

BAB I

PENDAHULUAN

1.1 Latar Belakang

Obesitas menjadi salah satu penyebab masalah kesehatan terbesar, yang mempengaruhi jutaan orang dari segala usia dan latar belakang. Obesitas tidak hanya menaikkan risiko penyakit kronis seperti diabetes, hipertensi, dan penyakit jantung, namun juga berdampak signifikan terhadap kualitas hidup seseorang. Ada beberapa periode kritis yang diakibatkan oleh obesitas, antara lain pada masa perkembangan janin, masa bayi, masa perkembangan anak dimana jumlah lemak tumbuh mulai meningkat dan masa remaja. Saat masa remaja terjadi obesitas 30%, saat dewasa akan terjadi obesitas parsisten. Oleh karena itu, perlunya ilmu pengetahuan dan keterampilan agar bisa mencegah terjadinya masalah obesitas dari anak-anak hingga dewasa[1]

Di Indonesia sendiri, obesitas dipengaruhi salah satunya makanan cepat saji yang memiliki kandungan lemak, gula, garam secara berlebihan. Obesitas juga terjadi karena konsumsi kalori yang tidak sebanding dengan pengeluaran kalori seperti olahraga. Menurut Riset Kesehatan Dasar (Riskesdas) pada tahun 2007 terdapat 11,7% orang dewasa yang memiliki umur kurang dari 18 tahun mengalami obesitas dan berdasarkan riset yang dilakukan terdapat peningkatan pada tahun 2010 sekitar 11,65% [2].

Pada tahun 2022 *World Health Organization* melaporkan, sekitar 2,5 miliar orang dewasa dengan umur lebih dari 18 tahun mengalami obesitas dan lebih dari 890 juta orang dewasa yang mengalami obesitas. Hal tersebut sama dengan 43% orang berusia lebih dari 18 tahun atau 43% pria dan 44% wanita yang mengalami obesitas. Hal ini mengalami peningkatan dari tahun 1990, saat orang dewasa yang berusia lebih dari 18 tahun mengalami obesitas sekitar 25%[3]. Karena peningkatan kasus obesitas ini diperlukan pengembangan metode yang lebih efisien untuk memprediksi tingkat obesitas yaitu dengan machine learning.

Dalam upaya mendukung transformasi digital, pengambilan keputusan terkait program vaksinasi, sanitasi, dan program kesehatan lainnya harus didukung oleh alat yang dapat mengolah data untuk memberikan wawasan prediktif. *Machine Learning* dapat mengoptimalkan proses pengambilan keputusan dengan

menyediakan wawasan prediktif dan berjalan secara otomatis. Pembelajaran mesin (*Machine Learning*) adalah bidang studi yang terus berkembang, fokus pada konsep pengenalan pola serta pembelajaran komputasi dalam kecerdasan buatan. *Machine Learning* memanfaatkan algoritma pembelajaran, baik yang bersifat terawasi maupun yang tidak terawasi untuk melakukan prediksi dan mendukung pengambilan keputusan otomatis berdasarkan data[4].

Dalam machine learning, terdapat teknik klasifikasi yang digunakan untuk memprediksi dan mengelompokkan data ke dalam kelas-kelas tertentu berdasarkan karakteristik atau fitur yang serupa[5]. Klasifikasi adalah teknik yang memprediksi kelas pada label yang sudah ditentukan sebelumnya[6]. Klasifikasi juga bisa diartikan sebagai salah satu proses yang bertujuan untuk menemukan pola yang berharga dari data yang berukuran besar[7]. Salah satu jenis klasifikasi yaitu *decision tree*, metode ini merupakan metode yang bertujuan untuk menggambarkan setiap kelas guna menemukan pola atau fungsi yang membantu dalam mengklasifikasikan atau memprediksi data yang belum memiliki klasifikasi. Teknik ini sangat diminati karena dapat diterapkan di berbagai bidang[8]. Kelebihan dari algoritma ini adalah mudah dipahami, mampu menangani data numerik dan kategorikal, serta dapat mengekstraksi informasi penting dari data. Namun, algoritma ini juga memiliki kelemahan, seperti kecenderungan untuk mengalami *overfitting* dan kurang stabil terhadap perubahan data input[9]. Untuk mengatasi *overfitting* ini dibutuhkan *ensemble learning* untuk menggabungkan berbagai model, kelemahan dari satu model dapat diimbangi oleh keunggulan model lainnya.

Beberapa penelitian sebelumnya telah membahas penerapan metode K-Nearest Neighbor (KNN) dalam mengklasifikasikan tingkat obesitas. Proses klasifikasi dilakukan melalui beberapa tahapan, antara lain seleksi fitur, transformasi data, serta normalisasi. Dataset yang digunakan bersumber dari UCI Machine Learning Repository, yang terdiri atas 2.111 entri dengan 16 atribut serta 1 label klasifikasi. Eksperimen dilakukan dengan memvariasikan nilai k pada algoritma KNN, yaitu $k=1, 3, 5, 7$, dan 9 . Selain itu, pengujian juga dilakukan menggunakan skema validasi silang sebanyak 3-fold, 5-fold, dan 7-fold. Penelitian ini juga mengevaluasi kombinasi atribut yang digunakan untuk mengetahui pengaruhnya terhadap hasil klasifikasi. Dari hasil pengujian, diperoleh bahwa

konfigurasi model paling optimal terjadi saat menggunakan nilai $k=1$, dengan atribut yang paling berkontribusi terhadap performa model adalah berat badan dan usia. Akurasi tertinggi yang berhasil dicapai mencapai 79,96%. Temuan ini menunjukkan bahwa metode KNN cukup efektif untuk digunakan dalam tugas klasifikasi tingkat obesitas[10].

Algoritma *decision tree* bekerja dengan memodelkan struktur pohon yang memiliki atribut atau fitur dengan hubungan yang berbeda. Sebaliknya, algoritma random forest merupakan kumpulan proses dari banyak *decision tree* yang diproses secara paralel untuk meningkatkan akurasi dan stabilitas model[4]. Metode *decision tree* merupakan metode yang bertujuan untuk menggambarkan setiap kelas guna menemukan pola atau fungsi yang membantu dalam mengklasifikasikan atau memprediksi data yang belum memiliki klasifikasi. Teknik ini sangat diminati karena dapat diterapkan di berbagai bidang[8]. Algoritma *Decision Tree* memiliki sejumlah keunggulan, seperti interpretabilitas tinggi dan kemampuan untuk menangani beragam tipe data. Namun, kelemahan utamanya adalah kecenderungan untuk *overfitting*, khususnya pada dataset yang kompleks. *Overfitting* terjadi ketika model terlalu menyesuaikan diri dengan data pelatihan sehingga mengurangi kemampuannya untuk menggeneralisasi pada data baru[11].

Sementara itu, *random forest* adalah salah satu algoritma yang berhasil mengatasi volume informasi yang besar sambil tetap mempertahankan efisiensi statistiknya. Algoritma *Random Forest* adalah *ensemble classifier* yang digunakan untuk meningkatkan akurasi model. Metode *Random Forest* memiliki tingkat kesalahan yang lebih rendah dibandingkan dengan algoritma klasifikasi lainnya[12]. Metode *Random Forest* memiliki beberapa kelemahan, yaitu komputasi yang tinggi, risiko *overfitting*, kecepatan pembelajaran yang lambat, dan ukuran model yang dihasilkan cukup besar. Kelemahan-kelemahan ini perlu diperhatikan dan diatasi untuk memastikan performa optimal dari model yang digunakan[13].

Untuk mengatasi beberapa kelemahan dari metode-metode tersebut, algoritma *XGBoost* dapat menjadi solusi yang lebih efisien dan efektif. *XGBoost* atau *Extreme Gradient Boosting*, merupakan implementasi lanjutan dari algoritma *gradient boosting* yang menggunakan *decision tree* sebagai basis klasifikasinya. Algoritma ini dirancang untuk mengoptimalkan kinerja model dengan

menggabungkan beberapa pohon keputusan (*decision trees*) secara bertahap, di mana setiap model baru memperbaiki kesalahan dari model sebelumnya. Teknik ini memungkinkan *XGBoost* untuk menghasilkan model yang lebih akurat dan mampu mengatasi *overfitting* dengan lebih baik. *XGBoost* memiliki keunggulan utama dalam hal efisiensi waktu pelatihan dan kemampuan menangani hubungan nonlinear yang kompleks melalui pendekatan boosting berbasis pohon. Model ini sangat efektif dalam menangani *outlier* dan data berukuran besar dengan tingkat akurasi yang tinggi. Meskipun demikian, *XGBoost* dapat mengalami risiko *overfitting* jika tuning parameter tidak dilakukan dengan tepat[14].

Pada penelitian lainnya yang menggunakan dataset *heart failure clinical records* yang tidak seimbang, yang dapat menurunkan kinerja model prediksi karena cenderung menghasilkan prediksi kelas mayoritas. Untuk mengatasi ketidakseimbangan kelas, digunakan teknik boosting dengan metode *XGBoost*, yang dikombinasikan dengan metode *hyper-parameter tuning* untuk meningkatkan kinerja model. Model kemudian dilatih dengan dataset dan dibandingkan dengan metode lain. Hasil penelitian menunjukkan bahwa *XGBoost* dengan *tree parzen estimator hyper-parameter tuning* mencapai nilai AUC sebesar 0.94, sementara *XGBoost* dengan *Random Search* memperoleh nilai AUC sebesar 0.933, *XGBoost* dengan *Grid Search* memperoleh nilai AUC sebesar 0.922, dan *XGBoost* tanpa *hyper-parameter tuning* memperoleh nilai AUC sebesar 0.904[15].

Pada penelitian sebelumnya dengan metode yang sama yaitu mengklasifikasikan nasabah kartu kredit yang mengalami keterlambatan pembayaran dengan menggunakan teknik *machine learning*, khususnya teknik klasifikasi. Salah satu metode klasifikasi yang digunakan adalah *XGBoost*, yang berguna untuk analisis regresi dan klasifikasi berdasarkan *Gradient Boosting Decision Tree* (GBDT). *XGBoost* memiliki sejumlah *hyperparameter* yang dapat dikonfigurasi untuk meningkatkan kinerja model. Metode *tuning hyperparameter* yang digunakan adalah *Grid Search cross-validation*, yang divalidasi menggunakan *10-Fold Cross-Validation*. *Hyperparameter* yang dikonfigurasi termasuk *n_estimators*, *max_depth*, *subsample*, *gamma*, *colsample_bylevel*, *min_child_weight*, dan *learning_rate*. Hasil penelitian ini menunjukkan bahwa penggunaan algoritma dengan tuning *hyperparameter* dapat meningkatkan kinerja

algoritma *Extreme Gradient Boosting* dalam klasifikasi nasabah kartu kredit, dengan akurasi mencapai 80,039%, presisi sebesar 81,338%, dan nilai *recall* sebesar 96,854%[16].

Model *ensembel* sangat penting karena terdiri dari pengelompokan atau penumpukan model-model yang berbeda untuk memperoleh hasil yang lebih kuat, yang mengonfirmasi atau melengkapi hasil akhir dari eksperimen. Pendekatan ensambel atau kontribusi dalam peningkatan kinerja dengan model yang lebih baik atau penanganan bias, menyediakan alat untuk mengelompokkan dan meningkatkan model guna menghasilkan hasil yang lebih baik atau lebih kuat [17]. Hal ini dilakukan karena setiap model memiliki kelebihan dan kekurangan masing-masing, dan dengan menggabungkannya, kita bisa mendapatkan hasil akhir yang lebih akurat dibandingkan dengan menggunakan satu metode saja[18]. Penggunaan teknik *ensemble* seperti *bagging*, *boosting*, dan *stacking* dalam penelitian telah terbukti mampu meningkatkan akurasi algoritma yang diterapkan[19]. Ensemble adalah teknik yang menggabungkan berbagai model *Decision Tree* yang dibentuk melalui subsampling data untuk meningkatkan akurasi prediksi.

Selanjutnya, Untuk mengatasi ketidakseimbangan data, ada dua pendekatan utama yang dapat digunakan. Pendekatan pertama adalah pada level data, yaitu dengan melakukan resampling. Pendekatan kedua adalah pada level algoritma, yaitu dengan menggunakan teknik *bagging* dan *boosting*. Salah satu cara untuk menangani masalah ketidakseimbangan kelas (*class imbalance*) adalah dengan menggunakan teknik *ensemble*, seperti *boosting* dan *bagging*[20].

Bagging dan *boosting* adalah teknik populer yang digunakan untuk meningkatkan aturan prediksi. Kedua teknik ini merupakan contoh dari metode *perturb and combine* (P&C) yang disebut oleh Breiman (1998), di mana metode klasifikasi atau regresi diterapkan pada berbagai gangguan dari set data asli, kemudian hasilnya digabungkan untuk menghasilkan satu model klasifikasi atau regresi. *Bagging* dan *boosting* dapat diterapkan pada metode berbasis pohon untuk meningkatkan akurasi prediksi, meskipun perlu dicatat bahwa kedua teknik ini juga dapat digunakan dengan metode lain selain metode berbasis pohon, seperti jaringan saraf (*neural networks*)[21].

Boosting memiliki kemiripan dengan *bagging*, *bagging* adalah pendekatan berbasis komite yang bertujuan meningkatkan akurasi metode klasifikasi atau regresi. Namun, berbeda dengan *bagging* yang menggunakan perataan sederhana dari hasil untuk mendapatkan prediksi keseluruhan, *boosting* menggunakan perataan berbobot hasil yang diperoleh dari penerapan metode prediksi pada berbagai sampel. Selain itu, pada *boosting*, sampel yang digunakan pada setiap langkah tidak semuanya diambil dengan cara yang sama dari populasi yang sama. Sebaliknya, kasus yang diprediksi salah pada langkah sebelumnya diberikan bobot yang lebih besar pada langkah berikutnya. Oleh karena itu, *boosting* adalah prosedur iteratif yang menggabungkan bobot, berbeda dengan perataan sederhana yang digunakan dalam *bagging*. Selain itu, *boosting* sering kali diterapkan pada pembelajar lemah, seperti *two node decision tree*, sementara hal ini tidak berlaku untuk *bagging*[21].

Maka dari itu diperlukannya *XGBoost*, *XGBoost* adalah salah satu implementasi lanjutan dari algoritma *boosting* yang menggunakan pohon keputusan sebagai basisnya. Metode ini merupakan metode yang bertujuan untuk menggambarkan setiap kelas guna menemukan pola atau fungsi yang membantu dalam mengklasifikasikan atau memprediksi data yang belum memiliki klasifikasi. Teknik ini sangat diminati karena dapat diterapkan di berbagai bidang. Metode *XGBoost* merupakan pengembangan dari algoritma *gradient tree boosting* yang berbasis pada algoritma *ensemble*. Algoritma ini efektif dalam menangani kasus *machine learning* berskala besar. *XGBoost* dipilih karena memiliki fitur tambahan yang dapat mempercepat proses perhitungan dan mengurangi risiko *overfitting*. *XGBoost* mampu menyelesaikan berbagai tugas seperti klasifikasi, regresi, dan *ranking*[22].

XGBoost adalah algoritma dengan banyak parameter penting yang mempengaruhi kinerja model. Konfigurasi parameter yang tepat dapat meningkatkan performa model. Namun, performa model *XGBoost* sangat bergantung pada pemilihan *hyperparameter* yang optimal. Oleh karena itu, diperlukan metode pencarian *hyperparameter* yang tepat untuk meningkatkan akurasi klasifikasi obesitas. Salah satu metode *tuning hyper-parameter* adalah pencarian manual, tetapi ini memakan waktu terutama untuk model dengan banyak

parameter. Dalam penelitian ini, digunakan metode *tuning hyper-parameter* seperti *Grid Search*[23].

Dalam proses *machine learning*, khususnya klasifikasi, umumnya melibatkan pengujian berbagai model terhadap dataset dan memilih model dengan kinerja terbaik. Untuk menemukan kombinasi *hyperparameter* yang optimal dengan cepat, penelitian ini menggunakan metode *Hyperparameter Tuning Grid Search*[24]. Metode *Grid Search* ini melakukan perhitungan pada setiap titik sampel yang terbentuk berdasarkan grid. Salah satu metodologi yang populer adalah *Grid Search*[25]. Meskipun terkadang metode ini tidak efisien dan bahkan dianggap tidak layak digunakan, *Grid Search* tetap menjadi solusi umum. Metode ini lebih efektif daripada mengubah satu *hyperparameter* pada satu waktu dan mengukur dampaknya terhadap kinerja model, yang dapat menjadi tidak efisien dan tidak menjamin hasil optimal karena mengabaikan interaksi antar *hyperparameter*[26].

Penelitian ini menggunakan metode *gradient boosting* sebagai metode utamanya. Menurut studi yang dilakukan oleh Khe Foon Hew, *gradient boosting* adalah *classifier* dengan performa terbaik untuk klasifikasi. *Gradient boosting* memiliki berbagai parameter yang memerlukan *tuning hyperparameter* untuk menemukan konfigurasi yang paling optimal. Penerapan teknik ini tidak hanya meningkatkan akurasi, tetapi juga menghasilkan model yang lebih andal dan konsisten[27].

Penelitian menggunakan metode *XGBoost* dengan optimasi *hyperparameter* menggunakan *Grid Search* dengan menawarkan solusi yang lebih baik dalam meningkatkan akurasi prediksi melalui penggabungan hasil dari beberapa model. Dengan menggunakan teknik ensemble untuk mengoptimalkan performa algoritma Hasil yang dapat memberikan wawasan baru dalam penerapan metode *Machine Learning* dalam bidang kesehatan, khususnya dalam deteksi obesitas berbasis data. Pada penelitian ini menggunakan dataset dari archive.ics.uci.edu sebanyak 2111 data. Dengan demikian, penelitian ini tidak hanya berkontribusi dalam ranah akademik, tetapi juga dapat memberikan manfaat praktis bagi tenaga medis dan masyarakat dalam upaya pencegahan obesitas dan penyakit terkait.

1.2 Rumusan masalah

1.2.1 Permasalahan

Pada penelitian ini, proses optimasi dilakukan dengan menggunakan metode *Grid Search* (GS) untuk memperoleh kombinasi *hyperparameter* terbaik pada model *XGBoost*. *Hyperparameter* yang dioptimalkan mencakup *n_estimators* yang memiliki 4 nilai, *max_depth* memiliki 5 nilai, *learning_rate* memiliki 5 nilai, dan *subsample* memiliki 4 nilai, parameter-parameter tersebut memiliki pengaruh yang signifikan terhadap kinerja model dalam melakukan klasifikasi tingkat obesitas. Optimasi dilakukan dengan menetapkan rentang nilai tertentu untuk setiap *hyperparameter*, kemudian mengevaluasi seluruh kombinasi berdasarkan *metriks* akurasi. Setiap kombinasi pada parameter diuji menggunakan validasi silang untuk memastikan generalisasikan model terhadap data baru. Melalui pendekaran ini diperoleh konfigurasi *hyperparameter* yang mampu menghasilkan performa klasifikasi terbaik dan lebih optimal dibandingkan penggunaan nilai default atau pemilihan manual.

1.2.1 Metode Usulan

Metode ini akan menggunakan metode *XGBoost* untuk mengatasi *overfitting* sehingga diperlukan strategi yang sistematis dalam penerapan dan evaluasi model. Metode yang diterapkan pada penelitian ini meliputi beberapa tahapan utama, yaitu pengumpulan data, pra-pemrosesan data, pembangunan model, optimasi *hyperparameter*, evaluasi model, dan analisis hasil untuk memperoleh performa terbaik.

1.2.2 Pertanyaan Penelitian

Berdasarkan permasalahan diatas, maka pertanyaan penelitian adalah berapa nilai kinerja optimasi *hyperparameter* pada *XGBoost* menggunakan *Grid Search* terhadap akurasi klasifikasi level obesitas?

1.3 Tujuan dan manfaat

Berdasarkan rumusan masalah diatas penelitian ini memiliki tujuan dan manfaat sebagai berikut :

1.3.1 Tujuan

Menganalisis nilai kinerja optimasi *hyperparameter XGBoost* menggunakan *Grid Search* terhadap akurasi klasifikasi level obesitas.

1.3.2 Manfaat

Dapat memberikan alternatif metode berbasis *Machine Learning* untuk mendukung pengambilan keputusan medis, serta dapat memberikan wawasan mengenai *tuning hyperparameter* dalam meningkatkan akurasi model klasifikasi.

1.4 Batasan masalah

Batasan-batasan masalah yang terdapat pada penelitian ini adalah :

1. Pada penelitian ini menggunakan dataset dari UCI *Machine Learning Repository* (<https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition>).
2. Dataset tersebut terdiri dari 2111 *record* data yang terdiri dari 17 atribut dan terdapat 7 label/kelas.
3. Data ini dikumpulkan dari individu di negara yang berbeda-beda seperti Meksiko, Peru, dan Kolombia dengan kisaran umur 14 tahun sampai 61 tahun.
4. Untuk data model diambil dari data yang diambil sendiri melalui halaman website google form yang akan digunakan untuk pengujian model.
5. Atribut yang digunakan disini berupa Jenis kelamin, Usia, Tinggi Badan, Berat Badan, Riwayat keluarga dengan kelebihan berat badan, Konsumsi makanan tinggi kalori secara sering (FAVC), Frekuensi konsumsi sayuran (FCVC), Jumlah makanan utama (NCP), Konsumsi makanan diantara waktu makan (CAEC), Merokok, Konsumsi air harian (CH20), Pemantauan konsumsi kalori (SCC), Frekuensi aktifitas fisik (FAF), Waktu penggunaan perangkat teknologi (TUE), Konsumsi alkohol (CALC), Transportasi yang digunakan (MTRANS), dan dengan label variabel kelas NObesity (Tingkat Obesitas), yang memungkinkan pengklasifikasian data menggunakan nilai Berat Badan Kurang, Berat Badan Normal, Kelebihan Berat Badan Level I, Kelebihan Berat Badan Level II, Obesitas Tipe I, Obesitas Tipe II, dan Obesitas Tipe III.

6. Penelitian ini menggunakan algoritma *XGBoost* sebagai metode utama dalam klasifikasi level obesitas dan proses optimasi dilakukan menggunakan teknik *Grid Search* untuk menemukan kombinasi parameter terbaik guna meningkatkan akurasi model.

1.5 Sistematika Penulisan

Sistematika penulisan terdiri dari 4 bab sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini mencakup latar belakang, rumusan masalah, tujuan dan manfaat, batasan penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi teori-teori tentang penelitian dan penelitian yang sudah dilakukan sebelumnya.

BAB III METODELOGI PENELITIAN

Pada bab ini berisi metodologi penelitian, alur yang dipakai pada penelitian ini mulai dari prosesing data hingga evaluasi hasil.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil dari implementasi metode yang telah diusulkan dengan menampilkan data hasil eksperimen, visualisasi (grafik, *confution matriks*, dan sebagainya), analisis peforma model, serta tampilan streamlit.

BAB V PENUTUP

Pada bab ini berisi kesimpulan dari penelitian yang telah dilakukan berdasarkan hasil dan pembahasan. Selain itu, disampaikan saran untuk pengembangan lebih lanjut dari penelitian untuk memperbaiki atau melengkapi penelitian.