 94,65%. Hal ini membuktikan bahwa pada kasus ini, data asli yang dibiarkan apa adanya lebih mudah dipelajari dan menghasilkan model yang lebih efektif dibandingkan data yang dimanipulasi dengan ADASYN.

Selain membandingkan akurasi antar skenario, Tabel 4.17 juga menampilkan bagaimana kombinasi *hyperparameter* yang dipilih berusaha menyesuaikan diri dengan kondisi data pada tiap skenario. Nilai *n_estimators* yang cenderung tinggi pada beberapa skenario menunjukkan bahwa model memerlukan lebih banyak iterasi pohon untuk mempelajari pola secara bertahap, terutama ketika *learning_rate* dibuat kecil agar proses belajar lebih stabil dan tidak terlalu cepat mengikuti variasi acak pada data. Sementara itu, variasi *max_depth* mencerminkan tingkat kompleksitas aturan keputusan yang dibangun, di mana kedalaman yang lebih besar biasanya dipakai ketika pola data lebih sulit dipisahkan, sedangkan kedalaman yang lebih kecil cukup ketika batas pemisah antar kelas sudah relatif jelas. Dengan demikian, tabel ini memberi gambaran bahwa performa terbaik muncul ketika pengaturan *hyperparameter* selaras dengan karakteristik data, terutama terkait jumlah fitur yang digunakan dan ada atau tidaknya *balancing*.

Selain digunakan untuk memperoleh konfigurasi terbaik, ringkasan pada Tabel 4.17 juga membantu melihat pola pengaturan *hyperparameter* yang cocok untuk karakteristik data pada tiap skenario. Setiap baris pada tabel merepresentasikan hasil pencarian kombinasi parameter optimal dari *bayesian optimization*, sehingga nilai yang tampil bukan dipilih secara manual, melainkan diputuskan berdasarkan performa terbaik selama proses optimasi. Oleh karena itu, tabel ini bukan hanya menampilkan angka akurasi, tetapi juga menunjukkan bagaimana XGBoost mengatur kompleksitas model melalui *max_depth*, intensitas pembelajaran melalui *learning_rate*, serta jumlah iterasi pembentukan pohon melalui *n_estimators* untuk mencapai hasil terbaik pada kondisi data yang berbeda.

Dengan membaca tabel ini, dapat dianalisis apakah peningkatan performa lebih banyak dipengaruhi oleh strategi belajar perlahan dengan banyak pohon atau justru oleh model yang lebih sederhana namun stabil karena dilatih pada distribusi data yang lebih jelas.

Tabel 4.17 Ringkasan *Hyperparameter Tuning* terhadap Akurasi

Skenario	***Balancing***	***Hyperparameter Tuning***	**Akurasi**
1	ADASYN	*n_estimators :* 386 *max_depth :* 7 *learning_rate :* 0.037144 *subsample :* 0.971156 *colsample_bytree :* 0.939866	94.65%
2	*Non*-ADASYN	*n_estimators :* 421 *max_depth :* 7 *learning_rate :* 0.016734 *subsample :* 0.856916 *colsample_bytree :* 0.943719	95.12%
3	ADASYN	*n_estimators :* 500 *max_depth :* 10 *learning_rate :* 0.010000 *subsample :* 1.000000 *colsample_bytree :* 1.000000	92.56%
4	*Non*-ADASYN	*n_estimators :* 500 *max_depth :* 3 *learning_rate :* 0.010000 *subsample :* 0.745184 *colsample_bytree :* 1.000000	95.12%
5	ADASYN	*n_estimators :* 487 *max_depth :* 6 *learning_rate :* 0.030749 *subsample :* 0.700000 *colsample_bytree :* 1.000000	76.05%
6	*Non*-ADASYN	*n_estimators :* 391 *max_depth :* 3 *learning_rate :* 0.010000 *subsample :* 1.000000 *colsample_bytree :* 1.000000	82.33%
7	ADASYN	*n_estimators :* 450 *max_depth :* 6 *learning_rate :* 0.102040 *subsample :* 0.877786 *colsample_bytree :* 0.885124	93.72%
8	*Non*-ADASYN	*n_estimators :* 458 *max_depth :* 10 *learning_rate :* 0.013146 *subsample :* 0.987064 *colsample_bytree :* 0.928859	94.19%
9	ADASYN	*n_estimators :* 181 *max_depth :* 10 *learning_rate :* 0.070921	90.81%

Skenario	*Balancing*	*Hyperparameter Tuning*	Akurasi
		subsample : 0.927599 *colsample_bytree :* 1.000000	
10	*Non*-ADASYN	*n_estimators :* 500 *max_depth :* 3 *learning_rate :* 0.010000 *subsample :* 0.924930 *colsample_bytree :* 0.700000	95.58%
11	ADASYN	*n_estimators :* 455 *max_depth :* 8 *learning_rate :* 0.017568 *subsample :* 0.820947 *colsample_bytree :* 0.887408	74.77%
12	*Non*-ADASYN	*n_estimators :* 100 *max_depth :* 3 *learning_rate :* 0.050325 *subsample :* 0.756051 *colsample_bytree :* 1.000000	84.19%

4.3.5 Penentuan Skenario Model Terbaik

Penentuan skenario model terbaik pada penelitian ini dilakukan dengan membandingkan nilai akurasi pada data uji dari seluruh 12 skenario yang telah diujikan secara komprehensif. Tabel 4.18 menyajikan ringkasan konfigurasi setiap skenario, yang mencakup metode *balancing* ADASYN atau *Non*-ADASYN, rasio *train–test split*, nilai *threshold pearson* untuk *feature selection*, serta capaian nilai metrik evaluasi. Berdasarkan tabel tersebut, terlihat bahwa nilai akurasi tertinggi secara keseluruhan diperoleh pada skenario 10 dengan akurasi sebesar 95,58%, sedangkan pada kelompok skenario dengan *balancing* ADASYN, akurasi tertinggi dicapai oleh skenario 1 dengan nilai 94,65%.

Berdasarkan Tabel 4.18, terlihat bahwa performa tertinggi secara umum dicapai oleh skenario 10 yang menggunakan rasio split 60:40, tanpa *balancing* ADASYN, dan *threshold pearson* 0,15. Skenario ini menghasilkan akurasi 95,58%, presisi 94,63%, *recall* 92,76%, dan *f1-score* 93,69%, sehingga dipilih sebagai

model terbaik tanpa ADASYN. Sementara itu, di antara skenario yang menggunakan ADASYN, kinerja terbaik diperoleh pada skenario 1 dengan akurasi 94,65%, presisi 93,88%, *recall* 90,79%, dan *f1-score* 92,31%. Oleh karena itu, skenario 1 dan skenario 10 diberi penanda warna kuning pada tabel karena masing-masing mewakili model terbaik untuk pendekatan dengan *balancing* ADASYN dan tanpa *balancing*, serta menjadi model utama yang merepresentasikan hasil terbaik dalam penelitian ini.

Tabel 4. 18 Hasil Pengujian 12 Skenario

Model	Split	Metode *Balancing*	*Threshold pearson*	Akurasi	Presisi	*Recall*	*F1-score*
1	80 : 20	ADASYN	0,00	94,65 %	93,88 %	90,79 %	92,31 %
2		*Non*-ADASYN	0,00	95,12 %	94,56 %	91,45 %	92,98 %
3		ADASYN	0,15	92,56 %	87,04 %	92,76 %	89,81 %
4		*Non*-ADASYN	0,15	95,12 %	93,96 %	92,11 %	93,02 %
5		ADASYN	0,25	76,05 %	62,96 %	78,29 %	69,79 %
6		*Non*-ADASYN	0,25	82,33 %	74,36 %	76,32 %	75,32 %
7	60 : 40	ADASYN	0,00	93,72 %	90,58 %	91,78 %	91,18 %
8		*Non*-ADASYN	0,00	94,19 %	93,20 %	90,13 %	91,64 %
9		ADASYN	0,15	90,81 %	83,99 %	91,45 %	87,56 %
10		*Non*-ADASYN	0,15	95,58 %	94,63 %	92,76 %	93,69 %
11		ADASYN	0,25	74,77 %	61,79 %	75,00 %	67,76 %
12		*Non*-ADASYN	0,25	84,19 %	77,27 %	78,29 %	77,78 %

4.4 Perbandingan dengan Penelitian Terdahulu

Untuk memvalidasi kontribusi penelitian ini, akan dilakukan perbandingan hasil model terbaik yang dilakulan penelitian ini dengan penelitian terkait yang dilakukan oleh Yahya, dkk (2025) yang juga menggunakan algoritma XGBoost pada dataset Alzheimer yang sama. Tabel 4.19 berikut menampilkan perbandingan

secara lengkap, mencakup perbedaan metodologi dan seluruh nilai dari metrik evaluasi.

Tabel 4. 19 Perbandingan Hasil Penelitian dengan Penelitian Terdahulu

Aspek Perbandingan	Penelitian Terdahulu	Penelitian Ini	Analisis Perbedaan
Metode	*XGBoost Classifier*	*XGBoost Classifier*	Metode yang digunakan sama
Strategi *Balancing*	SMOTE (*Synthetic Minority Oversampling Technique*)	Tanpa Balancing (Non-ADASYN)	Yahya menggunakan data sintetik (SMOTE) untuk menaikkan performa. Penelitian ini membuktikan bahwa data asli, justru lebih akurat karena menghindari noise pada area overlap.
Metode Optimasi	*Grid Search*	*Bayesian Optimization*	Penelitian ini menerapkan metode optimasi yang lebih efisien dan adaptif untuk menemukan parameter internal terbaik.
Seleksi Fitur	Manual (berdasarkan EDA)	Korelasi *Pearson* (*Threshold* 0.15)	Penelitian ini menggunakan pendekatan statistik yang terukur untuk memilih 5 fitur klinis paling relevan.
Rasio Data Latih	80%	60%	Penelitian ini mencapai performa tinggi dengan data latih yang jauh lebih sedikit dengan 60% membuktikan efisiensi pembelajaran model yang dibangun.
Akurasi	95,3%	95,58%	Unggul (+0,63%). Model penelitian ini lebih handal dalam meminimalkan kesalahan diagnosis pada orang sehat (*false positive*).
Presisi	94%	94,63%	Yahya unggul karena SMOTE secara agresif memaksa model mengenali kelas positif, namun dengan mengorbankan sedikit presisi.
Recall	96,1%	92,76%	Yahya unggul tipis karena nilai recall yang sangat tinggi mendongkrak rata-rata harmonis.
F1-score	95%	93,69%	Yahya unggul tipis karena nilai r*ecall* yang sangat tinggi mendongkrak rata-rata harmonis.

Tabel 4.19 menampilkan perbandingan kinerja antara model terbaik dalam penelitian ini, yaitu skenario 10 (tanpa ADASYN dengan rasio split 60:40, dan model yang dikembangkan oleh Yahya, dkk (2025). Kedua penelitian sama-sama menggunakan algoritma XGBoost dan dataset yang serupa, namun terdapat beberapa perbedaan penting dalam pendekatan yang digunakan sehingga menghasilkan karakteristik kinerja yang berbeda.

Penelitian ini memperoleh nilai akurasi sebesar 95,58% dan presisi 94,63%, sedikit lebih tinggi dibandingkan hasil penelitian sebelumnya, yang mendapatkan akurasi 95,3% dan presisi 94,0%. Peningkatan pada presisi menunjukkan bahwa model yang dikembangkan dalam penelitian ini lebih tepat dalam memberikan prediksi positif, sehingga jumlah kasus pasien sehat yang terklasifikasi sebagai Alzheimer dapat ditekan. Salah satu faktor yang berkontribusi terhadap hal ini adalah pemilihan skenario tanpa teknik *balancing* tambahan, sehingga model belajar langsung dari distribusi data asli tanpa penambahan data sintetik yang berpotensi menambah gangguan pada pola data.

Di sisi lain, Yahya, dkk (2025) mendapatkan nilai *recall* yang lebih tinggi, yaitu 96,1%, sedangkan penelitian ini menghasilkan *recall* 92,76%. Perbedaan ini sejalan dengan penggunaan metode SMOTE pada penelitian terdahulu yang secara khusus dirancang untuk meningkatkan jumlah sampel pada kelas minoritas. Pendekatan tersebut cenderung meningkatkan kemampuan model dalam menangkap sebanyak mungkin kasus positif, namun dengan konsekuensi meningkatnya risiko kesalahan prediksi pada kelas lain. Dengan demikian, model

pada penelitian ini menawarkan kinerja yang lebih seimbang antara presisi dan *recall*, walaupun nilai *recall*nya sedikit lebih rendah.

Perbedaan lain yang cukup penting terletak pada strategi penyetelan *hyperparameter*. Penelitian sebelumnya menggunakan metode *grid search* yang menguji kombinasi parameter secara menyeluruh dalam ruang pencarian yang telah ditentukan. Pendekatan ini bersifat *brute force*, yaitu dengan mencoba hampir semua kombinasi parameter yang mungkin tanpa memanfaatkan informasi dari hasil percobaan sebelumnya, sehingga membutuhkan lebih banyak waktu dan sumber daya komputasi. Sementara itu, penelitian ini menggunakan *bayesian optimization* yang mengarahkan pencarian parameter secara bertahap berdasarkan informasi dari evaluasi sebelumnya. Penelitian yang dilakukan oleh Hidayaturrohman & Hanada (2025) mendapatkan bahwa pendekatan berbasis *bayesian* umumnya lebih efisien dibandingkan metode pencarian standar, karena mampu menemukan kombinasi *parameter* yang baik dengan jumlah percobaan yang lebih sedikit. Hal ini sejalan dengan hasil penelitian ini, di mana model terbaik dapat diperoleh tanpa harus mengeksplorasi seluruh kombinasi secara *brute force*.

Keunggulan pendekatan tersebut juga terlihat dari efisiensi penggunaan data latih. Model terbaik dalam penelitian ini diperoleh pada rasio split 60:40, sehingga hanya 60 persen data yang digunakan untuk pelatihan. Meskipun demikian, performa yang dihasilkan tetap bersaing dengan penelitian terdahulu yang menggunakan porsi data latih lebih besar. Hal ini menunjukkan bahwa kualitas pemilihan fitur melalui korelasi *pearson* dan ketepatan pengaturan *hyperparameter* melalui *bayesian optimization* mempunyai peran yang sangat penting. Dengan kata

lain, tidak hanya jumlah data latih yang menentukan keberhasilan model, tetapi juga bagaimana data tersebut diproses dan bagaimana parameter internal model diatur secara sistematis.

4.5 Evaluasi *K-Fold Cross Validation*

Evaluasi menggunakan 5-*Fold Cross Validation* dilakukan untuk menilai konsistensi performa model pada dua skenario terbaik dari masing-masing kelompok, yaitu skenario 1 yang menggunakan ADASYN dengan *threshold pearson* 0,0 sehingga memakai 32 fitur, serta skenario 10 yang tidak menggunakan ADASYN dengan *threshold pearson* 0,15 sehingga memakai 5 fitur. Pada setiap *fold*, data dibagi menjadi data latih dan data uji secara seimbang proporsi kelasnya, lalu model diuji menggunakan metrik akurasi, presisi, *recall*, dan *f1-score*. Hasil *K-Fold* ini penting karena menampilkan performa model pada beberapa pembagian data yang berbeda, sehingga lebih menggambarkan kestabilan model ketika diterapkan pada data baru. Selain menampilkan hasil per *fold*, evaluasi juga ditambahkan dalam bentuk rata-rata performa per kelas serta *boxplot* untuk memudahkan melihat kestabilan performa antar *fold*.

Pada bagian evaluasi ini, beberapa sel pada tabel diberi *highlight* warna kuning untuk menandai nilai ringkasan hasil 5-*Fold Cross Validation*, yaitu nilai rata-rata (*mean*) dan standar deviasi (std). Pada tabel evaluasi per *fold*, pada baris Rata-rata (std) di-*highlight* karena menunjukkan performa umum model sekaligus tingkat variasinya antar *fold*. Sementara itu, pada tabel performa per kelas, bagian Rata-rata *Macro Average* (std) di-*highlight* karena merepresentasikan rata-rata performa dua kelas yaitu Alzheimer dan *non*-Alzheimer beserta penyebarannya,

sehingga memudahkan pembaca melihat kestabilan model secara keseluruhan sejak awal pembahasan.

4.5.1 Evaluasi 5-*Fold Cross Validation* Skenario 1

Berdasarkan Tabel 4.20, performa model pada skenario 1 menunjukkan bahwa kelas *non*-Alzheimer (0) memiliki nilai yang lebih tinggi dibanding kelas Alzheimer (1). Pada kelas Alzheimer, nilai presisi sebesar 93,20%, *recall* sebesar 91,97%, dan *f1-score* sebesar 92,58%, yang berarti model sudah cukup baik mengenali kasus Alzheimer walaupun masih ada sebagian kecil kasus positif yang terlewat. Sementara itu, pada kelas *non*-Alzheimer, nilai presisi sebesar 95,65%, *recall* sebesar 96,33%, dan *f1-score* sebesar 95,98%, yang menunjukkan model sangat kuat dalam mengenali kelas negatif. Nilai *Macro Average* yang berada di sekitar 94% mengindikasikan bahwa jika dilihat secara rata untuk dua kelas, performa model cenderung tinggi dan seimbang.

Tabel 4.20 Rata-Rata Performa Klasifikasi per Kelas 5-*Fold Cross Validation* Pada Skenario 1

Kelas	Presisi	*Recall*	*F1-score*
Alzheimer (1)	93,20% (0,54%)	91,97% (1,88%)	92,58% (1,07%)
Non-ALZHEIMER (0)	95,65% (0,98%)	96,33% (0,30%)	95,98% (0,53%)
Rata-rata *Macro Average* (std)	94,42% (0,63%)	94,15% (0,97%)	94,28% (0,80%)

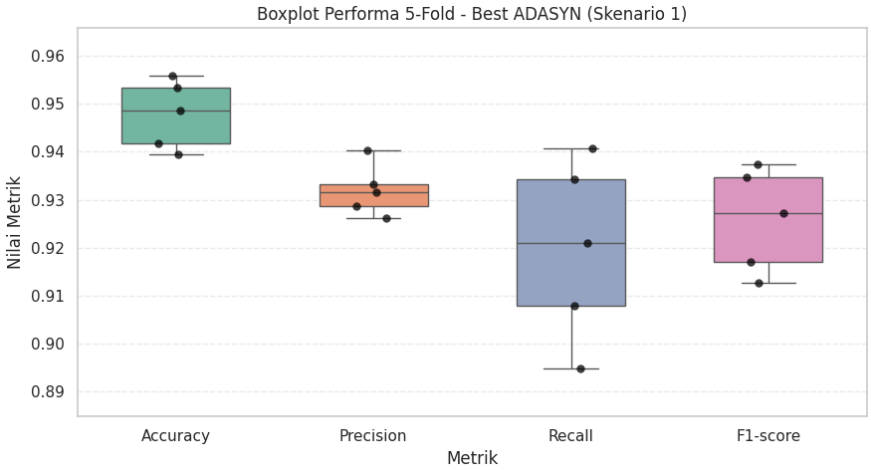
Berdasarkan Tabel 4.21, hasil evaluasi per *fold* pada skenario 1 juga menunjukkan performa yang konsisten. Nilai akurasi rata-rata sebesar 94,79% dengan variasi yang kecil, dan rentang akurasi per *fold* berada pada 93,95% hingga 95,58%, sehingga perbedaannya tidak terlalu jauh antar pembagian data. Nilai presisi rata-rata sebesar 93,20%, *recall* rata-rata sebesar 91,97%, dan *f1-score* rata-rata sebesar 92,58%, yang menandakan model tetap stabil saat diuji pada *fold* yang berbeda-beda. Secara umum, skenario 1 dengan ADASYN sudah memberikan

performa tinggi, namun metrik *recall* terlihat sedikit lebih bervariasi dibanding metrik lain karena jumlah dan tingkat kesulitan kasus Alzheimer bisa berbeda pada tiap *fold*.

Tabel 4.21 Hasil Evaluasi 5-*Fold Cross Validation* Pada Skenario 1

Iterasi	Akurasi	Presisi	*Recall*	*F1-score*
Fold 1	95,58%	94,04%	93,42%	93,73%
Fold 2	94,19%	92,62%	90,79%	91,69%
Fold 3	93,95%	93,15%	89,47%	91,28%
Fold 4	95,35%	92,86%	94,08%	93,46%
Fold 5	94,87%	93,33%	92,11%	92,72%
Rata-rata (std)	94,79% (0,71%)	93,20% (0,54%)	91,97% (1,88%)	92,58% (1,07%)

Pada Gambar 4.31, kestabilan performa antar *fold* dapat dilihat dari rentang nilai (minimum–maksimum) dan standar deviasi pada setiap metrik. Untuk metrik akurasi, nilai per *fold* berada pada rentang 93,95% hingga 95,58% (selisih 1,63 poin persentase) dengan standar deviasi 0,71%, sehingga perbedaan akurasi antar *fold* tergolong kecil. Untuk presisi, nilai berada pada rentang 92,62% hingga 94,04% (selisih 1,42 poin persentase) dengan standar deviasi 0,54%, yang menunjukkan presisi juga stabil antar *fold*. Sebaliknya, metrik *recall* memiliki variasi paling besar karena nilainya berada pada rentang 89,47% hingga 94,08% (selisih 4,61 poin persentase) dengan standar deviasi 1,88%, sehingga kemampuan model dalam menangkap kasus Alzheimer lebih sensitif terhadap perbedaan pembagian data pada tiap *fold*. Adapun *f1-score* berada pada rentang 91,28% hingga 93,73% (selisih 2,45 poin persentase) dengan standar deviasi 1,07%, yang dapat dipahami karena *f1-score* merupakan gabungan presisi dan *recall*. Ketika *recall* meningkat atau menurun pada *fold* tertentu, *f1-score* ikut berubah, tetapi variasinya tidak sebesar *recall*.



Gambar 4.31 *Boxplot* Performa 5-*Fold Cross Validation* Skenario 1

4.5.2 Evaluasi 5-*Fold Cross Validation* Skenario 10

Berdasarkan Tabel 4.22, pada skenario 10 tanpa ADASYN, performa per kelas juga menunjukkan pola yang baik dan cenderung lebih tinggi. Pada kelas Alzheimer (1), nilai presisi sebesar 95,13%, *recall* sebesar 92,24%, dan *f1-score* sebesar 93,65%, sehingga model semakin baik dalam menghasilkan prediksi positif yang benar. Pada kelas Non-Alzheimer (0), nilai *precision* sebesar 95,83%, recall sebesar 97,41%, dan F1-score sebesar 96,61%, yang menunjukkan model sangat kuat dalam mengenali kelas negatif. Nilai *Macro Average* berada di kisaran 95%, yang menegaskan bahwa secara rata untuk kedua kelas, performa skenario ini lebih unggul dibanding skenario terbaik yang menggunakan ADASYN.

Tabel 4.22 Rata-Rata Performa Klasifikasi per Kelas 5-*Fold Cross Validation* Pada Skenario 10

Kelas	**Presisi**	***Recall***	***F1-score***
Alzheimer (1)	95,13% (0,75%)	92,24% (2,00%)	93,65% (0,75%)
Non-ALZHEIMER (0)	95,83% (1,00%)	97,41% (0,47%)	96,61% (0,33%)
Rata-rata *Macro Average* (std)	95,48% (0,27%)	94,82% (0,80%)	95,13% (0,54%)

Berdasarkan Tabel 4.23, hasil evaluasi per *fold* menunjukkan bahwa skenario 10 memiliki performa yang sangat stabil. Nilai akurasi rata-rata sebesar

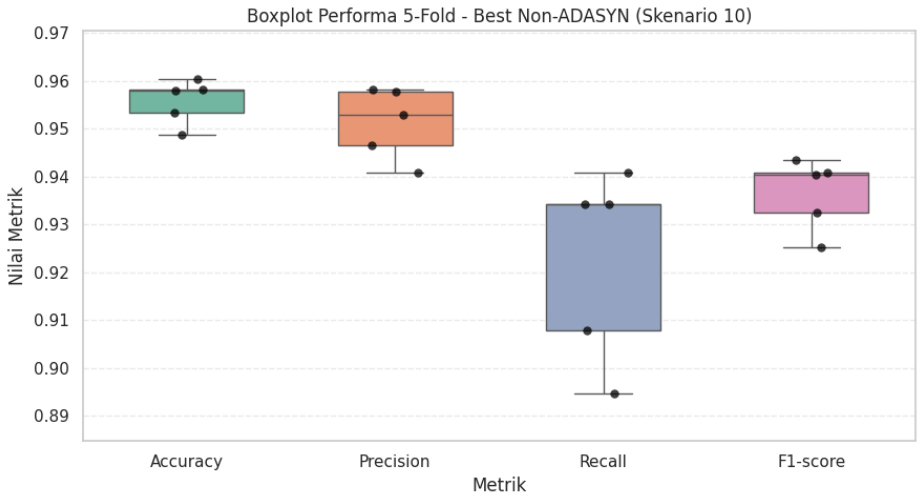
95,58% dengan rentang antar *fold* 94,88% hingga 96,05%, sehingga variasinya kecil dan cenderung konsisten. Nilai presisi rata-rata sebesar 95,13%, *recall* rata-rata sebesar 92,24%, dan *f1-score* rata-rata sebesar 93,65%, yang menandakan model tetap baik dalam mengenali kelas Alzheimer maupun *non*-Alzheimer walaupun tidak menggunakan data sintetis. Dengan kata lain, pada dataset ini, pelatihan menggunakan data asli tanpa ADASYN justru menghasilkan performa yang lebih optimal dan tetap stabil ketika diuji pada pembagian data yang berbeda.

Tabel 4.23 Hasil Evaluasi 5-*Fold Cross Validation* Pada Skenario 10

Iterasi	**Akurasi**	**Presisi**	***Recall***	***F1-score***
Fold 1	95,81%	94,67%	93,42%	94,04%
Fold 2	95,35%	95,83%	90,79%	93,24%
Fold 3	94,88%	95,77%	89,47%	92,56%
Fold 4	96,05%	95,30%	93,42%	94,35%
Fold 5	95,80%	94,08%	94,08%	94,08%
Rata-rata (std)	95,58% (0,46%)	95,13% (0,75%)	92,24% (2,00%)	93,65% (0,75%)

Berdasarkan Gambar 4.32, kestabilan performa skenario 10 dapat dilihat dari rentang nilai (minimum–maksimum) serta standar deviasi pada setiap metrik. Untuk akurasi, nilai per *fold* berada pada rentang 94,88% hingga 96,05% (selisih 1,17 poin persentase) dengan standar deviasi 0,46%, sehingga menunjukkan variasi akurasi antar *fold* kecil. Untuk presisi, nilai berada pada rentang 94,08% hingga 95,83% (selisih 1,75 poin persentase) dengan standar deviasi 0,75%, yang menandakan presisi tetap stabil meskipun ada perbedaan jumlah prediksi positif pada tiap *fold*. Pada *recall*, variasi menjadi yang paling besar karena nilainya berada pada rentang 89,47% hingga 94,08% (selisih 4,61 poin persentase) dengan standar deviasi 2,00%, sehingga kemampuan model menangkap kasus Alzheimer paling sensitif terhadap perbedaan pembagian data uji pada tiap *fold*. Adapun *f1-score* berada pada rentang 92,52% hingga 94,35% (selisih 1,83 poin persentase) dengan

standar deviasi 0,75%, yang wajar karena *f1-score* dipengaruhi perubahan presisi dan *recall* secara bersamaan.



Gambar 4. 32 *Boxplot* Performa 5-*Fold Cross Validation* Skenario 10

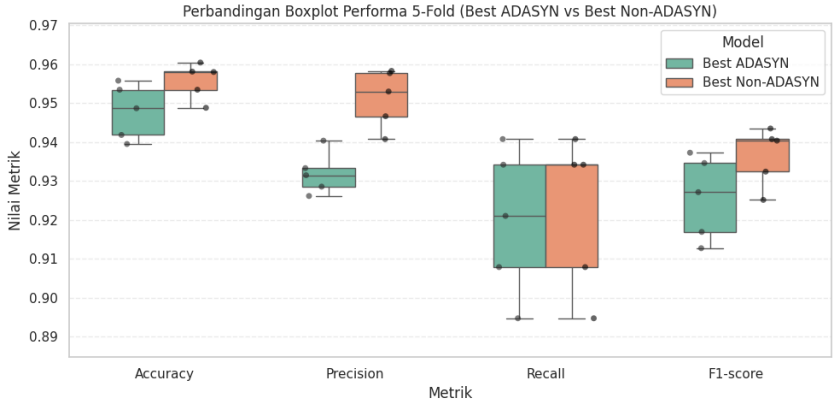
Langkah berikutnya adalah melakukan perbandingan langsung antara skenario 1 dan skenario 10. Perbandingan ini diperlukan untuk memperjelas seberapa besar perubahan nilai performa ketika konfigurasi terbaik berpindah dari skenario 1 yang menggunakan ADASYN dengan *threshold pearson* 0,0 (32 fitur) menuju skenario 10 yang tidak menggunakan ADASYN dengan *threshold pearson* 0,15 (5 fitur). Selain melihat nilai rata-rata, perbandingan juga memperhatikan standar deviasi untuk menilai kestabilan performa antar *fold*, sehingga dapat diketahui apakah peningkatan performa terjadi secara konsisten atau hanya pada *fold* tertentu. Oleh karena itu, hasil perbandingan kedua skenario ditampilkan pada tabel berikut sebagai dasar penarikan kesimpulan model terbaik.

Tabel 4.24 Perbandingan Hasil 5-*Fold Cross Validation* Skenario 1 dan Skenario 10 (Mean ± Std) serta Perubahan Nilai

Metrik	Skenario 1	Skenario 10	Perubahan
Akurasi	94,79% (0,71%)	95,58% (0,46%)	+0,79%
Presisi	93,20% (0,54%)	95,13% (0,75%)	+1,93%
Recall	91,97% (1,88%)	92,24% (2,00%)	+0,26%

Metrik	Skenario 1	Skenario 10	Perubahan
F1-score	92,58% (1,07%)	93,65% (0,75%)	+1,07%

Berdasarkan Tabel 4.24, perpindahan konfigurasi terbaik dari skenario 1 ke skenario 10 menghasilkan peningkatan pada seluruh metrik evaluasi. Akurasi meningkat dari 94,79% menjadi 95,58% (naik 0,79 poin persentase), presisi meningkat dari 93,20% menjadi 95,13% (naik 1,93 poin persentase), *recall* meningkat dari 91,97% menjadi 92,24% (naik 0,26 poin persentase), dan *f1-score* meningkat dari 92,58% menjadi 93,65% (naik 1,07 poin persentase). Peningkatan terbesar terjadi pada presisi dan *f1-score*, yang menunjukkan bahwa pada skenario 10 model lebih baik dalam menghasilkan prediksi Alzheimer yang tepat sekaligus menjaga keseimbangan antara presisi dan *recall*.



Gambar 4.33 Perbandingan *Boxplot* Performa 5-*Fold Cross Validation* Skenario 1 & Skenario 10

Jika dilihat pada Gambar 4.33, performa skenario 10 cenderung lebih tinggi dibanding skenario 1 pada metrik akurasi, presisi, dan *f1-score*. Pada skenario 10, nilai akurasi per *fold* berada pada rentang 94,88%–96,05% (selisih rentang 1,17 poin persentase), sedangkan pada skenario 1 berada pada rentang 93,95%–95,58% (selisih rentang 1,63 poin persentase), sehingga variasi akurasi antar fold pada

skenario 10 lebih kecil. Untuk *f1-score*, skenario 10 memiliki rentang 92,52%–94,35% (selisih rentang 1,83 poin persentase), sedangkan skenario 1 memiliki rentang 91,28%–93,73% (selisih rentang 2,45 poin persentase), yang menunjukkan *f1-score* skenario 10 juga lebih stabil. Namun, *recall* tetap menjadi metrik dengan variasi terbesar pada kedua scenario. *Recall* skenario 10 berada pada rentang 89,47%–94,08% (selisih 4,61 poin persentase) dan *recall* skenario 1 berada pada rentang 89,47%–94,08% (selisih 4,61 poin persentase), sehingga dapat disimpulkan bahwa kemampuan menangkap kasus Alzheimer paling sensitif terhadap perbedaan pembagian data antar *fold*.

Berdasarkan hasil validasi 5-*Fold Cross Validation*, konfigurasi terbaik pada penelitian ini terdapat pada skenario 10 (tanpa ADASYN) dengan *threshold pearson* 0,15 dan jumlah fitur terpilih 5 fitur. Skenario ini menghasilkan performa tertinggi pada metrik utama, yaitu akurasi 95,58%, presisi 95,13%, *recall* 92,24%, dan *f1-score* 93,65%, serta menunjukkan performa yang konsisten antar *fold*. Hasil perbandingan dengan skenario 1 memperkuat bahwa peningkatan performa terutama terjadi pada presisi dan *f1-score*, sementara *recall* tetap menjadi metrik yang paling sensitif terhadap variasi pembagian data. Temuan ini menunjukkan bahwa pada dataset penelitian ini, performa model lebih dipengaruhi oleh pemilihan konfigurasi parameter dan seleksi fitur yang tepat dibandingkan penerapan penyeimbangan data sintetis, sehingga model mampu menghasilkan prediksi yang lebih akurat dan stabil untuk klasifikasi Alzheimer.

4.6 Hubungan Hasil Penelitian dalam Perspektif *Maqasid Syariah*

Hasil penelitian ini memberikan kontribusi nyata terhadap prinsip-prinsip dasar etika Islam, yaitu *Maqasid Syariah* atau biasa dikenal dengan tujuan-tujuan luhur syariah. Kontribusi ini tidak hanya terletak pada hasil akhir, tetapi juga pada keseluruhan proses penelitian. Analisis ini secara spesifik menghubungkan temuan penelitian dengan dua tujuan utama syariah yaitu, *Hifz al-'Aql* (pemeliharaan akal) dan *Hifz al-Nafs* (pemeliharaan jiwa), yang dicapai melalui fondasi *'ilm* (ilmu) dan *itqan* (profesionalitas).

Keterkaitan utama penelitian ini adalah pemenuhan *Hifz al-'Aql* atau pemeliharaan akal. Penyakit Alzheimer secara khusus merupakan ancaman langsung terhadap akal, karena menyerang dan merusak fungsi kognitif, memori, dan kemampuan berpikir jernih. Dalam Islam, akal adalah anugerah yang membedakan manusia dan menjadi dasar *taklif* atau beban syariat. Oleh karena itu, *Hifz al-'Aql* tidak hanya dipahami secara terbatas sebagai larangan meminum *khamr*. Dalam konteks modern, *Maqasid* ini menuntut ikhtiar proaktif untuk melindungi otak dari segala ancaman, termasuk penyakit degeneratif (Tahir & Hamid, 2024). Upaya teknologis untuk membangun model klasifikasi dengan akurasi 95.8% merupakan bentuk ikhtiar nyata untuk menjaga anugerah akal, sebab deteksi dini memberikan kesempatan untuk intervensi yang dapat memperlambat kerusakan kognitif tersebut.

Penelitian ini juga merupakan implementasi langsung dari *Hifz al-Nafs* atau pemeliharaan jiwa dan kehidupan. *Nafs* atau jiwa mencakup kehidupan fisik dan kesejahteraan psikologis. Penyakit Alzheimer memberikan beban berat tidak hanya

pada fisik pasien tetapi juga pada kondisi mental dan emosional pasien serta keluarga. Temuan model terbaik yaitu model 1 dan 10 dengan nilai tertingginya yang mencapai *f1-score* 93.69% sangat penting, karena ini menunjukkan kemampuan model yang seimbang dalam mengidentifikasi kasus positif secara akurat, yang berarti meminimalkan *false negative* dan tidak salah mendiagnosis individu sehat atau meminimalkan *false positive*. Akurasi ini sangat penting karena diagnosis dini memungkinkan penanganan medis dan dukungan psikososial yang tepat, yang dapat memperlambat progresi penyakit, mengurangi penderitaan, serta menjaga kualitas hidup (Akmal et al., 2024). Tindakan menjaga kualitas hidup dan mengurangi penderitaan ini adalah bagian integrasi dari *Hifz al-Nafs*, yang selaras dengan firman Allah SWT dalam Q.S. Al-Ma'idah ayat 32:

مِنْ اَجْلِ ذٰلِكَۚ كَتَبْنَا عَلٰى بَنِيْٓ اِسْرَاۤءِيْلَ اَنَّهٗۗ مَنْ قَتَلَ نَفْسًاۢ بِغَيْرِ نَفْسٍ اَوْ فَسَادٍ فِى الْاَرْضِ فَكَاَنَّمَا قَتَلَ

النَّاسَ جَمِيْعًاۗ وَمَنْ اَحْيَاهَا فَكَاَنَّمَآ اَحْيَا النَّاسَ جَمِيْعًاۗ وَلَقَدْ جَاۤءَتْهُمْ رُسُلُنَا بِالْبَيِّنٰتِ ثُمَّ اِنَّ كَثِيْرًا مِّنْهُمْ بَعْدَ ذٰلِكَ فِى

الْاَرْضِ لَمُسْرِفُوْنَ

"Oleh karena itu, Kami menetapkan (suatu hukum) bagi Bani Israil bahwa siapa yang membunuh seseorang bukan karena (orang yang dibunuh itu) telah membunuh orang lain atau karena telah berbuat kerusakan di bumi, maka seakan-akan dia telah membunuh semua manusia. Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia. Sungguh, rasul-rasul Kami benar-benar telah datang kepada mereka dengan (membawa) keterangan-keterangan yang jelas. Kemudian, sesungguhnya banyak di antara mereka setelah itu melampaui batas di bumi."

Berdasarkan Tafsir Tahlili Kementerian Agama RI yang dikutip dari laman NU Online (*Surat Al-Ma'idah Ayat 32*, n.d.), ayat ini menegaskan tingginya nilai

kemanusiaan dalam Islam. Memelihara kehidupan di sini mencakup segala upaya untuk menyelamatkan manusia dari kebinasaan, baik melalui pengobatan, pemenuhan kebutuhan, maupun pencegahan penyakit. Dalam konteks penelitian ini, mengembangkan sistem deteksi dini untuk pasien Alzheimer adalah bentuk nyata dari "memelihara kehidupan" (*ihya*), karena sistem ini berupaya menyelamatkan kualitas hidup pasien dari kerusakan permanen yang lebih parah, yang pada hakikatnya adalah upaya menjaga kelangsungan hidup manusia itu sendiri.

Pencapaian kedua *Maqasid Syariah* tersebut menuntut landasan profesionalitas (*itqan*) dan ilmu (*'ilm*). Sebuah ikhtiar tidak bernilai jika dilakukan secara asal-asalan tanpa aturan yang jelas. Proses sistematis dalam penelitian ini, yang menguji 12 skenario hingga menemukan konfigurasi terbaik adalah bentuk cerminan dari penerapan nilai *itqan* dalam membangun sistem yang teratur. Sikap ilmiah untuk memastikan setiap data terklasifikasi pada kelas yang benar ini selaras dengan prinsip keteraturan alam semesta, di mana setiap benda langit beredar pada porosnya tanpa saling melampaui batas, sebagaimana firman Allah SWT dalam Q.S. Yasin ayat 40:

لَا الشَّمْسُ يَنْۢبَغِيْ لَهَآ اَنْ تُدْرِكَ الْقَمَرَ وَلَا الَّيْلُ سَابِقُ النَّهَارِۗ وَكُلٌّ فِيْ فَلَكٍ يَّسْبَحُوْنَ

“Tidaklah mungkin bagi matahari mengejar bulan dan malam pun tidak dapat mendahului siang. Masing-masing beredar pada garis edarnya.”

Merujuk pada Tafsir Tahlili yang bersumber dari NU *Online* (*Surat Yasin Ayat 40*, n.d.), ayat ini menegaskan adanya *Sunnatullah* atau ketetapan peraturan Allah yang berlaku bagi benda-benda alam, di mana tidak mungkin terjadi tabrakan

antara matahari dan bulan karena masing-masing tetap bergerak menurut garis edarnya yang telah ditetapkan. Tafsir ini menekankan betapa kekuasaan Allah menciptakan sistem yang berjalan dengan tertib dan presisi, berbeda dengan aturan buatan manusia seperti peraturan lalu lintas yang sering kali masih memiliki celah kesalahan atau kecelakaan.

Hubungannya dengan penelitian ini sangat kuat. Pengembangan model XGBoost merupakan upaya ilmiah untuk meniru prinsip keteraturan tersebut dengan menetapkan batas keputusan atau *decision boundary* yang tepat sebagai garis edar bagi data. Tujuannya adalah memastikan data pasien sehat dan pasien Alzheimer berjalan pada jalurnya masing-masing sehingga tidak terjadi tabrakan berupa kesalahan diagnosis atau *misclassification*. Hasil akurasi tinggi sebesar 95.8% membuktikan bahwa model ini telah mencapai tingkat keteraturan yang tinggi dan meminimalkan *error*, sehingga menjadi wujud nyata dari nilai *itqan* dalam menjaga keselamatan jiwa manusia.

Pengembangan teknologi cerdas seperti XGBoost di bidang kesehatan wajib selaras dengan nilai-nilai syariah, khususnya terkait prinsip keterbukaan (Yunos & Hamdan, 2024). Sistem diagnosis tidak boleh bekerja secara tertutup tanpa alur yang jelas atau biasa dikenal dengan *black box*, melainkan harus memiliki landasan logika yang dapat dipertanggungjawabkan. Temuan pada Subbab 4.3.2 menjawab tantangan ini, di mana model terbaik ada pada model 10, terbukti dapat diinterpretasikan dengan baik. Hal ini terlihat dari proses seleksi fitur yang memastikan keputusan model didasarkan pada indikator klinis yang relevan.

Dengan demikian, logika yang dibangun oleh model sejalan dengan prinsip klinis, menjadikannya layak diterapkan baik secara medis maupun syariah.

Penelitian ini, mengonfirmasi bahwa pengembangan model klasifikasi XGBoost untuk Alzheimer bukan sekadar pencapaian teknis akademis. Namun, sebuah kontribusi ilmiah yang berakar kuat pada *Maqasid Syariah*. Model ini berfungsi sebagai sarana untuk *Hifz al-'Aql* dengan melindungi akal dari dampak penyakit, dan *Hifz al-Nafs* melalui penjagaan jiwa dengan deteksi dini. Prosesnya sendiri menjunjung tinggi nilai *'ilm* (ilmu) dan *itqan* (ketelitian), menghasilkan model yang tidak hanya akurat secara teknis tetapi juga dapat dipertanggungjawabkan secara etis dan sejalan dengan nilai keislaman.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, dapat disimpulkan bahwa metode XGBoost menunjukkan performa yang sangat efektif dalam mengklasifikasikan penyakit Alzheimer. Model terbaik yang dikembangkan yaitu skenario 10 berhasil mencapai tingkat akurasi pengujian sebesar 95,58%, dengan presisi 94,63%, *recall* 92,76%, dan *f1-score* 93,69%. Keandalan model ini semakin diperkuat melalui evaluasi validasi silang lima lipatan atau 5-*Fold Cross Validation* yang menunjukkan konsistensi tinggi dengan rata-rata akurasi sebesar 95,58% dan standar deviasi yang sangat rendah yaitu 0,46%. Tingginya nilai metrik evaluasi serta kestabilan hasil validasi ini membuktikan bahwa algoritma XGBoost mampu memberikan prediksi yang tidak hanya tepat secara global, tetapi juga seimbang dalam meminimalkan kesalahan diagnosis baik *false positive* maupun *false negative* sehingga sangat potensial untuk digunakan sebagai alat bantu deteksi dini yang akurat.

Penelitian ini juga menemukan bahwa performa metode XGBoost sangat dipengaruhi oleh tiga faktor penting, yakni metode penyeimbang data, seleksi fitur, dan konfigurasi *hyperparameter*. Penggunaan metode penyeimbang data ADASYN terbukti menjadi faktor yang justru menurunkan performa model pada dataset ini. Berbeda dengan asumsi umum pada data tidak seimbang, penerapan ADASYN menyebabkan penurunan akurasi rata-rata dari 91,09% menjadi 87,09%

dibandingkan model tanpa penyeimbang data. Penurunan ini disebabkan oleh karakteristik data asli yang bersih dan minim pencilan, sehingga penambahan data sintetis di area perbatasan kelas oleh ADASYN justru menciptakan distorsi atau gangguan yang membingungkan model dalam melakukan klasifikasi.

Selain itu, seleksi fitur menggunakan korelasi *pearson* memberikan dampak signifikan terhadap efisiensi dan akurasi model. Penggunaan nilai ambang batas 0,15 terbukti paling optimal karena mampu mereduksi dimensi data menjadi lima fitur kunci tanpa menurunkan performa model. Sebaliknya, penggunaan ambang batas yang terlalu besar yaitu 0,25 menjadi faktor negatif yang menyebabkan penurunan performa drastis dengan *f1-score* di bawah 79% akibat hilangnya informasi penting terutama skor MMSE yang merupakan indikator klinis utama. Terakhir, optimasi *hyperparameter* menunjukkan bahwa model bekerja paling baik dengan strategi pembelajaran lambat atau *slow learning* dengan laju pembelajaran rendah sebesar 0,01 dan jumlah pohon sebanyak 500 serta struktur pohon yang sederhana dengan kedalaman maksimal 3. Hal ini mengindikasikan bahwa pola data asli cukup jelas sehingga tidak memerlukan model yang terlalu kompleks untuk mencapai generalisasi terbaik.

5.2 Saran

Peneliti menyadari bahwa hasil penelitian ini belum sepenuhnya sempurna dan masih terdapat ruang untuk perbaikan. Agar sistem dapat berfungsi dengan lebih optimal dan memberikan wawasan yang lebih mendalam, diperlukan beberapa langkah pengembangan. Berikut adalah saran yang disampaikan peneliti untuk penelitian selanjutnya:

a. Penelitian ini hanya berfokus pada data klinis berbentuk tabular. Penelitian selanjutnya sangat disarankan untuk menggabungkan data tabular dengan data pencitraan medis (*medical imaging*) seperti MRI atau *CT Scan* otak. Kombinasi data multimodal berpotensi meningkatkan akurasi diagnostik secara signifikan karena menggabungkan indikator kognitif dengan bukti fisik atrofi otak.
b. Mengingat ADASYN terbukti kurang efektif pada dataset ini, penelitian selanjutnya dapat mengeksplorasi teknik penanganan *imbalance* lainnya. Metode *Undersampling* seperti *Tomek Links* untuk membersihkan perbatasan kelas, atau pendekatan *Cost-Sensitive Learning* dengan memberikan bobot penalti lebih besar pada kesalahan kelas minoritas dapat diuji coba untuk melihat apakah dapat mengungguli performa data asli.
c. Disarankan untuk membandingkan performa XGBoost dengan arsitektur *Deep Learning modern* yang dirancang khusus untuk data tabular, seperti *TabNet* atau *FT-Transformer*. Perbandingan ini berguna untuk mengetahui apakah model berbasis *neural network* mampu menangkap pola non-linear yang lebih kompleks yang mungkin terlewatkan oleh model berbasis *tree*.
d. Agar manfaat penelitian ini dapat dirasakan langsung oleh masyarakat atau praktisi kesehatan, model yang telah dilatih sebaiknya dikembangkan menjadi aplikasi berbasis web atau *mobile* dengan antarmuka yang ramah pengguna. Hal ini akan memudahkan proses skrining awal Alzheimer secara mandiri maupun di fasilitas kesehatan tingkat pertama.

DAFTAR PUSTAKA

A Ilemobayo, J., Durodola, O., Alade, O., J Awotunde, O., T Olanrewaju, A., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., Ifeanyi, A., E Odezuligbo, I., & E Edu, O. (2024). Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*, *26*(6), 388–395. https://doi.org/10.9734/jerr/2024/v26i61188

Aftha Harianto, F. R., Alawi, Z., & Sa'ida, I. A. (2025). PENGARUH KOMPOSISI SPLIT DATA PADA AKURASI KLASIFIKASI PENDERITA DIABETES MENGGUNAKAN ALGORITMA MACHINE LEARNING. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *8*(1), 36–44. https://doi.org/10.47080/simika.v8i1.3663

Akmal, H., Muqorobin, A., Marjany, N., & Romadonika, F. (2024). The Concept of Nursing Care in Maqashid Sharia Perspective. *Ijtihad*, *18*(2). https://doi.org/10.21111/ijtihad.v18i2.13055

Allorerung, P. P., Erna, A., Bagussahrir, M., & Alam, S. (2024). Analisis Performa Normalisasi Data untuk Klasifikasi K-Nearest Neighbor pada Dataset Penyakit. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *9*(3), 178–191. https://doi.org/10.14421/jiska.2024.9.3.178-191

Alzheimer's Association. (2025). 2025 Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, *21*(4), e70235. https://doi.org/10.1002/alz.70235

Andryan, M. R., Fajri, M., & Sulistyowati, N. (2022). KOMPARASI KINERJA ALGORITMA XGBOOST DAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) UNTUK DIAGNOSIS PENYAKIT KANKER PAYUDARA. *JIKO (Jurnal Informatika Dan Komputer)*, *6*(1), 1. https://doi.org/10.26798/jiko.v6i1.500

Arif Ali, Z., H. Abduljabbar, Z., A. Tahir, H., Bibo Sallow, A., & Almufti, S. M. (2023). eXtreme Gradient Boosting Algorithm with Machine Learning: A Review. *Academic Journal of Nawroz University*, *12*(2), 320–334. https://doi.org/10.25007/ajnu.v12n2a1612

Awal, Md. A., Masud, M., Hossain, Md. S., Bulbul, A. A.-M., Mahmud, S. M. H., & Bairagi, A. K. (2021). A Novel Bayesian Optimization-Based Machine Learning Framework for COVID-19 Detection From Inpatient Facility Data. *IEEE Access*, *9*, 10263–10281. https://doi.org/10.1109/ACCESS.2021.3050852

Barry-Straume, J., Tschannen, A., Engels, D., & Fine, E. (2019). An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional

Neural Networks. *SMU Data Science Review*, *1*(4). https://scholar.smu.edu/datasciencereview/vol1/iss4/12

Bestari, A. P. B. (2023, Oktober). *Mengenal Demensia Alzheimer* [Artikel Kesehatan]. Kemenkes, Direktorat Jenderal Kesehatan Lanjut. https://keslan.kemkes.go.id/view_artikel/2819/mengenal-demensia-alzheimer

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

Borders, J. C., Blanke, S., Johnson, S., Gilmore-Bykovskyi, A., & Rogus-Pulia, N. (2020). Efficacy of Mealtime Interventions for Malnutrition and Oral Intake in Persons With Dementia: A Systematic Review. *Alzheimer Disease & Associated Disorders*, *34*(4), 366–379. https://doi.org/10.1097/WAD.0000000000000387

Budi Utomo, P., Faruqziddan, M., Herdika Septa Aulia, E., & Dini Azzahra, S. (2024). Perbandingan Skenario Balancing Oversampling dan Undersampling dalam Klasifikasi Resiko Kambuh Kanker Tiroid menggunakan Algoritma SVM Linear. *JAMI: Jurnal Ahli Muda Indonesia*, *5*(2), 172–182. https://doi.org/10.46510/jami.v5i2.320

Carbonell, M. C., & Paulini, J. Z. (2025). Evaluation of machine learning models for the prediction of Alzheimer's: In search of the best performance. *Brain, Behavior, & Immunity - Health*, *47*, 100957. https://doi.org/10.1016/j.bbih.2025.100957

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Daffaa, R. M., Santika, D., & Mahardika. (2025). Perbandingan Xgboost dan Logistic Regression dalam Memprediksi Credit Card Customer Churn. *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro Dan Informatika*, *3*(3), 07–21. https://doi.org/10.61132/jupiter.v3i3.807

Darmawan, D. B., Azahwa, I. R., Saputra, R. W., Septiadi, R., & Rosyani, P. (2024). Klasifikasi Penyakit Jantung Menggunakan Extreme Gradient Boosting (XGBoost). *JRIIN : Jurnal Riset Informatika dan Inovasi*, *2*(8), 1427–1434.

Digambiro, R. A., Marsiati, H., Hadi, R. S., Hastuty, D., & Parwanto, E. (2024). ANALISIS KOMPARATIF ATROFI SEREBRAL PADA GANGGUAN KOGNITIF NON-ALZHEIMER DAN PENYAKIT ALZHEIMER DI INDONESIA. *Medika Alkhairaat: Jurnal Penelitian Kedokteran Dan Kesehatan*, *6*(3), 684–692. https://doi.org/10.31970/ma.v6i3.245

Elshawi, R., Maher, M., & Sakr, S. (2019). *Automated Machine Learning: State-of-The-Art and Open Challenges* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1906.02287

Frazier, P. I. (2018). *A Tutorial on Bayesian Optimization* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1807.02811

Ginting, J., Ginting, R., & Hartono, H. (2022). DETEKSI DAN PREDIKSI PENYAKIT DIABETES MELITUS TIPE 2 MENGGUNAKAN MACHINE LEARNING (SCOOPING REVIEW). *Jurnal Keperawatan Priority*, *5*(2), 93–105. https://doi.org/10.34012/jukep.v5i2.2671

Goel, P., Chakrabarti, S., Goel, K., Bhutani, K., Chopra, T., & Bali, S. (2022). Neuronal cell death mechanisms in Alzheimer's disease: An insight. *Frontiers in Molecular Neuroscience*, *15*, 937133. https://doi.org/10.3389/fnmol.2022.937133

Haibo He, Yang Bai, Garcia, E. A., & Shutao Li. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Haluska, R., Brabec, J., & Komarek, T. (2022). Benchmark of Data Preprocessing Methods for Imbalanced Classification. *2022 IEEE International Conference on Big Data (Big Data)*, 2970–2979. https://doi.org/10.1109/BigData55660.2022.10021118

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

Idris, M., Wijaya, A., Septiani, L., Happy, T. A., & Graharti, R. (2025). Pemanfaatan Kecerdasan Buatan sebagai Alat Bantu Diagnosis di Bidang Kesehatan: Literatur Review: Mohamad Idris, Angga Wijaya, Linda Septiani, Terza Aflika Happy, Risti Graharti. *Jurnal Kedokteran Universitas Lampung*, *9*(1), 117–121. https://doi.org/10.23960/jkunila.v9i1.pp117-121

Iriananda, S. W., Paramita, N., & Kurniawan, R. P. (2025). DETECTION OF YOUTUBE COMMENT SPAM USING XGBOOST: OPTIMIZATION

AND COMPARISON OF MACHINE LEARNING ALGORITHMS. *JOURNAL OF SCIENCE AND APPLIED ENGINEERING*, *8*(2), 56–66. https://doi.org/10.31328/jsae.v8i2.7296

Jung, Y., Park, Y., Jo, J., & Jeong, J. (2025). MMSE-Based Dementia Prediction: Deep vs. Traditional Models. *Life*, *15*(10), 1544. https://doi.org/10.3390/life15101544

Komala, W., & Nasution, A. N. (2025). Penyakit Degeneratif: Tantangan dan Peluang dalam Penelitian dan Pengobatan. *JURNAL PENDIDIKAN TAMBUSAI*, *9*(1), 7637–7647.

Krinitski, D., Kasina, R., Klöppel, S., & Lenouvel, E. (2021). Associations of delirium with urinary tract infections and asymptomatic bacteriuria in adults aged 65 and older: A systematic review and meta-analysis. *Journal of the American Geriatrics Society*, *69*(11), 3312–3323. https://doi.org/10.1111/jgs.17418

Kumar, S., & Rani, P. (2024). An AI-based Liver Disease Prediction Model based on Pearson Correlation Feature Selection Method. *Biomedical and Pharmacology Journal*, *17*(4), 2187–2202.

Kurniawan, I. G. J., Dewi, C., & Rahman, M. A. (2025). Penerapan Machine Learning Extreme Gradient Boosting Dalam Klasifikasi Potensi Tsunami Berdasarkan Data Gempa Bumi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *9*(2). https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14437

Kusuma, E. J., Nurmandhani, R., Aryani, L., Pantiawati, I., & Shidik, G. F. (2025). Optimasi Model Extreme Gradient Boosting Dalam Upaya Penentuan Tingkat Risiko Pada Ibu Hamil Berbasis Bayesian Optimization (BOXGB). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *12*(1), 111–120. https://doi.org/10.25126/jtiik.2025129001

Leni, D., Dwiharzandis, A., Sumiati, R., Haris, H., & Afriyani, S. (2023). Seleksi Fitur Berdasarkan Korelasi Pearson dalam Pemodelan Efisiensi Energi Bangunan. *Teknika Sains: Jurnal Ilmu Teknik*, *8*(2), 103–115. https://doi.org/10.24967/teksis.v8i2.2525

Lumbantobing, H. E., & Nirwana, A. (2023). Lebah dan Madu dalam Surah An-Nahl. *Al Karima : Jurnal Studi Ilmu Al Quran Dan Tafsir*, *7*(2), 30. https://doi.org/10.58438/alkarima.v7i2.176

Maulaningrum, P., Mujanah, S., & Fianto, A. Y. A. (2025). Transformasi Digital di Sektor Kesehatan Tinjauan Literatur tentang Penerapan Teknologi Informasi dalam Manajemen Pelayanan. *Jurnal Ilmu Manajemen, Ekonomi*

Dan Kewirausahaan, *5*(1), 494–503. https://doi.org/10.55606/jimek.v5i1.6399

Mirafuddin, M., & Supriyatna, S. (2024). Penerapan Machine Learning pada Aplikasi Kesehatan HEALTIME. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, *7*(6), 1601–1606. https://doi.org/10.32672/jnkti.v7i6.8233

Moreno-Ibarra, M.-A., Villuendas-Rey, Y., Lytras, M. D., Yáñez-Márquez, C., & Salgado-Ramírez, J.-C. (2021). Classification of Diseases Using Machine Learning Algorithms: A Comparative Study. *Mathematics*, *9*(15), 1817. https://doi.org/10.3390/math9151817

Murdiansyah, D. T. (2024). Prediksi Stroke Menggunakan Extreme Gradient Boosting. *JIKO (Jurnal Informatika Dan Komputer)*, *8*(2), 419. https://doi.org/10.26798/jiko.v8i2.1295

Nalasari, L. T., Anam, S., & Shofianah, N. (2025). Liver Cirrhosis Classification using Extreme Gradient Boosting Classifier and Harris Hawk Optimization as Hyperparameter Tuning. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, *7*(2), 508–519. https://doi.org/10.35882/jeeemi.v7i2.730

Natsir, F. M., Bakti, R. Y., & Wahyuni, T. (2024). Analisis Deteksi Dini Penyakit Jantung dengan Pendekatan Support Vector Machine pada Data Pasien. *Arus Jurnal Sains Dan Teknologi*, *2*(2), 437–446. https://doi.org/10.57250/ajst.v2i2.669

Neyra, J., Siramshetty, V. B., & Ashqar, H. I. (2024). *The effect of different feature selection methods on models created with XGBoost* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2411.05937

Nguyen, K., Nguyen, M., Dang, K., Pham, B., Huynh, V., Vo, T., Ngo, L., & Ha, H. (2023). Early Alzheimer's disease diagnosis using an XG-Boost model applied to MRI images. *Biomedical Research and Therapy*, *10*(9), 5896–5911. https://doi.org/10.15419/bmrat.v10i9.832

Nurbaiti, N., Sutoro, S. G., Supriyaningsih, E., Wiyanti, S. W., & Maesaroh, I. (2023). Edukasi untuk Deteksi Dini dan Perawatan Lansia dengan Alzheimer di Masa Pandemi Covid-19. *Jurnal Kreativitas Pengabdian Kepada Masyarakat (PKM)*, *6*(7), 2887–2895. https://doi.org/10.33024/jkpm.v6i7.10093

Nurmawanti, L., & Sudaryanto, A. (2025). CASE REPORT: PENATALAKSANAAN HOLISTIK PENYAKIT ALZHEIMER PADA LANSIA DI PANTI SOSIAL. *Jurnal Penelitian Perawat Profesional*, *7*(1).

https://jurnal.globalhealthsciencegroup.com/index.php/JPPP/article/view/5783

Oktafiani, R., Hermawan, A., & Avianto, D. (2023). Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning. *Jurnal Sains Dan Informatika*, 19–28. https://doi.org/10.34128/jsi.v9i1.622

Opitasari, O., Natsir, F., & Marsiani, E. S. (2024). Klasifikasi Diagnosis untuk Penyakit Kanker Serviks Menggunakan Algoritma Extreme Gradient Boosting (XGBoost). *Jurnal Ilmiah FIFO*, *16*(1), 55. https://doi.org/10.22441/fifo.2024.v16i1.006

Permana, I., & Salisah, F. N. S. (2022). Pengaruh Normalisasi Data Terhadap Performa Hasil Klasifikasi Algoritma Backpropagation: The Effect of Data Normalization on the Performance of the Classification Results of the Backpropagation Algorithm. *Indonesian Journal of Informatic Research and Software Engineering (IJIRSE)*, *2*(1), 67–72. https://doi.org/10.57152/ijirse.v2i1.311

Pratama, V. (2025). Penerapan Metode Machine Learning Berbasis Pohon Keputusan Untuk Mendiagnosis Penyakit Alzheimer. *JURNAL INTEKSIS (Jurnal Informasi Teknologi Dan Sistem)*, *12*(1). https://doi.org/10.5281/ZENODO.15621168

Putra, B. S. C., Tahyudin, I., Kusuma, B. A., & Isnaini, K. N. (2024). Efektivitas Algoritma Random Forest, XGBoost, dan Logistic Regression dalam Prediksi Penyakit Paru-paru. *Techno.Com*, *23*(4), 909–922. https://doi.org/10.62411/tc.v23i4.11705

Rahmi, U., & Putri, R. (2021). KUALITAS HIDUP (QUALITY OF LIFE) CAREGIVER PASIEN DEMENSIA. *JURNAL AKADEMI KEPERAWATAN HUSADA KARYA JAYA*, *7*(2). https://doi.org/10.59374/jakhkj.v7i2.166

Rainforth, T., Le, T. A., van de Meent, J.-W., Osborne, M. A., & Wood, F. (2017). *Bayesian Optimization for Probabilistic Programs* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1707.04314

Rayadin, M. A., Musaruddin, M., Saputra, R. A., & Isnawaty, I. (2024). Implementasi Ensemble Learning Metode XGBoost dan Random Forest untuk Prediksi Waktu Penggantian Baterai Aki. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, *5*(2), 111–119. https://doi.org/10.37148/bios.v5i2.128

Rokhman, N., Ningtyas, A. M., Salim, M. F., & Santoso, D. B. (2020). Penerapan Sistem Data Cleansing untuk Mencegah dan Menghilangkan Duplikasi

Rekam Medis. *Jurnal Pengabdian Kepada Masyarakat (Indonesian Journal of Community Engagement)*, *6*(4). https://doi.org/10.22146/jpkm.51073

Rumah Sakit Advent. (2024, August 28). *PENYAKIT ALZHEIMER : PENELITIAN TERKINI TENTANG PENYEBAB, RISIKO, DAN PENCEGAHAN MELALUI GAYA HIDUP SEHAT* [Artikel Kesehatan]. Rumah Sakit Advent Bandung. https://rsadventbandung.com/articles/penyakit-alzheimer-penelitian-terkini-tentang-penyebab-risiko-dan-pencegahan-melalui-gaya-hidup-sehat

Saputra, A. A., Sari, B. N., & Rozikin, C. (2024). Penerapan Algoritma Extreme Gradient Boosting (Xgboost) Untuk Analisis Risiko Kredit. *Jurnal Ilmiah Wahana Pendidikan*, *10*(7), 27–36. https://doi.org/10.5281/ZENODO.10960080

Sari, A. P., Billy, Tsaqif, D. A., Sartono, B., & Firdawanti, A. R. (2024). Classification of Drinking Water Source Suitability in West Java Using XGBoost and Cluster Analysis Based on SHAP Values: Klasifikasi Kelayakan Sumber Air Minum di Jawa Barat Menggunakan XGBoost dan Analisis Klasterisasi Berdasarkan Nilai SHAP. *Indonesian Journal of Statistics and Its Applications*, *8*(2), 202–214. https://doi.org/10.29244/ijsa.v8i2p202-214

Sausan, S., Pratiwi, D. M., & Mufidah, L. (2025). Perbandingan Metode Decision Tree Classifier dan XGBoost Classifier Dalam Memprediksi Penyakit Jantung. *Proceedings of the National Conference on Electrical Engineering, Informatics, Industrial Technology, and Creative Media (CENTIVE 2024)*, *4*(1). https://doi.org/doi.org/10.20895/centive.v4i1.336

Shabariah, R., Karnina, R., Husna, I., Zahra, A., & Edison, A. A. (2022). PENGENALAN DEMENSIA PADA ALZHEIMER: TINJAUAN DIAGNOSIS DAN TATALAKSANA. *Prosiding Seminar Nasional Pengabdian Masyarakat LPPM UMJ*, *PROSIDING SEMNASKAT LPPM UMJ 2022*. https://jurnal.umj.ac.id/index.php/semnaskat/article/view/23927

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, *104*(1), 148–175. https://doi.org/10.1109/JPROC.2015.2494218

Sidarta, E., Sari, T., & Nataprawira, S. M. D. (2025). Demensia: Patofisiologi, Faktor Resiko dan Peran Inflamasi dalam Perkembangan Penyakit. *Termometer: Jurnal Ilmiah Ilmu Kesehatan Dan Kedokteran*, *3*(1), 235–244. https://doi.org/10.55606/termometer.v3i1.4801

Sihombing, S. F., Pakpahan, J. P., & Lubis, A. H. (2025). Klasifikasi Produk Iphone dengan Menggunakan Algoritma XGBoost. *Journal of Informatics Management and Information Technology*, *5*(3), 371–380. https://doi.org/10.47065/jimat.v5i3.649

Sitompul, L. R., Nababan, A. A., Manihuruk, M. L., Ponsen, W. A., & Supriyandi, S. (2025). Comparison of Xgboost, Random Forest and Logistic Regression Algorithms in Stroke Disease Classification. *Sinkron*, *9*(2), 957–968. https://doi.org/10.33395/sinkron.v9i2.14794

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, Md. M. A., Prabhat, & Adams, R. P. (2015). *Scalable Bayesian Optimization Using Deep Neural Networks* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1502.05700

Soelistijadi, R., Wismarini, T. D., Eniyati, S., & Sunard, S. (2024). Pemodelan Prediktif Menggunakan Metode Ensemble Learning XGBoost dalam Peningkatan Akurasi Klasifikasi Penyakit Ginjal. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *5*(4), 1866–1875.

Surat Al-Ma'idah Ayat 32: Arab, Latin, Terjemah dan Tafsir Lengkap | Quran NU Online. (n.d.). Retrieved November 20, 2025, from https://quran.nu.or.id/al-maidah/32

Surat Yasin Ayat 40: Arab, Latin, Terjemah dan Tafsir Lengkap | Quran NU Online. (n.d.). Retrieved December 3, 2025, from https://quran.nu.or.id/yasin/40

Suriastini, N. W., Turana, Y., Supraptilah, B., Wicaksono, T. Y., & Mulyanto, E. D. (2020). Prevalence and Risk Factors of Dementia and Caregiver's Knowledge of the Early Symptoms of Alzheimer's Disease. *Aging Medicine and Healthcare*, *11*(2), 60–66. https://doi.org/10.33879/AMH.2020.065-1811.032

Suryadewiansyah, M. K., & Tju, T. E. E. (2022). Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *8*(2), 81–88. https://doi.org/10.25077/TEKNOSI.v8i2.2022.81-88

Susanti, N., Siregar, N. H., Ramadhani, N., & Sihite, R. N. (2024). ALZHEIMER DAN DIMENSIA. *Jurnal Kesehatan Tambusai*, *5*(2). https://doi.org/10.31004/jkt.v5i2.28471

Susanto, E. R., Cahyana, A., & Parjito, P. (2025). Penerapan Algoritma XGBoost untuk Prediksi Diabetes: Analisis Confusion Matrix dan ROC Curve. *Fountain of Informatics Journal*, *10*(1), 40–50. https://doi.org/10.21111/fij.v10i1.14311

Tahir, T., & Hamid, S. H. A. (2024). Maqasid Al-Syari'ah Transformation in Law Implementation for Humanity. *International Journal Ihya' 'Ulum al-Din*, *26*(1), 119–131. https://doi.org/10.21580/ihya.26.1.20248

Tarimana, A. A., Fajar, M. R. S., Saktiawan, M. A., & Saputra, R. A. (2024). PREDIKSI PENYAKIT HIPERTENSI MENGGUNAKAN MACHINE LEARNING DENGAN ALGORITMA REGRESI LOGISTIK. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(6), 12062–12068. https://doi.org/10.36040/jati.v8i6.11793

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2021). *Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2104.10201

Ulfah, M., Saragih, T. H., Kartini, D., Mazdadi, M. I., & Abadi, F. (2023). PENERAPAN MWMOTE UNTUK MENGATASI KETIDAKSEIMBANGAN KELAS PADA KLASIFIKASI RISIKO KREDIT. *Jurnal Informatika Polinema*, *9*(4), 477–486. https://doi.org/10.33795/jip.v9i4.1331

Wardhana, R. G., Wang, G., & Sibuea, F. (2023). PENERAPAN MACHINE LEARNING DALAM PREDIKSI TINGKAT KASUS PENYAKIT DI INDONESIA. *Journal of Information System Management (JOISM)*, *5*(1), 40–45. https://doi.org/10.24076/joism.2023v5i1.1136

Yahya, F. N., Anshori, M., & Khudori, A. N. (2025). Evaluasi Performa XGBoost dengan Oversampling dan Hyperparameter Tuning untuk Prediksi Alzheimer. *Techno.Com*, *24*(1), 1–12. https://doi.org/10.62411/tc.v24i1.12057

Yu, B., Shen, H., Yang, Y., Xu, Q., & He, W. (2025). VFBoost-MO: Scable vertical federated gradient boosting decision tree for multi-class problem. *Journal of Systems Architecture*, *168*, 103545. https://doi.org/10.1016/j.sysarc.2025.103545

Yulianti, S. E. H., Oni Soesanto, & Yuana Sukmawaty. (2022). Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics: Theory and Applications*, 21–26. https://doi.org/10.31605/jomta.v4i1.1792

Yunos, A. S., & Hamdan, M. N. (2024). ARTIFICIAL INTELLIGENCE IN HEALTHCARE: ANALYSIS FROM THE PERSPECTIVE OF MAQASID AL-SHARIAH. *Journal of Information System and Technology Management*, *9*(35), 58–74. https://doi.org/10.35631/JISTM.935004

Zega, S. A., Saropi, A., Ronaldy, T., Ilhami, Y., & Farabi, M. (2022). Mengatasi Data Error Pada Proses Data Cleaning Motion Capture Motive Optitrack. *JOURNAL OF APPLIED MULTIMEDIA AND NETWORKING*, *6*(1), 89–94. https://doi.org/10.30871/jamn.v6i1.4210

Zhou, W., & Xie, Z. (2025). Enhancing Sealing Performance Predictions: A Comprehensive Study of XGBoost and Polynomial Regression Models with Advanced Optimization Techniques. *Materials*, *18*(10), 2392. https://doi.org/10.3390/ma18102392

Zhu, J., Pu, S., He, J., Su, D., Cai, W., Xu, X., & Liu, H. (2024). Processing imbalanced medical data at the data level with assisted-reproduction data as an example. *BioData Mining*, *17*(1), 29. https://doi.org/10.1186/s13040-024-00384-y

LAMPIRAN

LAMPIRAN-LAMPIRAN

Lampiran I. Perhitungan Korelasi *Pearson* untuk Seleksi Fitur

Lampiran ini berisi proses perhitungan *feature selection* menggunakan korelasi *pearson* untuk fitur-fitur yang tereliminasi ketika nilai *threshold* dinaikkan dari 0,15 menjadi 0,25. Berdasarkan hasil seleksi fitur, terdapat dua fitur yang sebelumnya masih memenuhi kriteria pada *threshold* 0,15 namun tidak lagi memenuhi kriteria pada *threshold* 0,25, yaitu MMSE dan *BehavioralProblems*. Oleh karena itu, pada lampiran ini ditampilkan langkah-langkah perhitungan korelasi *pearson* secara rinci, mulai dari perhitungan rata-rata, komponen per baris, penjumlahan komponen, hingga nilai korelasi r dan | r |) menggunakan seluruh data penelitian (n = 2149), sebagai bukti numerik bahwa kedua fitur tersebut memiliki nilai $| r | < 0{,}25$sehingga dieliminasi pada *threshold* 0,25.

A. Perhitungan Manual untuk Fitur MMSE

Pada bagian ini dilakukan perhitungan manual korelasi *pearson* untuk fitur MMSE terhadap variabel target Diagnosis sebagai bagian dari proses seleksi fitur. Perhitungan ini bertujuan untuk menunjukkan secara rinci bagaimana nilai korelasi *pearson* diperoleh dari seluruh data penelitian, sehingga dapat dibuktikan apakah fitur MMSE memenuhi kriteria seleksi pada *threshold* tertentu. Hasil korelasi ini kemudian digunakan untuk menjelaskan alasan fitur MMSE masih lolos pada *threshold* 0,15 tetapi tidak lagi memenuhi syarat ketika *threshold* dinaikkan menjadi 0,25. Adapun langkah-langkah perhitungan korelasi *pearson* untuk fitur MMSE ditampilkan sebagai berikut:

1. Menentukan Data X dan Y

   Dapat diketahui bahwa :

   X : nilai fitur MMSE (menggunakan data hasil normalisasi *Min–Max*)

   Y : label Diagnosis

   n = 2149

2. Menghitung Rata-rata $\bar{X}$dan $\bar{Y}$

$$\overline{X} = \frac{\sum X_i}{n} = \frac{1057.069649}{2149} = 0.491889$$

$$\overline{Y} = \frac{\sum Y_i}{n} = \frac{760}{2149} = 0.353653$$

   Nilai $\sum X_i$ diperoleh dari penjumlahan seluruh nilai fitur MMSE hasil normalisasi *Min–Max* pada 2149 data, sedangkan $\sum Y_i$ diperoleh dari penjumlahan seluruh label Diagnosis (0/1) sehingga bernilai sama dengan jumlah sampel Alzheimer (kelas 1). Karena $\bar{Y} = 0.353653$, berarti sekitar 35.37% data merupakan Alzheimer.

3. Menghitung Komponen Per Baris

   Untuk setiap data ke-i, dihitung menggunakan Persamaan 3.3 berikut:

$$(X_i - \overline{X}), \quad (Y_i - \overline{Y}), \quad (X_i - \overline{X})(Y_i - \overline{Y}), \quad (X_i - \overline{X})^2, \quad (Y_i - \overline{Y})^2$$

   Dapat diketahui bahwa:

   $\overline{X}$=0.491889 (rata-rata MMSE)

   $\overline{Y} = 0.353653$ (rata-rata Diagnosis)

   X_1 = 0.715606 (data MMSE pada baris ke-1)

   Y_1 = 0 (data Diagnosis pada baris ke-1)

Contoh perhitungan untuk data ke-1

a. Selisih fitur

$$(X_1 - \bar{X}) = 0.715606 - 0.491889 = 0.223717$$

b. Selisih target

$$(Y_1 - \bar{Y}) = 0 - 0.353653 = -0.353653$$

c. Perkalian selisih

$$(X_1 - \bar{X})(Y_1 - \bar{Y}) = (0.223717) \times (-0.353653) = -0.079118$$

d. Kuadrat selisih fitur

$$(X_1 - \bar{X})^2 = (0.223717)^2 = 0.050049$$

e. Kuadrat selisih target

$$(Y_1 - \bar{Y})^2 = (-0.353653)^2 = 0.125070$$

Komponen ini dibentuk untuk menyusun pembilang dan penyebut dalam rumus *pearson*. Selisih $(X_i - \bar{X})$ menunjukkan deviasi nilai fitur terhadap rata-rata, sedangkan $(Y_i - \bar{Y})$ menunjukkan deviasi label kelas terhadap rata-rata label. Di bawah ini akan ditampilkan tabel komponen perbaris dengan 10 data saja sebagai perwakilan dari 2.149 data, namun perhitungan tetap dilakukan pada semua data.

Tabel L.1 Komponen Per-baris untuk Fitur MMSE

i	X_i	Y_i	$X_i - \bar{X}$	$Y_i - \bar{Y}$	$(X-\bar{X})(Y-\bar{Y})$	$(X-\bar{X})^2$	$(Y-\bar{Y})^2$
1	0.715606	0	0.223717	-0.353653	-0.079118	0.050049	0.125070
2	0.687251	0	0.195362	-0.353653	-0.069090	0.038166	0.125070
3	0.245145	0	-0.246744	-0.353653	0.087262	0.060883	0.125070
4	0.466410	0	-0.025479	-0.353653	0.009011	0.000649	0.125070
5	0.450619	0	-0.041270	-0.353653	0.014595	0.001703	0.125070
6	0.917500	0	0.425611	-0.353653	-0.150519	0.181145	0.125070
7	0.065334	0	-0.426555	-0.353653	0.150853	0.181950	0.125070
8	0.337965	1	-0.153924	0.646347	-0.099488	0.023692	0.417765
9	0.860914	0	0.369025	-0.353653	-0.130507	0.136179	0.125070
10	0.946543	0	0.454654	-0.353653	-0.160790	0.206710	0.125070

4. Menjumlahkan Komponen

   Hasil penjumlahan seluruh komponen adalah:

   a. Jumlah perkalian selisih

$$\sum(\mathrm{X_i} - \overline{\mathrm{X}})(\mathrm{Y_i} - \overline{\mathrm{Y}}) = -69.964519$$

   b. Jumlah kuadrat selisih fitur

$$\sum(\mathrm{X_i} - \overline{\mathrm{X}})^2 = 177.222710$$

   c. Jumlah kuadrat selisih target

$$\sum(\mathrm{Y_i} - \overline{\mathrm{Y}})^2 = 491.223825$$

   Nilai pembilang $\sum(\mathrm{X_i} - \bar{\mathrm{X}})(\mathrm{Y_i} - \bar{\mathrm{Y}})$ yang negatif menunjukkan adanya kecenderungan hubungan berlawanan arah antara MMSE dan diagnosis (semakin tinggi MMSE, cenderung semakin kecil peluang Alzheimer). Nilai kuadrat deviasi ($\sum(\mathrm{X_i} - \bar{\mathrm{X}})^2$ dan $\sum(\mathrm{Y_i} - \bar{\mathrm{Y}})^2$) digunakan sebagai normalisasi agar menghasilkan nilai korelasi dalam rentang $[-1,1]$.

5. Menghitung Korelasi Pearson r

$$\mathrm{r} = \frac{-69.964519}{\sqrt{177.222710 \times 491.223825}} = \frac{-69.964519}{295.052567} = -0.237126$$

   Nilai korelasi *pearson* $\mathrm{r} = -0.237126$ menunjukkan hubungan negatif antara MMSE dan diagnosis Alzheimer. Besarnya korelasi $|\,\mathrm{r}\,| = 0.237126$ menandakan bahwa kekuatan hubungan yang relatif lemah, sehingga MMSE memiliki keterkaitan terhadap diagnosis namun tidak dominan jika berdiri sendiri.

6. Interpretasi dan Hubungannya dengan *Threshold*

   - Pada *threshold* 0.15, fitur MMSE lolos karena $| r | = 0.237126 > 0.15$
   - Pada *threshold* 0.25, fitur MMSE tidak lolos karena $| r | = 0.237126 < 0.25$

Maka, MMSE termasuk fitur yang berkontribusi cukup pada *threshold* 0.15, namun tidak cukup kuat untuk dipertahankan pada *threshold* 0.25.

B. Perhitungan Manual untuk Fitur *BehavioralProblems*

Pada bagian ini dilakukan perhitungan manual korelasi *pearson* untuk fitur *BehavioralProblems* terhadap variabel target Diagnosis sebagai bagian dari proses seleksi fitur. Perhitungan ini bertujuan untuk menunjukkan secara rinci bagaimana nilai korelasi *pearson* diperoleh dari seluruh data penelitian, sehingga dapat dibuktikan apakah fitur *BehavioralProblems* memenuhi kriteria seleksi pada *threshold* tertentu. Hasil korelasi ini kemudian digunakan untuk menjelaskan alasan fitur *BehavioralProblems* masih lolos pada *threshold* 0,15 tetapi tidak lagi memenuhi syarat ketika *threshold* dinaikkan menjadi 0,25. Adapun langkah-langkah perhitungan korelasi *pearson* untuk fitur *BehavioralProblems* ditampilkan sebagai berikut:

1. Menentukan Data Xdan Y

   Dapat diketahui bahwa :

   X : nilai fitur *BehavioralProblems* (menggunakan data hasil normalisasi *Min–Max*)

   Y : label Diagnosis

   n = 2149

2. Menghitung Rata-rata $\bar{X}$ dan $\bar{Y}$

$$\overline{X} = \frac{\sum X_i}{n} = \frac{337}{2149} = 0.156817$$

$$\overline{Y} = \frac{\sum Y_i}{n} = \frac{760}{2149} = 0.353653$$

Nilai $\sum X_i$ diperoleh dari penjumlahan seluruh nilai fitur *BehavioralProblems* hasil normalisasi *Min–Max* pada 2149 data, sedangkan $\sum Y_i$ diperoleh dari penjumlahan seluruh label Diagnosis (0/1) sehingga bernilai sama dengan jumlah sampel Alzheimer (kelas 1). Karena $\bar{Y} = 0.353653$, berarti sekitar 35.37% data merupakan Alzheimer.

3. Menghitung Komponen Per Baris

Untuk setiap data ke-i, dihitung menggunakan Persamaan 3.3 berikut:

$$(X_i - \overline{X}), \quad (Y_i - \overline{Y}), \quad (X_i - \overline{X})(Y_i - \overline{Y}), \quad (X_i - \overline{X})^2, \quad (Y_i - \overline{Y})^2$$

Dapat diketahui bahwa:

$\overline{X}$= 0.156817 (rata-rata *BehavioralProblems*)

$\overline{Y} = 0.353653$ (rata-rata Diagnosis)

X_1 = 0 (data *BehavioralProblems* pada baris ke-1)

Y_1 = 0 (data Diagnosis pada baris ke-1)

Contoh perhitungan untuk data ke-1

a. Selisih fitur

$$(X_1 - \overline{X}) = 0 - 0.156817 = -0.156817$$

b. Selisih target

$$(Y_1 - \overline{Y}) = 0 - 0.353653 = -0.353653$$

c. Perkalian selisih

$$(X_1 - \bar{X})(Y_1 - \bar{Y}) = (-0.156817) \times (-0.353653) = 0.055459$$

d. Kuadrat selisih fitur

$$(X_1 - \bar{X})^2 = (-0.156817)^2 = 0.024592$$

e. Kuadrat selisih target

$$(Y_1 - \bar{Y})^2 = (-0.353653)^2 = 0.125070$$

Komponen ini dibentuk untuk menyusun pembilang dan penyebut dalam rumus *pearson*. Selisih $X_i - \bar{X}$menunjukkan deviasi nilai fitur terhadap rata-rata, sedangkan $Y_i - \bar{Y}$ menunjukkan deviasi label kelas terhadap rata-rata label. Di bawah ini akan ditampilkan tabel komponen per baris dengan 10 data saja sebagai perwakilan dari 2.149 data, namun perhitungan tetap dilakukan pada semua data.

Tabel L.2 Komponen Per-baris untuk Fitur *BehavioralProblems*

i	X_i	Y_i	$X_i - \bar{X}$	$Y_i - \bar{Y}$	$(X-\bar{X})(Y-\bar{Y})$	$(X-\bar{X})^2$	$(Y-\bar{Y})^2$
1	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
2	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
3	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
4	1.0	0	0.843183	-0.353653	-0.298194	0.710957	0.125070
5	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
6	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
7	0.0	0	-0.156817	-0.353653	0.055459	0.024592	0.125070
8	0.0	1	-0.156817	0.646347	-0.101358	0.024592	0.417765
9	1.0	0	0.843183	-0.353653	-0.298194	0.710957	0.125070
10	1.0	0	0.843183	-0.353653	-0.298194	0.710957	0.125070

4. Menjumlahkan Komponen

Hasil penjumlahan seluruh komponen adalah:

a. Jumlah perkalian selisih

$$\sum(X_i - \overline{X})(Y_i - \overline{Y}) = 83.818986$$

b. Jumlah kuadrat selisih fitur

$$\sum(X_i - \overline{X})^2 = 284.152629$$

c. Jumlah kuadrat selisih target

$$\sum(Y_i - \overline{Y})^2 = 491.223825$$

Nilai pembilang $\sum(X_i - \bar{X})(Y_i - \bar{Y})$ yang positif menunjukkan adanya kecenderungan hubungan searah antara *BehavioralProblems* dan diagnosis (semakin tinggi *BehavioralProblems*, cenderung semakin besar peluang Alzheimer). Nilai kuadrat deviasi $\sum(X_i - \bar{X})^2$dan $\sum(Y_i - \bar{Y})^2$digunakan sebagai normalisasi agar menghasilkan nilai korelasi dalam rentang $[-1,1]$.

5. Menghitung Korelasi *Pearson* r

$$r = \frac{83.818986}{\sqrt{284.152629 \times 491.223825}} = \frac{83.818986}{373.607470} = 0.224350$$

Nilai korelasi *pearson* $r = 0.224350$ menunjukkan hubungan positif antara *BehavioralProblems* dan diagnosis Alzheimer. Besarnya korelasi $|r| = 0.224350$ menandakan bahwa kekuatan hubungan yang relatif lemah, sehingga *BehavioralProblems* memiliki keterkaitan terhadap diagnosis namun tidak dominan jika berdiri sendiri.

6. Interpretasi dan Hubungannya dengan *Threshold*

a. Pada *threshold* 0.15, fitur *BehavioralProblems* lolos karena $|r| = 0.224350 > 0.15$

b. Pada *threshold* 0.25, fitur *BehavioralProblems* tidak lolos karena | r |= $0.224350 < 0.25$

   Maka, *BehavioralProblems* termasuk fitur yang berkontribusi cukup pada *threshold* 0.15, namun tidak cukup kuat untuk dipertahankan pada *threshold* 0.25.

Berdasarkan perhitungan manual korelasi *pearson* menggunakan seluruh data (n = 2149), diperoleh | r |untuk fitur MMSE sebesar 0.237126 dan untuk fitur *BehavioralProblems* sebesar 0.224350. Kedua nilai tersebut memenuhi kriteria seleksi pada *threshold* 0.15, namun tidak mencapai batas *threshold* 0.25. Dengan demikian, MMSE dan *BehavioralProblems* tereliminasi pada *threshold* 0.25 karena dianggap memiliki hubungan yang belum cukup kuat terhadap label diagnosis dibanding fitur lain yang tetap dipertahankan.

Lampiran II. Hasil Korelasi *Pearson* Seluruh Fitur terhadap Diagnosis dan Status Seleksi Berdasarkan *Threshold*

Lampiran ini memuat hasil perhitungan korelasi *pearson* antara seluruh fitur (hasil normalisasi *Min–Max*) dengan variabel target Diagnosis (0 = Non-Alzheimer, 1 = Alzheimer) menggunakan seluruh data penelitian (n = 2149). Pada lampiran ini ditampilkan nilai koefisien korelasi *pearson* (r), nilai absolut korelasi (| r |), serta status seleksi fitur pada tiga nilai *threshold* yaitu 0.0, 0.15, dan 0.25. Penentuan fitur terpilih dilakukan berdasarkan kriteria | r |≥ *threshold*. Oleh karena itu, tabel pada lampiran ini digunakan untuk memperlihatkan fitur yang lolos dan tidak lolos pada

setiap *threshold*, sekaligus menjelaskan perubahan jumlah fitur terpilih ketika nilai *threshold* dinaikkan.

Tabel L.3 Hasil Perhitungan Korelasi *Pearson* Semua Fitur

No	*Feature*	*r* *(corr_pearson)*	*|r|* *(abs_corr)*	*Threshold*		
				0.0	**0.15**	**0.25**
1	*FunctionalAssessment*	-0.364898	0.364898	Lolos	Lolos	Lolos
2	*ADL*	-0.332346	0.332346	Lolos	Lolos	Lolos
3	*MemoryComplaints*	0.306742	0.306742	Lolos	Lolos	Lolos
4	*MMSE*	-0.237126	0.237126	Lolos	Lolos	Tidak
5	*BehavioralProblems*	0.224350	0.224350	Lolos	Lolos	Tidak
6	*SleepQuality*	-0.056548	0.056548	Lolos	Tidak	Tidak
7	*EducationLevel*	-0.043966	0.043966	Lolos	Tidak	Tidak
8	*CholesterolHDL*	0.042584	0.042584	Lolos	Tidak	Tidak
9	*Hypertension*	0.035080	0.035080	Lolos	Tidak	Tidak
10	*FamilyHistoryAlzheimers*	-0.032900	0.032900	Lolos	Tidak	Tidak
11	*CholesterolLDL*	-0.031976	0.031976	Lolos	Tidak	Tidak
12	*Diabetes*	-0.031508	0.031508	Lolos	Tidak	Tidak
13	*CardiovascularDisease*	0.031490	0.031490	Lolos	Tidak	Tidak
14	*BMI*	0.026343	0.026343	Lolos	Tidak	Tidak
15	*Disorientation*	-0.024648	0.024648	Lolos	Tidak	Tidak
16	*CholesterolTriglycerides*	0.022672	0.022672	Lolos	Tidak	Tidak
17	*HeadInjury*	-0.021411	0.021411	Lolos	Tidak	Tidak
18	*Gender*	-0.020975	0.020975	Lolos	Tidak	Tidak
19	*PersonalityChanges*	-0.020627	0.020627	Lolos	Tidak	Tidak
20	*Confusion*	-0.019186	0.019186	Lolos	Tidak	Tidak
21	*SystolicBP*	-0.015615	0.015615	Lolos	Tidak	Tidak
22	*Ethnicity*	-0.014782	0.014782	Lolos	Tidak	Tidak
23	*DifficultyCompletingTasks*	0.009069	0.009069	Lolos	Tidak	Tidak
24	*DietQuality*	0.008506	0.008506	Lolos	Tidak	Tidak
25	*AlcoholConsumption*	-0.007618	0.007618	Lolos	Tidak	Tidak
26	*CholesterolTotal*	0.006394	0.006394	Lolos	Tidak	Tidak
27	*PhysicalActivity*	0.005945	0.005945	Lolos	Tidak	Tidak
28	*Depression*	-0.005893	0.005893	Lolos	Tidak	Tidak
29	*Age*	-0.005488	0.005488	Lolos	Tidak	Tidak
30	*DiastolicBP*	0.005293	0.005293	Lolos	Tidak	Tidak
31	*Smoking*	-0.004865	0.004865	Lolos	Tidak	Tidak
32	*Forgetfulness*	-0.000354	0.000354	Lolos	Tidak	Tidak

Berdasarkan Tabel L.3, pada *threshold* 0.0 seluruh fitur dinyatakan lolos karena semua fitur memenuhi $|r| \geq 0.0$, sehingga seluruh fitur tetap digunakan. Ketika *threshold* dinaikkan menjadi 0.15, hanya fitur yang memiliki $|r| \geq 0.15$ yang dipertahankan, yaitu F*unctionalAssessment, ADL, MemoryComplaints, MMSE,* dan *BehavioralProblems* (total 5 fitur). Selanjutnya pada *threshold* 0.25,

seleksi menjadi lebih ketat sehingga hanya fitur dengan $|\ r\ | \geq 0.25$ yang lolos, yaitu *FunctionalAssessment, ADL,* dan *MemoryComplaints* (total 3 fitur). Dengan demikian, terdapat dua fitur yang tereliminasi saat *threshold* berubah dari 0.15 ke 0.25 yaitu *MMSE* ($|\ r\ | = 0.237126$) dan *BehavioralProblems* ($|\ r\ | = 0.224350$), karena nilai korelasi absolut keduanya berada di bawah 0.25. Hasil pada Lampiran II ini memperkuat pembahasan pada Lampiran I yang menampilkan proses perhitungan manual untuk fitur-fitur yang tereliminasi tersebut.