

**KLASIFIKASI TINGKAT PEMBANGUNAN MANUSIA DARI  
PROVINSI – KABUPATEN - KOTA**

**WILLY BOEN**

**2702749733**



**PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA  
BINUS GRADUATE PROGRAM  
UNIVERSITAS BINA NUSANTARA  
JAKARTA  
TAHUN 2025/2026**

## DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR .....	iii
DAFTAR TABEL.....	iv
BAB I PENDAHULUAN.....	1
1.1    Latar Belakang Masalah .....	1
1.2    Tujuan dan Ruang Lingkup .....	2
BAB II PEMAHAMAN DATA DAN PRA PEMROSESAN DATA .....	3
2.1    Deskripsi dan Eksplorasi Data .....	3
2.2    Pra-pemrosesan Data .....	9
2.3    Penentuan Model .....	10
BAB III PENGEMBANGAN MODEL DAN EVALUASI.....	12
3.1    Arsitektur Model.....	12
3.2    Pelatihan Model .....	13
3.3 <i>Hyperparameter Tuning</i> atau <i>Fine Tuning</i> .....	14
3.4    Hasil Evaluasi Model.....	15
3.5    Analisis .....	16
BAB IV KESIMPULAN DAN SARAN .....	34
4.1    Kesimpulan .....	34
4.2    Saran .....	34
4.3    Lampiran .....	35
DAFTAR PUSTAKA .....	36

## **DAFTAR GAMBAR**

Gambar 2.1 Matriks Korelasi.....	6
Gambar 2.2 Distribusi Atribut .....	7
Gambar 2.3 Alur Praproses Data .....	9
Gambar 3.1 Matriks Evaluasi Decision Tree: SMOTE (a) vs Non-SMOTE (b).....	17
Gambar 3.2 Visualisasi Prediksi Decision Tree: SMOTE (a) vs Non-SMOTE (b) .....	18
Gambar 3.3 Matriks Evaluasi Random Forest: SMOTE (a) vs Non-SMOTE (b).....	19
Gambar 3.4 Visualisasi Prediksi Random Forest: SMOTE (a) vs Non-SMOTE (b) .....	20
Gambar 3.5 Matriks Evaluasi Gradient Boosting: SMOTE (a) vs Non-SMOTE (b).....	21
Gambar 3.6 Visualisasi Prediksi Gradient Boosting: SMOTE (a) vs Non-SMOTE (b) .....	22
Gambar 3.7 Matriks Evaluasi AdaBoost: SMOTE (a) vs Non-SMOTE (b) .....	23
Gambar 3.8 Visualisasi Prediksi AdaBoost: SMOTE (a) vs Non-SMOTE (b).....	24
Gambar 3.9 Matriks Evaluasi SVM: SMOTE (a) vs Non-SMOTE (b).....	25
Gambar 3.10 Visualisasi Prediksi SVM: SMOTE (a) vs Non-SMOTE (b) .....	26
Gambar 3.11 Matriks Evaluasi ANN: SMOTE (a) vs Non-SMOTE (b).....	27
Gambar 3.12 Visualisasi Prediksi ANN: SMOTE (a) vs Non-SMOTE (b) .....	28
Gambar 3.13 Matriks Evaluasi Logistic Regression: SMOTE (a) vs Non-SMOTE (b) .....	29
Gambar 3.14 Visualisasi Prediksi Logistic Regression: SMOTE (a) vs Non-SMOTE (b).....	30

## **DAFTAR TABEL**

Tabel 2.1 Statistik Deskriptif .....	4
Tabel 3.1 Laporan Klasifikasi Decision Tree: SMOTE vs Non-SMOTE .....	16
Tabel 3.2 Laporan Klasifikasi Random Forest: SMOTE vs Non-SMOTE .....	18
Tabel 3.3 Laporan Klasifikasi Gradient Boosting: SMOTE vs Non-SMOTE .....	20
Tabel 3.4 Laporan Klasifikasi AdaBoost: SMOTE vs Non-SMOTE.....	22
Tabel 3.5 Laporan Klasifikasi SVM: SMOTE vs Non-SMOTE .....	25
Tabel 3.6 Laporan Klasifikasi ANN: SMOTE vs Non-SMOTE .....	27
Tabel 3.7 Laporan Klasifikasi Logistic Regression: SMOTE vs Non-SMOTE.....	29
Tabel 3.8 Akurasi, Log-loss, Weighted-F1: SMOTE vs Non-SMOTE.....	31
Tabel 3.9 Tabel Peringkat Model.....	32

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang Masalah**

Indeks Pembangunan Manusia (IPM) merupakan salah satu indikator penting yang digunakan untuk mengukur kualitas pembangunan suatu wilayah melalui tiga dimensi utama, yaitu kesehatan, pendidikan, dan standar hidup layak. Di Indonesia, IPM menjadi indikator strategis bagi pemerintah dalam merumuskan kebijakan pembangunan daerah, karena perbedaan capaian IPM antar wilayah dapat mengindikasikan adanya kesenjangan pembangunan yang perlu ditangani secara komprehensif (BPS, t.t). Pemantauan perkembangan IPM secara akurat dan berkelanjutan sangat penting, terutama dalam mendukung perencanaan program pembangunan dan evaluasi capaian kinerja pemerintah.

Seiring meningkatnya ketersediaan data pembangunan setiap tahun, penggunaan pendekatan analitik modern seperti Machine Learning menjadi semakin relevan dalam memprediksi atau mengklasifikasikan tingkat IPM secara lebih efektif. Machine Learning menawarkan kemampuan untuk mempelajari pola kompleks yang tidak selalu terlihat melalui analisis statistik konvensional (Géron, 2022). Dalam konteks data sosial dan pembangunan, teknik klasifikasi telah banyak digunakan untuk mengkategorikan wilayah berdasarkan status pencapaian pembangunan sehingga dapat membantu pengambilan keputusan berbasis data.

Namun, salah satu tantangan dalam pengolahan data IPM adalah adanya ketidakseimbangan kelas, perbedaan skala antar fitur, serta nilai-nilai yang hilang dari data tahunan. Oleh karena itu, diperlukan proses preprocessing yang tepat, seperti interpolasi nilai hilang, standarisasi fitur menggunakan StandardScaler, serta penyeimbangan kelas dengan teknik oversampling seperti SMOTE untuk meningkatkan kualitas pelatihan model (Aggarwal, 2023). Selain itu, data kategorikal seperti tingkat IPM harus dikonversi ke bentuk numerik karena algoritma Machine Learning tidak dapat memproses nilai kategorikal secara langsung tanpa representasi numerik yang terstruktur (Han et al., 2021).

Berbagai algoritma pembelajaran mesin seperti Decision Tree, Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine, Multi-Layer Perceptron, dan Multinomial Logistic Regression telah terbukti efektif dalam menyelesaikan permasalahan klasifikasi multikelas pada data tabular (Géron, 2022). Dengan

membandingkan performa beragam model tersebut, penelitian ini diharapkan dapat memberikan gambaran komprehensif mengenai model mana yang paling sesuai dalam mengklasifikasikan tingkat IPM berdasarkan variabel kesehatan, pendidikan, dan ekonomi.

Dengan demikian, eksperimen ini tidak hanya berkontribusi pada pengembangan metode analitik untuk sektor pembangunan, tetapi juga dapat menjadi dasar bagi penyusunan strategi kebijakan berbasis data yang lebih tepat sasaran dan efektif dalam meningkatkan kualitas pembangunan manusia di Indonesia.

## **1.2 Tujuan dan Ruang Lingkup**

### **1.2.1 Tujuan**

- 1.2.1.1 Mengklasifikasikan tingkat Indeks Pembangunan Manusia (IPM) dengan menggunakan model pembelajaran mesin
- 1.2.1.2 Mengevaluasi performa model pembelajaran mesin
- 1.2.1.3 Menganalisis pengaruh prapemrosesan data
- 1.2.1.4 Mengetahui model mana yang menunjukkan kinerja terbaik
- 1.2.1.5 Menyediakan gambaran komparatif

### **1.2.2 Ruang Lingkup**

Ruang lingkup eksperimen ini mencakup beberapa tahapan utama. Pertama, eksperimen difokuskan pada proses pengolahan dataset IPM dari tahun 2014 hingga 2024. Ruang lingkup pelatihan model mencakup tujuh algoritma pembelajaran mesin, yaitu Decision Tree, Random Forest, Gradient Boosting, AdaBoost, SVM, MLP, dan Multinomial Logistic Regression. Evaluasi dilakukan menggunakan metrik akurasi, weighted F1-score, log-loss (untuk model yang mendukung probabilitas), serta Matriks Evaluasi Klasifikasi untuk memeriksa kinerja per kelas. Penelitian ini tidak mencakup hyperparameter tuning, analisis spasial IPM, penggunaan model deep learning lanjutan, maupun integrasi data eksternal di luar variabel IPM. Dengan pembatasan tersebut, penelitian ini diarahkan untuk fokus pada eksperimen klasifikasi multikelas pada data tabular menggunakan metode pembelajaran mesin konvensional.

## **BAB II**

### **PEMAHAMAN DATA DAN PRA PEMROSESAN DATA**

#### **2.1 Deskripsi dan Eksplorasi Data**

##### **2.1.1 Sumber Data**

Sumber data berasal dari website Badan Pusat Statistik Indonesia (BPS). Data yang diambil Adalah data Indeks Pembangunan Manusia (IPM), data Umur Harapan Hidup (UHH), data Rata-Rata Lama Sekolah (RLS), data Harapan Lama Sekolah (HLS), data Pengeluaran per Kapita (PPP). Semuanya digunakan sesuai dengan *Booklet IPM* yang diberikan oleh BPS, untuk memenuhi indeks kesehatan, pengetahuan, dan standar hidup layak.

##### **2.1.2 Jenis Data dan Karakteristiknya**

Data yang digunakan dalam penelitian ini merupakan data sekunder berbentuk deret waktu tahunan yang berasal dari indikator pembangunan daerah. Setiap file data mewakili satu tahun pengamatan dan berisi variabel kuantitatif. Seluruh variabel bersifat numerik. Data memiliki karakteristik khas berupa keberadaan simbol seperti “–” yang menandakan data hilang, dan distribusi kelas IPM yang tidak seimbang setelah proses pelabelan. Selain itu, adanya pemekaran wilayah membuat semua wilayah yang terkena pemekaran tidak terdapat data.

##### **2.1.3 Ulasan Kualitas Data**

Kualitas data yang digunakan pada penelitian ini secara umum berada pada kategori cukup baik, tetapi masih ditemukan beberapa permasalahan yang memerlukan penanganan khusus sebelum dapat digunakan sebagai input model pembelajaran mesin. Beberapa file tahunan menunjukkan adanya nilai hilang yang ditandai dengan simbol “–”, karena adanya pemekaran wilayah. nilai yang hilang ini berpotensi menurunkan akurasi analisis deskriptif maupun performa model klasifikasi apabila tidak ditangani dengan benar. Selain itu, distribusi kelas IPM yang dihasilkan setelah proses pelabelan menunjukkan ketidakseimbangan yang cukup signifikan, di mana beberapa kelas seperti sedang dan tinggi memiliki jumlah sampel lebih banyak dibanding kelas lainnya. Ketidakseimbangan ini dapat mengakibatkan model cenderung belajar kelas dominan. Meskipun demikian, ketika dilakukan pembersihan data, interpolasi linier, standar-isasi fitur, dan penggunaan teknik oversampling, kualitas data menjadi lebih stabil.

Dengan demikian, data hasil pra-pemrosesan dapat dianggap memenuhi syarat untuk digunakan dalam pembangunan model prediksi IPM secara akurat dan reliabel.

#### 2.1.4 Eksplorasi Analisis Data

Eksplorasi analisis data adalah hal yang esensial dilakukan dalam analisis riset yang bertujuan untuk mencari pola alami data (Komorowski et al., 2016). Hal yang dilakukan dalam eksperimen ini untuk melihat suatu pola dengan melihat statistik deskriptif per atribut, melihat korelasi per atribut, melihat distribusi per atribut, dan melihat distribusi IPM jika dipetakan dengan ketentuan dari BPS.

##### 2.1.4.1 Statistik Deskriptif

Tabel statistik deskriptif memberikan gambaran umum mengenai distribusi dan karakteristik dasar dari masing-masing variabel pembangunan manusia. Statistik ini meliputi jumlah data (*count*), nilai rata-rata (*mean*), standar deviasi (*std*), nilai minimum (*min*), maksimum (*max*), serta kuartil 25%, 50% (*median*), dan 75%. Data tersebut disajikan pada Tabel 2.1.

Tabel 2.1 Statistik Deskriptif

	UHH	HLS	RLS	PPP	IPM
count	6369	6369	6369	6369	6369
Mean	69.448	12.869	8.301	10334.818	69.340
std	3.453	1.350	1.652	2711.682	6.765
min	53.600	2.160	0.630	3607.000	25.380
25%	67.300	12.260	7.340	8515.000	65.900
50%	69.800	12.850	8.190	10191.000	69.360
75%	71.670	13.556	9.300	11818.000	73.040
max	78.260	17.940	13.100	25573.000	88.770

Pertama dari UHH, menunjukkan ada wilayah yang mempunyai tingkat kesehatan yang sangat rendah sekali, dengan ditunjukkan nilai minimumnya. Hal ini terjadi ketimpangan kesehatan antar daerah, jika dilihat nilai maksimumnya. Hal ini menunjukkan bahwa perlu dibangunnya kesehatan di wilayah yang mempunyai umur harapan hidup yang rendah.

Kemudian dari HLS, menunjukkan bahwa adanya ketimpangan juga dalam akses pendidikan yang besar, dapat dilihat dari nilai maksimum dan minimumnya. Lalu dari RLS, menunjukkan bahwa ada wilayah yang manusianya sebentar bersekolah. Hal ini menunjukkan bahwa akses pendidikan itu terlalu ketimpangan dan mengindikasikan bahwa wilayah tersebut perlu dibangun aksesnya, agar manusianya dapat pendidikan yang layak.

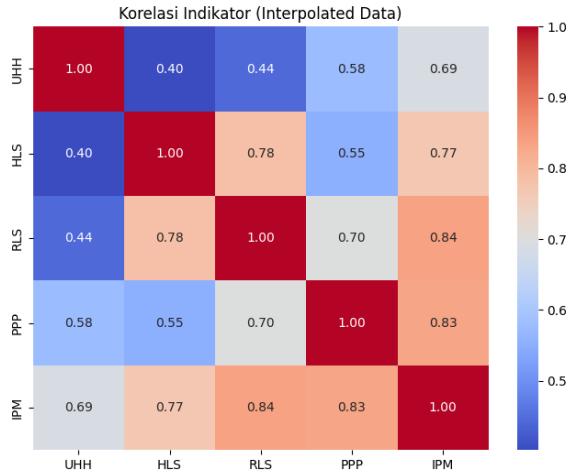
Setelah itu, dilihat dari PPP yang mencatat pengeluaran orang per tahun dalam bentuk ribuan, menunjukkan dari standar deviasinya hanya 2,711,682.- bahwa wilayah-wilayah tersebut mengalami disparitas ekonomi yang tinggi. Selain itu, nilai minimum dan maksimum, menunjukkan bahwa rata-rata pengeluaran orang per tahun di berbagai wilayah mengalami ketimpangan, karena dalam setahun ada daerah yang hanya mengeluarkan 8,515,000 saja. Hal ini melihat bahwa kesejahteraan antarwilayah itu tidak sejahtera.

Terakhir, dari IPM, yang dimana menunjukkan adanya ketimpangan luar biasa. Ada wilayah yang mempunyai indeks pembangunan manusia yang terlalu rendah. Nilai maksimum dan minimum menunjukkan bahwa ketimpangan yang luar biasa. Hal ini berdampak pada sumber daya manusianya, yang mana manusia di wilayah tersebut tidak dapat membangun wilayahnya dan dirinya sendiri.

Secara keseluruhan, menggambarkan bahwa seluruhnya saling berkaitan dalam menggambarkan kesenjangan dan kualitas manusia di Indonesia. Informasi ini menunjukkan juga bahwa pembangunan di Indonesia itu tidak merata dari berbagai sisi yang telah ditetapkan. Informasi ini juga menjadi awal dari pencarian pola natural dari pembangunan manusia.

#### **2.1.4.2 Matriks Korelasi**

Gambar 2.1 menunjukkan bahwa semua saling berkorelasi kuat secara positif. IPM terhadap indikator kesehatan, pengetahuan, dan standar hidup layak menunjukkan bahwa saling terhubung. Hal ini menggambarkan bahwa semakin diperbaiki wilayah-wilayah yang mengalami ketimpangan, maka IPM-nya akan semakin bagus.



Gambar 2.1 Matriks Korelasi

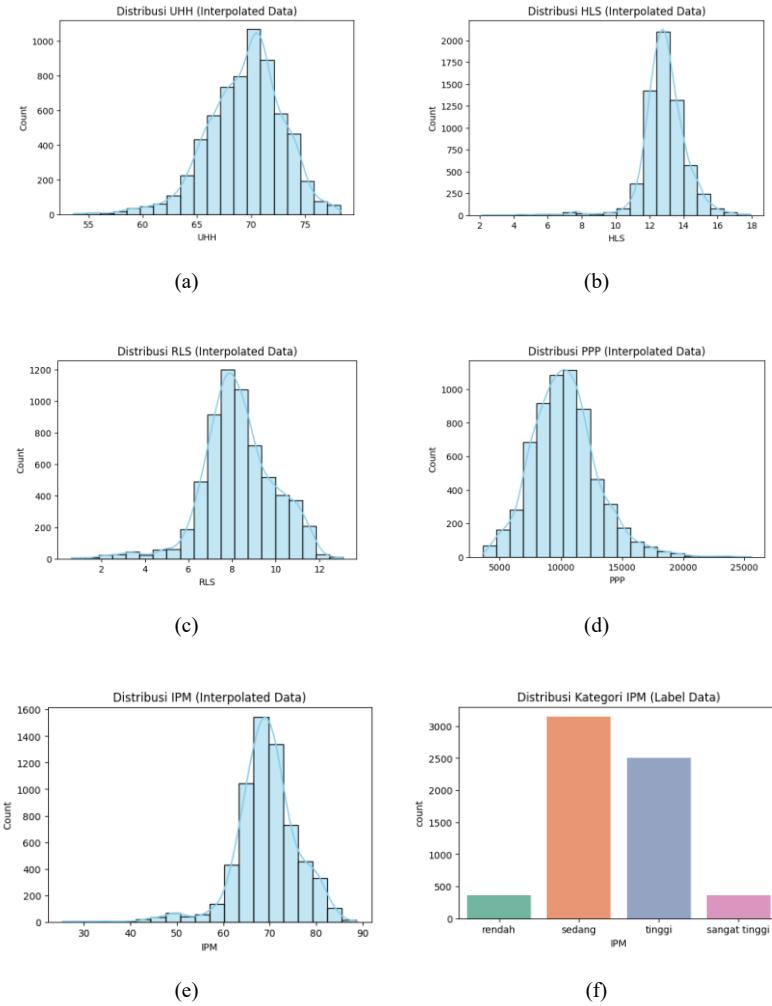
IPM dengan variabel indikator pendidikan bahwa adanya korelasi positif yang kuat. Hal ini menunjukkan bahwa semakin tinggi harapan dan rata-rata lama sekolah, maka semakin tinggi pula indeks pembangunan manusianya. Temuan ini sejalan dengan konsep pembangunan manusia yang menekankan pentingnya pendidikan sebagai komponen utama pembentuk IPM (Sen, 1999).

Indikator kesehatan dengan IPM juga memiliki korelasi positif yang kuat. Hal ini menunjukkan bahwa daerah yang memiliki UHH yang tinggi, cenderung kulitas manusianya juga ikut membaik. Inilah yang mencerminkan kontribusi signifikan dari aspek kesehatan

Indikator kesejahteraan dengan IPM juga menunjukkan korelasi positif yang kuat. Hal ini menunjukkan bahwa wilayah dengan pengeluaran per kapita lebih tinggi umumnya menunjukkan akses lebih besar terhadap layanan dasar manusia. Hal ini ditambah dengan hubungan indikator pendidikan dan kesejahteraan. yang mana menunjukkan bahwa kondisi ekonomi suatu wilayah turut memengaruhi pencapaian pendidikan.

Selain analisis pola natural, didapatkan juga bahwa variabel-variabel independentnya tidak mengalami multikolinearitas yang ekstrem. Hal ini ditunjukkan bahwa korelasinya tidak mendekati 1. Selain itu, ini menunjukkan bahwa seluruh variabelnya adalah relevan digunakan dan tidak mengganggu proses pemodelan berbasis machine learning.

#### 2.1.4.3 Distribusi per Atribut



Gambar 2.2 Distribusi Atribut

Distribusi setiap atribut dianalisis melalui histogram yang dilengkapi *kernel density estimation* (KDE). Visualisasi ini digunakan untuk memahami pola sebaran data, mendeteksi potensi *outlier*, serta melihat apakah data cenderung mengikuti distribusi tertentu. Gambar 2.2 menampilkan hasil distribusi dari seluruh variabel numerik serta kategori IPM setelah pelabelan.

Gambar 2.2(a) menunjukkan bahwa distribusi UHH cenderung mengikuti pola mendekati normal dengan nilai rataan berada pada kisaran 69–70 tahun. Sebagian besar wilayah berada pada rentang 65–72 tahun, sementara nilai ekstrem berada di bawah 60 atau di atas 75 tahun relatif jarang. Pola ini menunjukkan bahwa aspek kesehatan penduduk relatif stabil antarwilayah, dengan variasi moderat yang

mencerminkan perbedaan akses fasilitas kesehatan dan standar hidup di berbagai daerah.

Distribusi pada Gambar 2.2(b) menunjukkan pola yang lebih terpusat dan sempit dibandingkan variabel lain. Mayoritas nilai HLS berada pada rentang 12–14 tahun, dengan rataan sekitar 12,8 tahun. Rentang nilai yang sempit dan standar deviasi kecil mengindikasikan bahwa pembangunan pendidikan, khususnya durasi pendidikan yang diharapkan, relatif seragam antarwilayah. Namun, terdapat beberapa nilai ekstrem di bawah 8 tahun yang menunjukkan adanya daerah dengan akses pendidikan sangat terbatas.

Gambar 2.2(c) menampilkan distribusi RLS yang juga menyerupai distribusi normal, tetapi dengan rentang yang lebih lebar daripada HLS. Mayoritas wilayah memiliki RLS antara 7–10 tahun, menggambarkan bahwa sebagian besar penduduk telah menyelesaikan pendidikan dasar hingga menengah pertama. Kemunculan nilai rendah (<3 tahun) meskipun sedikit, menunjukkan adanya wilayah yang masih menghadapi tantangan dalam distribusi akses pendidikan formal.

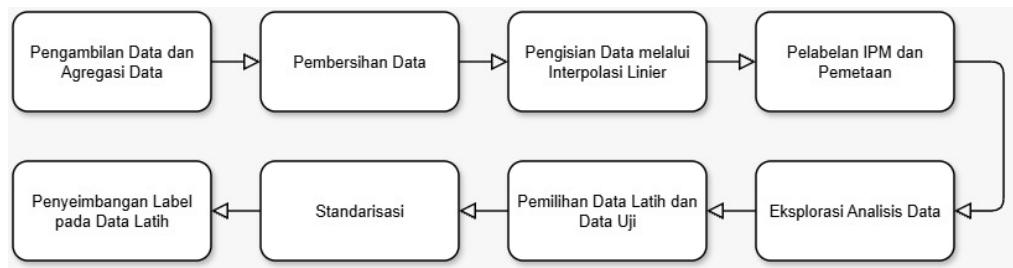
Pada Gambar 2.2(d), distribusi PPP menunjukkan pola *right-skewed* atau miring ke kanan. Hal ini berarti sebagian besar wilayah memiliki pengeluaran per kapita yang berkisar pada nilai menengah (sekitar 8–12 juta rupiah per orang per tahun dalam satuan ribu rupiah), tetapi terdapat beberapa wilayah dengan nilai sangat tinggi (hingga 25 juta rupiah). Skewness ini merupakan pola umum dalam data ekonomi karena pertumbuhan ekonomi antarwilayah cenderung tidak merata.

Gambar 2.2(e) memperlihatkan distribusi IPM numerik yang membentuk pola mendekati normal dengan puncak distribusi pada kisaran 68–72. Distribusi yang cukup simetris ini menggambarkan bahwa sebagian besar wilayah berada pada tingkat pembangunan manusia “sedang” hingga “tinggi”. Namun ekstrem nilai rendah (<50) dan tinggi (>85) menunjukkan adanya ketimpangan pembangunan yang signifikan antarwilayah di Indonesia.

Gambar 2.2(f) menunjukkan ketidakseimbangan kelas yang cukup jelas. Ketidakseimbangan ini penting dicermati karena dapat menyebabkan model klasifikasi bias terhadap kelas mayoritas. Oleh

karena itu, pada tahap pemodelan diterapkan teknik SMOTE dan tanpa *upsampling* untuk melihat evaluasi antar kelas dan model, terutama pada kelas minoritas.

## 2.2 Pra-pemrosesan Data



Gambar 2.3 Alur Praproses Data

Sebelum dilakukan praproses, dilakukan pengambilan data sekaligus agregasi data melalui website BPS. Pengambilan dan agregasi data dilakukan untuk mengambil data IPM dan data-data yang mempengaruhi IPM. Setelah itu data diunduh ke dalam bentuk “.xlsx”. Alasannya, kalau diunduh ke CSV maka datanya berantakan sekali. Setelah itu, barulah masuk ke dalam pra-pemrosesan data, agar data dapat dipakai untuk melatih dan menguji mesin belajar.

Kemudian, masuk ke dalam pembersihan data. Pertama kali mengkonversikan karakter “-” menjadi nilai hilang. Setelah itu, barulah dilakukan interpolasi linier. Interpolasi linier dipakai untuk mengisi data yang hilang. Sebab, ada daerah yang disebabkan oleh pemekaran wilayah. Berdasarkan UU No. 16 Tahun 2022 dan UU No. 29 Tahun 2022, pemekaran wilayah dilakukan untuk mempercepat pembangunan. Hal inilah yang menyebabkan data tersebut kosong, tidak mungkin daerah tersebut dibuang, sebab itu dilakukan interpolasi linier. Namun, pengisian ini bisa saja menjadi bias, sebab nilainya belum pasti sama dengan BPS kalau data tersebut diperbarui.

Setelah itu, dilakukan pelabelan IPM dan pemetaan. Hal ini berguna, agar model dapat dengan mudah melakukan klasifikasi. Pelabelan IPM dilakukan sesuai dengan *booklet* IPM metode-baru, dimana terdapat nilai untuk diklasifikasi. Nilai-nilai tersebut diklasifikasikan menjadi rendah, sedang, tinggi, dan sangat tinggi. Setelah itu, dipetakan menjadi 1-4, agar model dapat mengerti dalam klasifikasikan wilayah-wilayah. Sebab, algoritma pembelajaran mesin tidak dapat memproses data kategorikal secara langsung tanpa representasi numerik (Pedregosa et al., 2024).

Selanjutnya, dilakukan eksplorasi analisis data. Eksplorasi analisis data, digunakan untuk melihat korelasi antar variabel, distribusi per variabel, dan distribusi

kategori IPM. Hal ini bertujuan untuk melihat variabel-variabel tersebut cocok dan dapat berjalan pada model pembelajaran mesin, serta mencari pola natural.

Kemudian, dilakukan pemilihan data uji dan tes. Hal ini digunakan untuk pelatihan dan validasi model. Ini bertujuan untuk melihat evaluasi setiap model dalam pelatihannya. Data pelatihan diambil dari tahun 2014-2022, sedangkan data pengujian diambil dari tahun 2023-2024.

Lalu, data pelatihan distandarisasikan. Standarisasi dilakukan pada seluruh fitur numerik menggunakan StandardScaler, yang merupakan metode transformasi data dengan cara mengubah setiap fitur menjadi memiliki rata-rata (mean) sebesar 0 dan standar deviasi sebesar 1. Transformasi ini penting karena banyak algoritma pembelajaran mesin bekerja secara sensitif terhadap skala fitur; model seperti SVM, Logistic Regression, dan Multi-Layer Perceptron mengandalkan perhitungan jarak atau optimisasi gradien yang dapat terganggu apabila fitur memiliki rentang nilai yang sangat berbeda (Géron, 2022).

Dengan melakukan standarisasi, setiap fitur ditempatkan pada skala yang sebanding sehingga model dapat melakukan proses pelatihan secara lebih stabil dan efisien. Selain itu, standarisasi juga membantu mempercepat proses konvergensi pada algoritma berbasis gradien, serta mencegah dominasi fitur tertentu yang memiliki nilai lebih besar dibanding fitur lainnya (Aggarwal, 2023). Proses ini hanya dilakukan pada data latih untuk mendapatkan parameter mean dan standar deviasi, kemudian parameter tersebut diterapkan kembali pada data uji, sesuai praktik yang dianjurkan agar tidak terjadi kebocoran informasi dari data uji ke tahap pelatihan (Han et al., 2021).

Terakhir, dilakukan penyeimbangan label pada data latih. Hal ini bertujuan agar model dapat mempelajari kelas minor. Label diseimbangkan dengan SMOTE, agar tidak banyak yang hilang. Sebab data mayor itu terlalu banyak dibandingkan data minor sehingga dilakukan peningkatan sampel data label pada data pelatihan.

### 2.3 Penentuan Model

Pemilihan model dalam eksperimen ini berdasarkan pada kemampuan masing-masing algoritma dalam menangani masalah klasifikasi multikelas secara efektif. Model yang digunakan, yaitu Decision Tree, Random Forest, Gradient Boosting, AdaBoost, SVM, MLP, dan Logistic Regression. Model-model ini merupakan algoritma yang secara luas direkomendasikan dalam berbagai literatur, sebab performanya yang stabil pada data tabular. Algoritma berbasis pohon seperti Decision Tree, Random Forest, dan Gradient

Boosting mampu menangani multikelas secara langsung melalui pemisahan berjenjang pada setiap node. AdaBoost juga dapat digunakan untuk kasus multikelas melalui penyesuaian bobot kesalahan secara iteratif (Zhu et al., 2020).

Sementara itu, SVM dapat melakukan klasifikasi multikelas menggunakan pendekatan one-vs-rest atau one-vs-one, terutama ketika dikombinasikan dengan kernel RBF yang mampu memetakan data non-linier (Han et al., 2021). Logistic Regression digunakan dalam bentuk multinomial yang memanfaatkan fungsi softmax untuk memprediksi tiga kelas atau lebih secara langsung, dan pendekatan ini telah menjadi metode standar dalam klasifikasi multikelas pada data terstruktur (Géron, 2022). Selain itu, Multi-Layer Perceptron (MLP) mendukung multiclass melalui penggunaan softmax di lapisan output, sehingga mampu mempelajari hubungan non-linier antarf fitur (Aggarwal, 2023). Dengan demikian, kombinasi ketujuh model ini memberikan variasi pendekatan yang komprehensif, dari metode interpretable hingga model berbasis jaringan saraf.

## BAB III

### PENGEMBANGAN MODEL DAN EVALUASI

#### 3.1 Arsitektur Model

Arsitektur Decision Tree berbentuk struktur pohon yang terdiri dari root node, internal nodes, dan leaf nodes, di mana setiap node melakukan pemilihan fitur terbaik menggunakan metrik seperti Gini Impurity atau Entropy untuk memecah data hingga mencapai kelas tertentu. Pemodelan berbasis pohon ini bekerja secara hierarkis, sehingga mampu menangani klasifikasi multikelas secara langsung tanpa modifikasi tambahan. Struktur pohon yang interpretable ini menjadikannya salah satu metode dasar dalam klasifikasi data tabular (Han et al., 2021).

Arsitektur Random Forest terdiri dari kumpulan (ensemble) banyak Decision Tree yang dilatih secara paralel menggunakan teknik *bootstrap sampling* serta pemilihan fitur secara acak pada setiap node. Setiap pohon menghasilkan prediksi kelas, dan hasil akhir diperoleh melalui proses *majority voting*. Pendekatan ensemble ini mengurangi risiko overfitting dan secara alami mendukung klasifikasi multikelas karena setiap pohon secara independen menangani beberapa kelas (Géron, 2022).

Gradient Boosting memiliki arsitektur sekuensial, di mana setiap pohon keputusan dibangun untuk memperbaiki kesalahan model sebelumnya dengan memanfaatkan gradien dari fungsi loss. Model ini membentuk *additive model* yang secara bertahap meningkatkan kualitas prediksi. Untuk kasus multikelas, gradient boosting mengoptimalkan fungsi loss secara terpisah bagi tiap kelas dan menggabungkannya dalam proses boosting, sehingga mampu menyesuaikan pola yang kompleks pada data (Géron, 2022).

Arsitektur AdaBoost terdiri dari sejumlah *weak learners*, biasanya Decision Tree stump, yang dilatih secara berurutan dengan bobot yang disesuaikan berdasarkan tingkat kesalahan pada iterasi sebelumnya. Untuk klasifikasi multikelas, AdaBoost menggunakan generalisasi seperti SAMME atau SAMME.R, di mana model memberikan kontribusi berbeda pada setiap weak learner sesuai akurasinya. Struktur adaptif ini memungkinkan AdaBoost memperkuat area yang sulit diprediksi sambil mempertahankan interpretabilitas model (Zhu et al., 2020).

Arsitektur SVM didasarkan pada pencarian hyperplane yang memaksimalkan margin antar kelas. Karena secara default SVM adalah binary classifier, maka untuk kasus multikelas digunakan pendekatan *One-Vs-Rest* atau *One-Vs-One*. Pada data non-

linier, kernel seperti RBF digunakan untuk memetakan data ke ruang berdimensi lebih tinggi sehingga dapat dipisahkan secara linear oleh hyperplane tersebut. Pendekatan ini terbukti efektif untuk klasifikasi multikelas pada berbagai domain data (Han et al., 2021).

Arsitektur MLP terdiri dari sebuah lapisan input, beberapa lapisan tersembunyi, dan lapisan output dengan fungsi aktivasi softmax untuk mendukung klasifikasi multikelas. Informasi bergerak secara feed-forward dari input menuju hidden layer yang berisi neuron dengan aktivasi non-linier seperti ReLU, lalu menuju output yang memetakan probabilitas dari masing-masing kelas. Melalui proses *backpropagation*, model menyesuaikan bobot untuk mempelajari pola kompleks dalam data tabular (Aggarwal, 2023).

Arsitektur Multinomial Logistic Regression menggunakan fungsi aktivasi linear  $z = W_x + b$  dan fungsi aktivasi softmax pada lapisan output untuk menghasilkan probabilitas pada lebih dari dua kelas. Setiap kelas memiliki parameter bobot tersendiri, sehingga model dapat melakukan pemisahan multikelas secara langsung. Metode ini dikenal stabil, interpretatif, dan secara luas digunakan pada klasifikasi multikelas berbasis data terstruktur (Géron, 2022).

### 3.2 Pelatihan Model

Tahap pelatihan model dilakukan setelah seluruh data melalui proses pembersihan, interpolasi, pelabelan, dan konversi label menjadi bentuk numerik. Dataset kemudian dibagi menjadi dua bagian, yaitu data latih dan data uji, dengan data tahun 2014–2022 sebagai data latih dan tahun 2023–2024 sebagai data uji. Pembagian ini mengikuti praktik umum dalam supervised learning, di mana train–test split digunakan untuk mengevaluasi kemampuan generalisasi model secara langsung tanpa teknik cross-validation (Géron, 2022). Empat fitur utama yang digunakan dalam pelatihan adalah UHH, HLS, RLS, dan PPP, sedangkan variabel target adalah IPM yang telah ditransformasikan menjadi empat kelas ordinal.

Sebelum pelatihan dilakukan, seluruh fitur numerik distandardisasi menggunakan StandardScaler agar nilai tiap fitur berada pada skala yang seragam. Standardisasi diperlukan terutama untuk model seperti SVM, Logistic Regression, dan MLP yang sensitif terhadap perbedaan skala fitur dalam proses optimasinya (Han et al., 2021). Selain itu, distribusi kelas pada data latih menunjukkan ketidakseimbangan, sehingga dilakukan penanganan imbalance menggunakan SMOTE, yaitu teknik oversampling sintetis yang menghasilkan sampel baru berdasarkan interpolasi antar sampel minoritas.

Pendekatan ini sering digunakan dalam klasifikasi multikelas untuk mengurangi bias model terhadap kelas mayoritas (Aggarwal, 2023).

Setelah proses SMOTE, bobot kelas dihitung menggunakan `class_weight` untuk menyeimbangkan kontribusi tiap kelas selama proses pelatihan. Tahap berikutnya adalah pelatihan model Machine Learning. Terdapat tujuh model yang digunakan, yaitu Decision Tree, Random Forest, Gradient Boosting, AdaBoost, Support Vector Machine, Multi-Layer Perceptron (MLP), dan Multinomial Logistic Regression. Setiap model dilatih menggunakan parameter yang disesuaikan dengan sifat datanya.

Decision Tree dilatih melalui pembentukan struktur pohon hierarkis berdasarkan pemilihan fitur terbaik, sementara Random Forest dan Gradient Boosting memanfaatkan teknik ensemble untuk meningkatkan kinerja dengan menggabungkan hasil dari beberapa pohon keputusan (Géron, 2022). AdaBoost dilatih secara bertingkat menggunakan pendekatan SAMME sehingga setiap weak learner berkontribusi sesuai tingkat kesalahannya (Zhu et al., 2020). SVM menggunakan kernel RBF dan pendekatan One-Vs-Rest untuk menangani empat kelas target, sesuai dengan rekomendasi metode SVM multikelas modern (Han et al., 2021).

Model MLP (ANN) dilatih menggunakan arsitektur dua hidden layer dengan 64 dan 32 neuron, aktivasi ReLU, serta fungsi softmax pada output. Proses pembaruan bobot dilakukan melalui backpropagation dengan optimisasi gradien, sehingga jaringan mampu menangkap relasi nonlinier antar fitur (Aggarwal, 2023). Logistic Regression dalam bentuk multinomial menggunakan fungsi softmax untuk memetakan probabilitas tiap kelas dan dioptimasi menggunakan cross-entropy loss, sehingga efektif untuk kasus klasifikasi multikelas berbasis data tabular (Géron, 2022).

Selama pelatihan, setiap model dievaluasi menggunakan data uji melalui metrik accuracy, weighted F1-score, serta log-loss bagi model yang mendukung probabilitas. Selain itu, dilakukan juga analisis berupa Matriks Evaluasi Klasifikasi dan visualisasi prediksi benar-salah per kelas untuk memahami pola kesalahan model. Pendekatan evaluasi langsung terhadap test set ini memberikan gambaran umum mengenai kemampuan generalisasi model terhadap data terbaru tanpa melibatkan proses cross-validation.

### 3.3 *Hyperparameter Tuning atau Fine Tuning*

Pada penelitian ini tidak dilakukan proses *hyperparameter tuning* maupun *fine-tuning* secara khusus terhadap model pembelajaran mesin yang digunakan. Seluruh

algoritma dilatih menggunakan konfigurasi hiperparameter dasar (*default parameters*) atau parameter yang direkomendasikan oleh pustaka Scikit-Learn, sehingga fokus utama penelitian berada pada pemahaman hubungan variabel input dan performa model secara umum. Tidak diterapkannya tuning disebabkan oleh dua alasan utama. Pertama, penelitian ini menekankan evaluasi perbandingan model dan bukan optimasi model secara mendalam; sehingga penggunaan parameter standar sudah memadai untuk menilai kemampuan dasar masing-masing algoritma (Géron, 2022). Kedua, proses tuning seperti *grid search* atau *random search* memerlukan waktu komputasi yang lebih besar, terutama pada model ensemble dan neural network, sedangkan penelitian ini mengutamakan efisiensi komputasi dengan tetap menjaga keandalan hasil (Han et al., 2021). Meskipun demikian, konfigurasi awal yang digunakan tetap mengikuti prinsip-prinsip dasar pemilihan hyperparameter yang umum direkomendasikan oleh literatur modern Machine Learning (Aggarwal, 2023), sehingga model yang dihasilkan tetap relevan dan representatif.

### 3.4 Hasil Evaluasi Model

Evaluasi model dilakukan untuk menilai kemampuan setiap algoritma dalam melakukan klasifikasi tingkat Indeks Pembangunan Manusia (IPM) pada data uji tahun 2023 dan 2024. Proses evaluasi dilakukan setelah seluruh model dilatih menggunakan data periode 2014–2022 yang telah melalui pra-pemrosesan berupa interpolasi, standarisasi fitur, dan penanganan ketidakseimbangan kelas menggunakan oversampling SMOTE. Evaluasi dilakukan secara langsung menggunakan hold-out test set tanpa menerapkan teknik cross-validation, sehingga performa model mencerminkan kemampuan generalisasi terhadap data baru yang tidak dilibatkan dalam proses pelatihan (Géron, 2022).

Beberapa metrik digunakan untuk mengukur kinerja model. Akurasi digunakan untuk melihat proporsi prediksi yang tepat dari seluruh sampel data uji. Namun karena klasifikasi IPM memiliki distribusi kelas yang tidak seimbang, metrik weighted F1-score dipilih sebagai ukuran tambahan untuk menilai keseimbangan antara precision dan recall pada seluruh kelas tanpa mengabaikan kelas minoritas (Aggarwal, 2023). Selain itu, untuk model yang mendukung prediksi probabilitas, seperti Random Forest, Logistic Regression, Gradient Boosting, AdaBoost, MLP, dan SVM dengan probabilitas, dihitung juga nilai log-loss. Metrik ini memberikan penilaian yang lebih sensitif terhadap probabilitas prediksi dan kesalahan tingkat ketidakpastian model, sehingga membantu

memahami seberapa baik model memprediksi distribusi kelas secara probabilistik (Han et al., 2021).

Selain metrik numerik, evaluasi juga dilakukan menggunakan Matriks Evaluasi Klasifikasi untuk melihat pola kesalahan prediksi tiap model terhadap empat kelas IPM (rendah, sedang, tinggi, sangat tinggi). Matriks Evaluasi Klasifikasi memudahkan identifikasi kelas mana yang sering tertukar dan seberapa baik model membedakan tiap kategori IPM. Hasil evaluasi model, termasuk tabel metrik performa serta visualisasi Matriks Evaluasi Klasifikasi, akan dibahas secara rinci pada bagian 3.5 Analisis. Dengan pendekatan evaluasi ini, penelitian memastikan bahwa performa model yang dihasilkan dapat diinterpretasikan secara komprehensif dan sesuai dengan karakteristik data pembangunan manusia.

### 3.5 Analisis

#### 3.5.1 Decision Tree

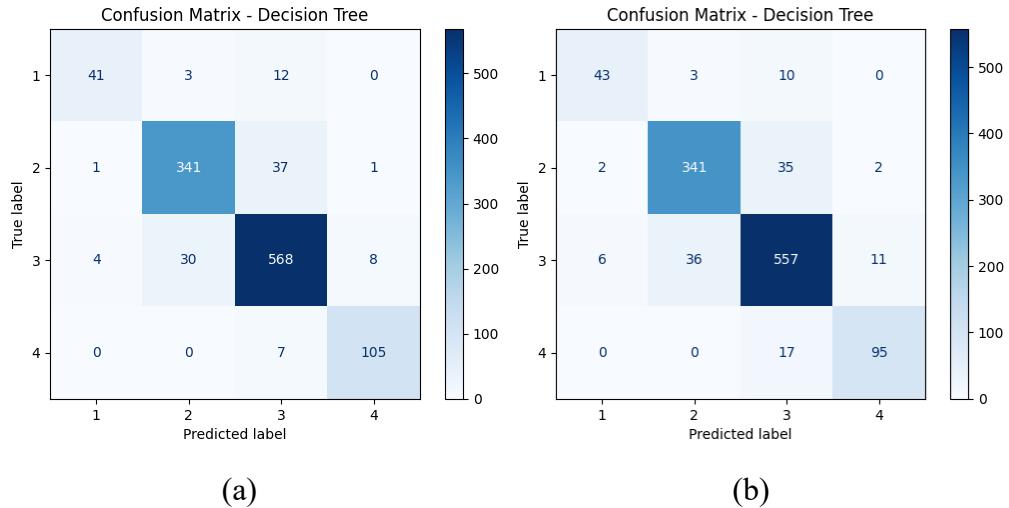
Tabel 3.1 Laporan Klasifikasi Decision Tree: SMOTE vs Non-SMOTE

Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.891	0.732	0.804	0.843	0.768	0.804
2	0.912	0.897	0.905	0.897	0.897	0.897
3	0.910	0.931	0.921	0.900	0.913	0.906
4	0.921	0.938	0.929	0.880	0.848	0.864

Berdasarkan Tabel 3.1, terlihat bahwa penerapan SMOTE memberikan peningkatan performa yang cukup signifikan pada model Decision Tree dibandingkan model tanpa SMOTE. Peningkatan ini terutama tampak pada kelas-kelas minoritas, sebagaimana tercermin dari nilai presisi dan kemampuan model dalam menangkap sampel kelas yang jumlahnya lebih sedikit. Meski demikian, performa pada kelas “Rendah” (kelas 1) justru sedikit menurun setelah SMOTE diterapkan, yang mengindikasikan bahwa penyeimbangan data sintetis belum sepenuhnya mampu memperbaiki karakteristik prediksi pada kelas tersebut.

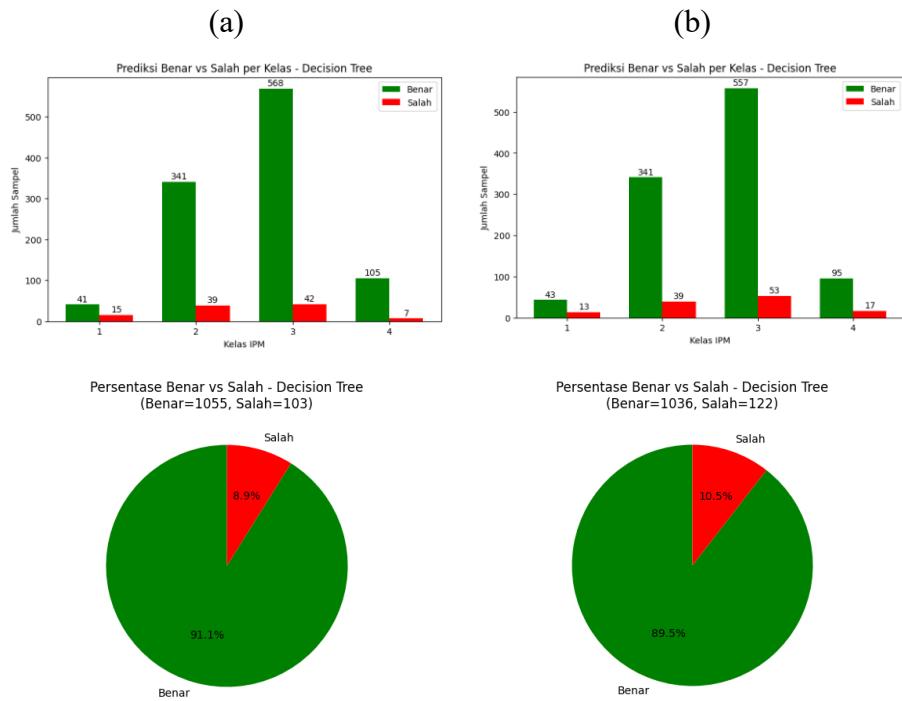
F1-score memberikan gambaran tambahan mengenai kekuatan dan kelemahan performa per kelas, dan hasilnya menunjukkan bahwa secara umum SMOTE memperkuat kemampuan model dalam mengklasifikasikan setiap kategori IPM dibandingkan dengan model tanpa SMOTE. Meskipun demikian,

penilaian berbasis metrik numerik saja tidak cukup untuk menjelaskan pola kesalahan prediksi yang terjadi. Oleh karena itu, analisis selanjutnya akan melihat matriks evaluasi klasifikasi untuk memperoleh gambaran yang lebih detail mengenai distribusi kesalahan yang muncul pada tiap kelas.



Gambar 3.1 Matriks Evaluasi Decision Tree: SMOTE (a) vs Non-SMOTE (b)

Berdasarkan Gambar 3.1, penggunaan SMOTE memberikan dampak positif terhadap performa model Decision Tree karena membuat model lebih mampu memprediksi kelas secara tepat, khususnya pada kelas-kelas minoritas yang sebelumnya kurang terwakili, sehingga distribusi pembelajaran menjadi lebih seimbang dan model tidak hanya berfokus pada kelas dominan, tetapi juga mampu mengenali kelas Rendah, Sedang, Tinggi, dan Sangat Tinggi secara lebih stabil. Hal ini diperkuat oleh visualisasi pada Gambar 3.2 yang menunjukkan peningkatan jumlah prediksi benar pada tiap kategori, selaras dengan konsep SMOTE yang memang dirancang untuk meningkatkan sensitivitas model terhadap kelas minor sehingga pola prediksi menjadi lebih merata, namun demikian performa awal yang terlihat baik ini belum cukup untuk menilai kualitas model secara menyeluruh tanpa validasi tambahan berbasis loss, karena evaluasi tersebut penting untuk mendeteksi potensi overfitting, yaitu ketika akurasi tinggi tetapi loss juga tinggi, maupun underfitting, ketika akurasi rendah dan loss tetap tinggi, sehingga penilaian model tidak dapat berhenti pada evaluasi per kelas saja, melainkan perlu diintegrasikan dengan metrik lain seperti akurasi keseluruhan, loss, serta keseimbangan performa antar kelas agar diperoleh pemahaman yang lebih komprehensif mengenai kemampuan generalisasi dan kualitas model Decision Tree secara keseluruhan.



Gambar 3.2 Visualisasi Prediksi Decision Tree: SMOTE (a) vs Non-SMOTE (b)

Dari gambar 3.2 menunjukkan bahwa model dengan SMOTE memang terbaik. Hal ini bisa dilihat hanya terdapat kesalahan kecil sebesar 8.9% daripada tanpa SMOTE sebesar 10.5%. Walaupun kelas paling minor masih belum menunjukkan perbaikan, malah makin parah, tetapi kelas lainnya menunjukkan model lebih bagus dengan SMOTE, tetapi perlu validasi loss dan akurasi totalnya.

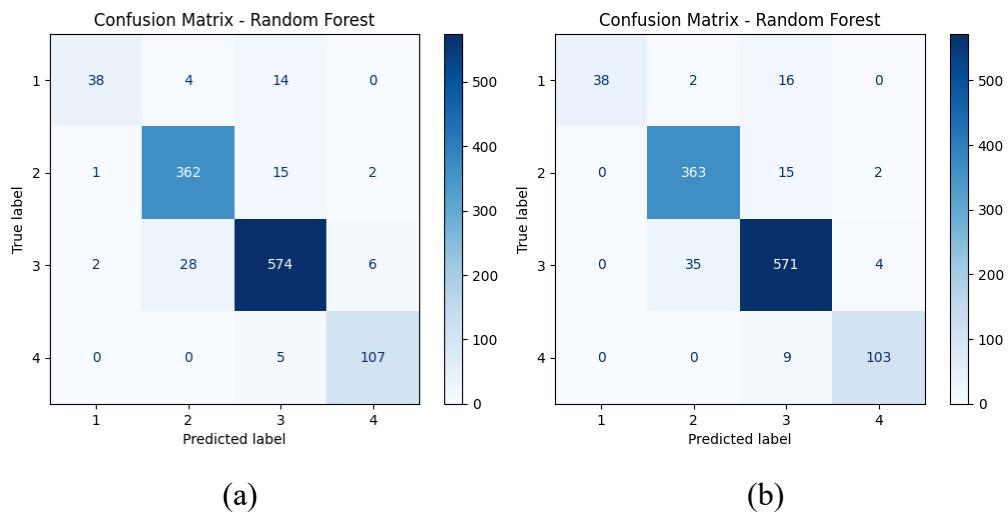
### 3.5.2 Random Forest

Tabel 3.2 Laporan Klasifikasi Random Forest: SMOTE vs Non-SMOTE

Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.927	0.679	0.784	1.000	0.679	0.809
2	0.919	0.953	0.935	0.907	0.955	0.931
3	0.944	0.941	0.943	0.935	0.936	0.935
4	0.955	0.955	0.943	0.945	0.920	0.932

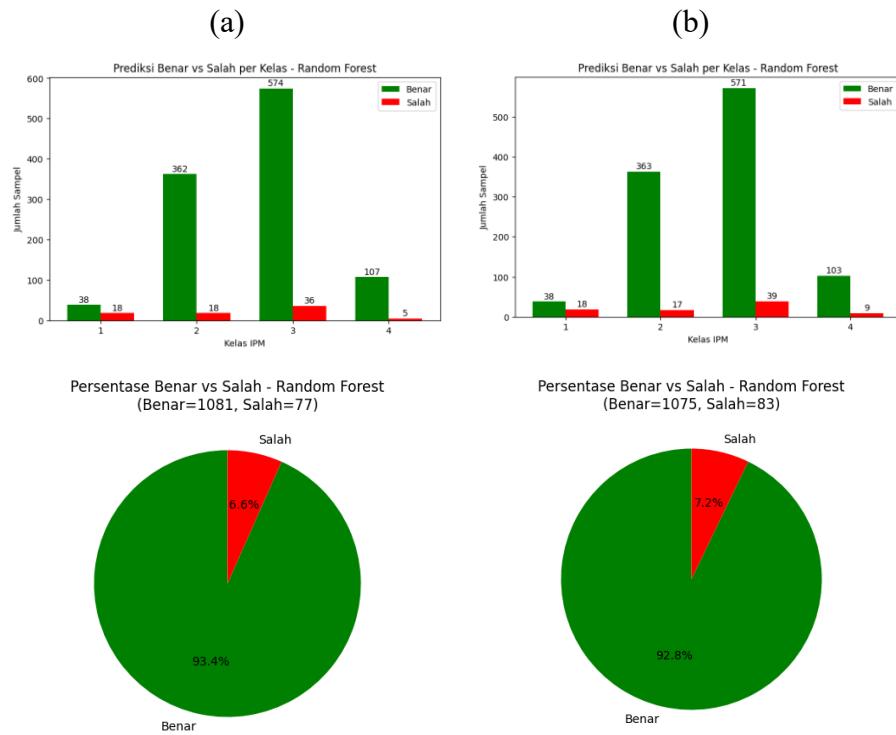
Berdasarkan Tabel 3.2, penerapan SMOTE pada model Random Forest menunjukkan dampak yang bervariasi terhadap performa per kelas, di mana pada kelas 1 terjadi penurunan nilai presisi sementara nilai recall tetap sama, yang mengindikasikan bahwa model menjadi lebih sering melakukan kesalahan prediksi pada kelas tersebut meskipun kemampuannya dalam menangkap seluruh sampel kelas

1 tidak mengalami peningkatan, sedangkan pada kelas-kelas lainnya terlihat adanya peningkatan presisi dan recall yang cukup jelas, sehingga menunjukkan bahwa SMOTE membantu Random Forest dalam mengenali pola data baik pada kelas dengan jumlah sampel lebih besar maupun pada beberapa kelas minoritas, namun perbedaan performa antarkelas ini menegaskan perlunya validasi lanjutan untuk memastikan kelayakan model dalam konteks prediksi IPM, sehingga analisis dilanjutkan melalui matriks evaluasi klasifikasi pada Gambar 3.3 yang memberikan gambaran lebih rinci mengenai jenis serta pola kesalahan prediksi yang dilakukan model pada setiap kelas.



Gambar 3.3 Matriks Evaluasi Random Forest: SMOTE (a) vs Non-SMOTE (b)

Berdasarkan Gambar 3.3, dapat dilihat bahwa penggunaan SMOTE memberikan peningkatan performa pada model Random Forest, ditandai dengan berkurangnya jumlah kesalahan prediksi dibandingkan model tanpa SMOTE. Meskipun perbaikan tersebut tidak bersifat drastis, SMOTE tetap memberikan kontribusi positif terutama pada kelas 3 dan kelas 4, yang menunjukkan peningkatan jumlah prediksi benar. Perbaikan ini mengindikasikan bahwa Random Forest lebih mampu menangkap pola pada kelas-kelas tersebut ketika distribusi data dibuat lebih seimbang melalui proses oversampling. Untuk memahami perubahan ini secara lebih jelas, visualisasi distribusi prediksi benar dan salah pada Gambar 3.4 memberikan perspektif tambahan mengenai bagaimana SMOTE memengaruhi kinerja model pada tiap kategori IPM.



Gambar 3.4 Visualisasi Prediksi Random Forest: SMOTE (a) vs Non-SMOTE (b)

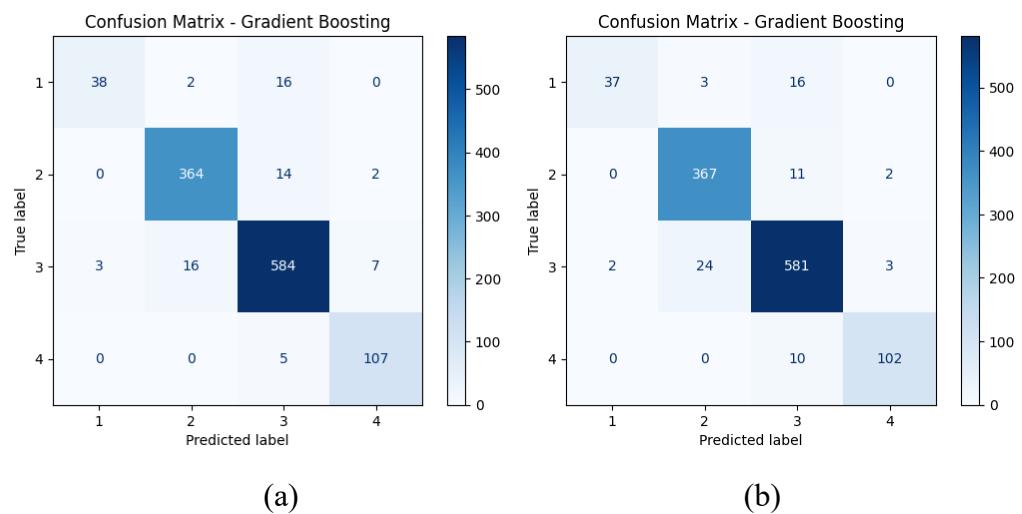
Gambar 3.4 memberikan visualisasi yang lebih cepat dan mudah dipahami dalam mengevaluasi performa model Random Forest. Melalui grafik ini, terlihat bahwa meskipun jumlah kesalahan prediksi relatif kecil, hasil tersebut tetap perlu dikonfirmasi dengan metrik evaluasi lainnya untuk memastikan konsistensi performa model. Penggunaan SMOTE menunjukkan hasil yang lebih baik secara keseluruhan dan selaras dengan temuan pada laporan klasifikasi, yang mengindikasikan bahwa proses penyeimbangan data membantu model mengenali pola antar kelas dengan lebih efektif. Dengan demikian, SMOTE terbukti memberikan kontribusi positif dalam peningkatan kinerja Random Forest.

### 3.5.3 Gradient Boosting

Tabel 3.3 Laporan Klasifikasi Gradient Boosting: SMOTE vs Non-SMOTE

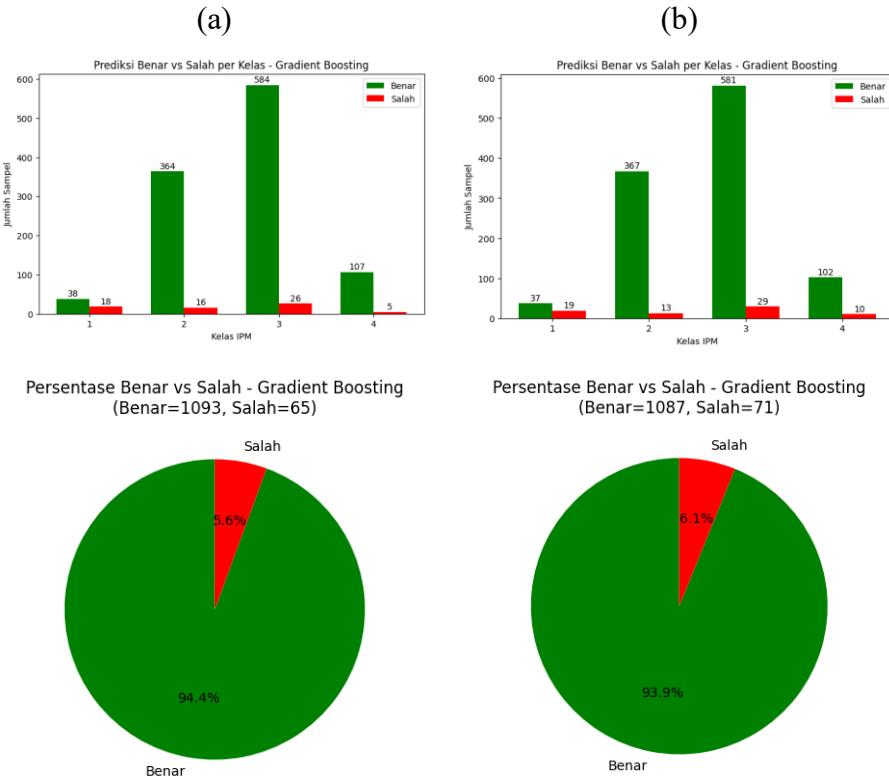
Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.927	0.679	0.784	0.949	0.661	0.779
2	0.953	0.958	0.955	0.931	0.966	0.948
3	0.943	0.957	0.950	0.940	0.952	0.946
4	0.922	0.955	0.939	0.953	0.911	0.932

Hasil evaluasi menunjukkan bahwa penerapan SMOTE pada model Gradient Boosting mampu meningkatkan keseimbangan dan stabilitas performa, terutama dalam mengenali kelas minoritas, meskipun terjadi penurunan presisi pada beberapa kelas tertentu. Secara keseluruhan, peningkatan F1-score mengindikasikan bahwa SMOTE membantu model mempelajari distribusi kelas yang tidak seimbang dengan lebih baik tanpa menunjukkan indikasi overfitting yang signifikan, walaupun terdapat sedikit peningkatan kesalahan prediksi pada kelas “Tinggi” akibat meningkatnya sensitivitas model terhadap variasi antarkelas.



Gambar 3.5 Matriks Evaluasi Gradient Boosting: SMOTE (a) vs Non-SMOTE (b)

Berdasarkan Gambar 3.5, penerapan SMOTE memberikan perbaikan positif pada pola prediksi model Gradient Boosting, di mana pada model tanpa SMOTE kesalahan prediksi banyak terjadi pada kelas 3 yang diprediksi sebagai kelas 2 serta pada kelas 1 yang sebagian diprediksi sebagai kelas 3, sementara setelah SMOTE diterapkan distribusi kesalahan menjadi lebih terkendali dengan peningkatan jumlah prediksi benar pada kelas 2 dari 364 menjadi 367 dan perbaikan pola kesalahan pada kelas 3 meskipun masih terdapat sedikit peningkatan kesalahan tertentu, yang secara keseluruhan menunjukkan bahwa SMOTE membantu model menjadi lebih stabil dalam mengenali pola antar kelas, menangkap variasi data pada kelas minoritas, dan memberikan dampak positif moderat terhadap kualitas prediksi Gradient Boosting, sejalan dengan peningkatan performa yang ditunjukkan pada hasil laporan klasifikasi.



Gambar 3.6 Visualisasi Prediksi Gradient Boosting: SMOTE (a) vs Non-SMOTE (b)

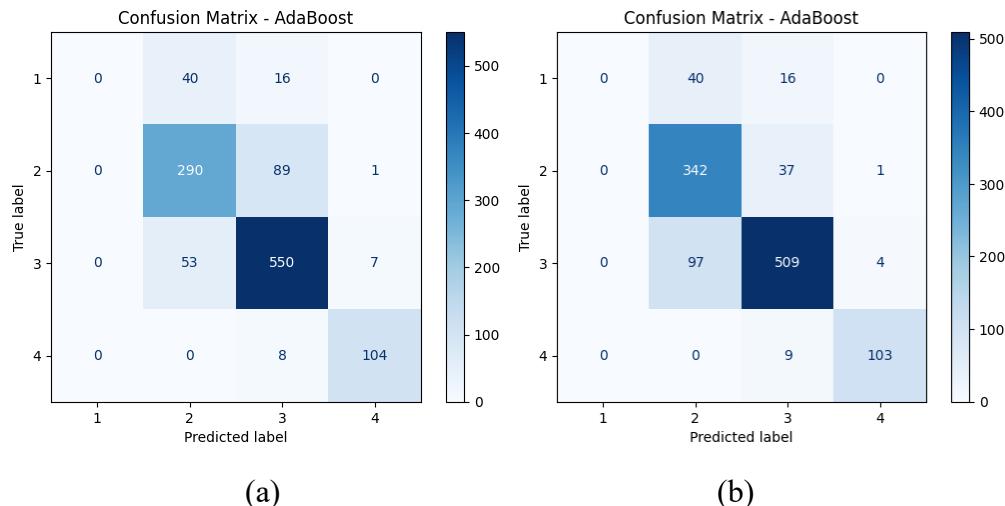
Gambar 3.6 menyajikan visualisasi prediksi benar dan salah pada model Gradient Boosting. Melalui grafik ini, terlihat bahwa model dengan SMOTE memiliki proporsi prediksi benar yang sedikit lebih tinggi dibandingkan model tanpa SMOTE, dengan tingkat kesalahan sebesar 5.6% pada SMOTE dan 6.1% pada model non-SMOTE. Meskipun selisihnya tidak terlalu besar, perbedaan ini menunjukkan bahwa SMOTE tetap memberikan kontribusi positif terhadap stabilitas dan ketepatan prediksi model. Visualisasi ini juga mempertegas bahwa penyeimbangan data melalui SMOTE membantu model mengenali struktur kelas dengan lebih efektif, terutama pada kelas yang sebelumnya kurang terwakili.

### 3.5.4 AdaBoost

Tabel 3.4 Laporan Klasifikasi AdaBoost: SMOTE vs Non-SMOTE

Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.000	0.000	0.000	0.000	0.000	0.000
2	0.757	0.763	0.760	0.714	0.900	0.796
3	0.830	0.902	0.864	0.891	0.834	0.862
4	0.929	0.929	0.929	0.954	0.920	0.936

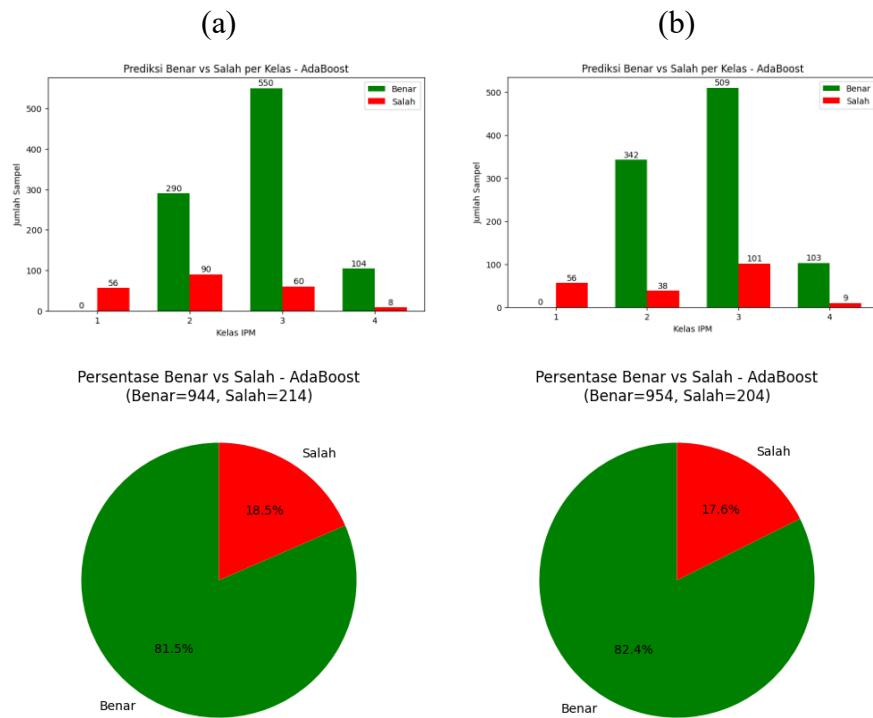
Berdasarkan Tabel 3.4, pada kelas 1 baik model dengan SMOTE maupun tanpa SMOTE tidak mampu menghasilkan prediksi yang baik karena nilai presisi, recall, dan F1-score seluruhnya bernilai 0, yang menunjukkan bahwa AdaBoost mengalami kesulitan mengenali kelas dengan jumlah sampel sangat sedikit meskipun telah dilakukan oversampling, sementara pada kelas 2 SMOTE hanya memberikan sedikit peningkatan presisi dengan recall pada model non-SMOTE tetapi lebih tinggi dan F1-score relatif seimbang, sedangkan pada kelas 3 dan kelas 4 performa model justru lebih baik tanpa SMOTE karena nilai presisi dan F1-score lebih tinggi, meskipun recall kelas 3 meningkat ketika SMOTE diterapkan, yang menandakan adanya peningkatan kemampuan menangkap sampel benar tetapi dengan penurunan ketepatan prediksi, sehingga secara keseluruhan pola ini menunjukkan bahwa AdaBoost tidak memperoleh peningkatan performa yang konsisten dari penggunaan SMOTE dan cenderung lebih stabil tanpa SMOTE, khususnya pada kelas 3 dan 4, serta menegaskan bahwa strategi penyeimbangan data tidak selalu berdampak positif untuk semua algoritma dan perlu disesuaikan dengan karakteristik model serta distribusi data.



Gambar 3.7 Matriks Evaluasi AdaBoost: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.7 menunjukkan perbandingan Matriks Evaluasi Klasifikasi model AdaBoost dengan dan tanpa SMOTE, di mana pada Gambar 3.7(a) penerapan SMOTE belum mampu meningkatkan kemampuan model secara signifikan dalam mengenali kelas-kelas IPM karena kelas 1 masih tidak dapat diprediksi dengan benar dan banyak dipetakan ke kelas 2 atau 3, sementara pada kelas 2, 3, dan 4 meskipun performa relatif cukup baik tetapi terdapat kesalahan yang menandakan SMOTE belum efektif membantu penangkapan karakteristik

kelas minoritas, sedangkan pada Gambar 3.7(b) model tanpa SMOTE justru menunjukkan performa yang lebih stabil dengan jumlah prediksi benar yang lebih tinggi pada kelas 2 dan 3 serta kesalahan yang lebih rendah, yang mengindikasikan bahwa sensitivitas AdaBoost terhadap pembaruan bobot kesalahan membuat penambahan data sintetis berpotensi mengganggu proses pembelajaran, sehingga secara keseluruhan hasil ini menegaskan bahwa AdaBoost bekerja lebih baik tanpa SMOTE, selaras dengan temuan pada Tabel 3.4, dan memperkuat kesimpulan bahwa penerapan SMOTE tidak selalu cocok untuk semua algoritma serta perlu disesuaikan dengan karakteristik model dan distribusi data.



Gambar 3.8 Visualisasi Prediksi AdaBoost: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.8 menyajikan visualisasi prediksi benar dan salah pada model AdaBoost, baik dengan SMOTE maupun tanpa SMOTE. Dari Gambar 3.8(a), terlihat bahwa penggunaan SMOTE menghasilkan tingkat akurasi yang lebih rendah dibandingkan model non-SMOTE. Jumlah prediksi benar pada model dengan SMOTE mencapai 944 sampel, sedangkan model tanpa SMOTE pada Gambar 3.8(b) menunjukkan prediksi benar yang lebih tinggi, yaitu 954 sampel. Selisih ini memberikan indikasi awal bahwa SMOTE tidak memberikan

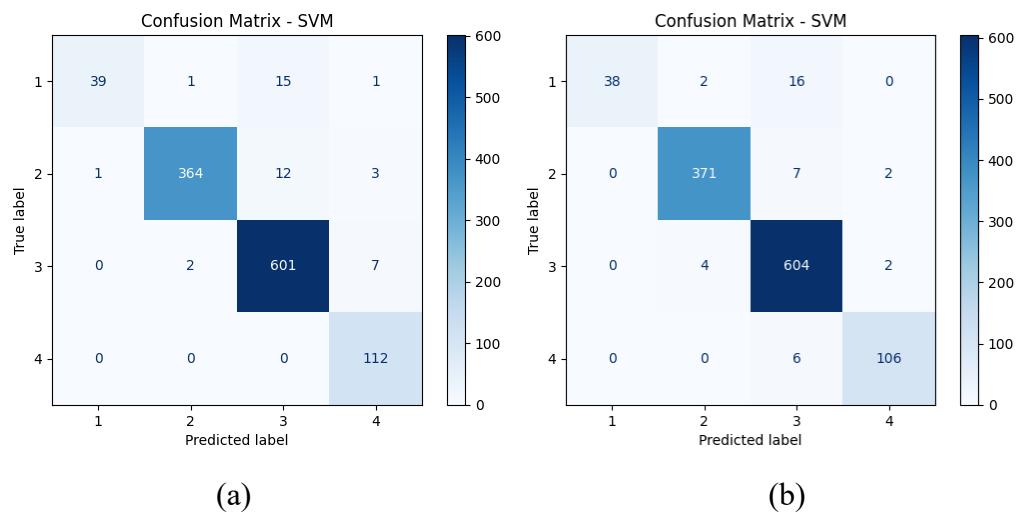
kontribusi signifikan dalam meningkatkan performa AdaBoost, bahkan cenderung sedikit menurunkannya.

### 3.5.5 Support Vector Machine (SVM)

Tabel 3.5 Laporan Klasifikasi SVM: SMOTE vs Non-SMOTE

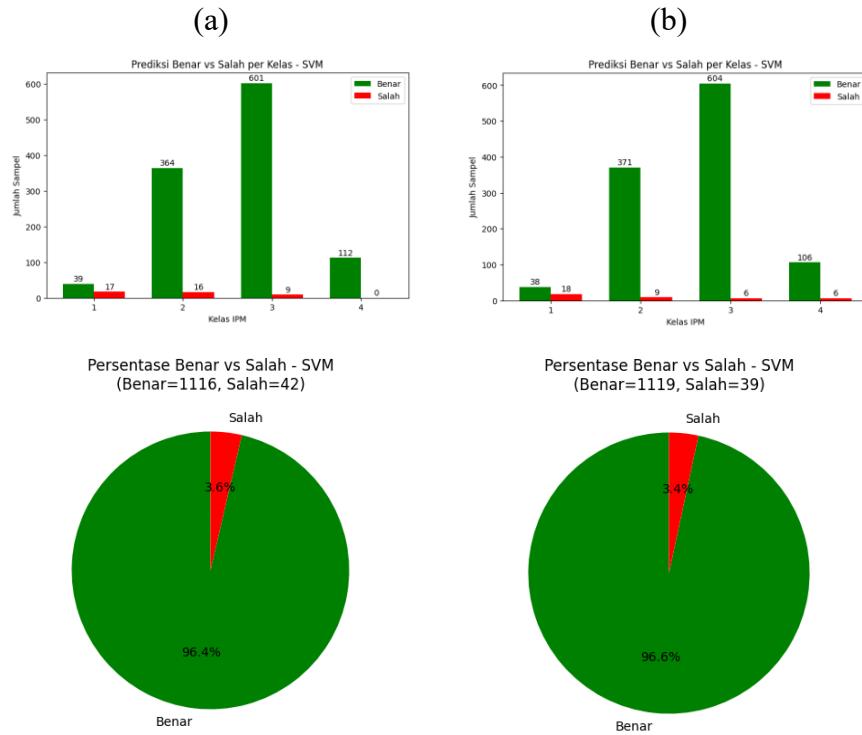
Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.975	0.696	0.812	1.000	0.679	0.809
2	0.992	0.958	0.975	0.984	0.976	0.980
3	0.957	0.985	0.971	0.954	0.990	0.972
4	0.911	1.000	0.953	0.964	0.946	0.955

Tabel 3.5 menunjukkan bahwa SVM memiliki performa yang kuat pada semua kelas, baik dengan SMOTE maupun tanpa SMOTE. Penggunaan SMOTE terutama memberikan peningkatan pada recall, seperti terlihat pada kelas 1 dan kelas 4, di mana kemampuan model menangkap seluruh sampel kelas meningkat. Namun, presisi pada beberapa kelas sedikit lebih rendah dibandingkan model non-SMOTE. Untuk kelas 2 dan 3, kedua model menunjukkan performa yang hampir serupa, dengan nilai presisi, recall, dan F1-score yang sangat tinggi. Secara keseluruhan, SMOTE memberikan perbaikan moderat pada sensitivitas model, sedangkan model non-SMOTE tetap unggul pada beberapa aspek ketepatan prediksi. Dengan demikian, kedua pendekatan sama-sama efektif, dan pemilihannya dapat disesuaikan dengan kebutuhan antara meningkatkan recall atau mempertahankan presisi.



Gambar 3.9 Matriks Evaluasi SVM: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.9 memperlihatkan Matriks Evaluasi Klasifikasi model SVM dengan dan tanpa SMOTE, di mana kedua model secara umum menunjukkan performa yang sangat kuat dengan jumlah prediksi benar yang tinggi pada hampir seluruh kelas IPM, namun pada model dengan SMOTE terlihat adanya peningkatan prediksi benar pada kelas 1 yang mengindikasikan meningkatnya sensitivitas model terhadap kelas minoritas, sementara kesalahan pada kelas 2 dan 3 tetap rendah, sedangkan pada model tanpa SMOTE distribusi prediksi tampak lebih stabil terutama pada kelas mayoritas dengan kesalahan yang sedikit lebih rendah pada kelas 3 dan 4 serta jumlah prediksi benar yang lebih tinggi pada kelas 2, sehingga menunjukkan bahwa SMOTE tidak selalu memberikan peningkatan signifikan pada seluruh kelas dan model non-SMOTE cenderung lebih unggul dalam menjaga ketepatan prediksi pada kelas dengan jumlah sampel besar.



Gambar 3.10 Visualisasi Prediksi SVM: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.10 menunjukkan bahwa performa model SVM dengan dan tanpa SMOTE hampir identik, dengan tingkat kesalahan masing-masing sebesar 3,6% dan 3,4%, sehingga oversampling tidak memberikan pengaruh signifikan terhadap kinerja keseluruhan. Meskipun SMOTE sedikit meningkatkan sensitivitas pada kelas minor, model non-SMOTE menunjukkan ketepatan yang

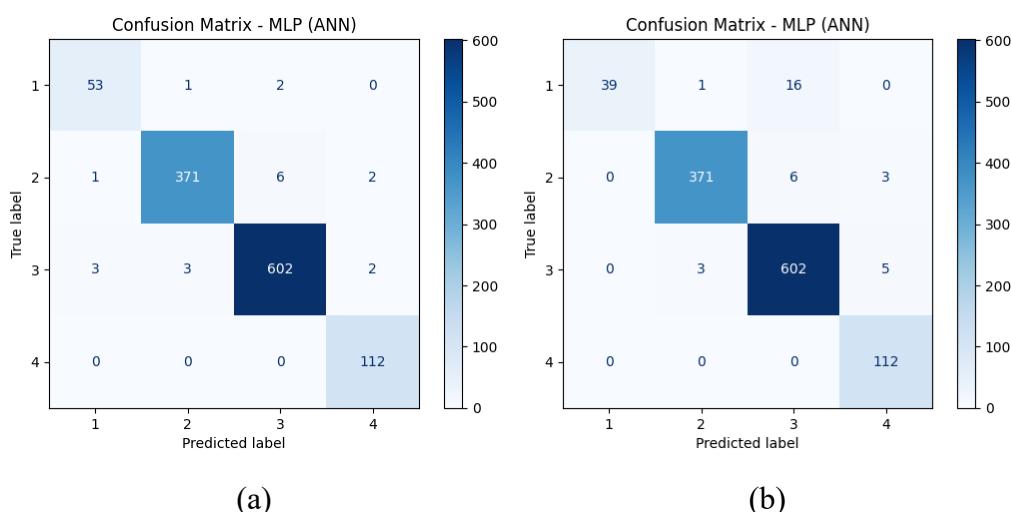
lebih baik pada beberapa kelas mayoritas, yang menegaskan bahwa SVM sudah bekerja sangat optimal pada distribusi data asli.

### 3.5.6 Artificial Neural Network (ANN)

Tabel 3.6 Laporan Klasifikasi ANN: SMOTE vs Non-SMOTE

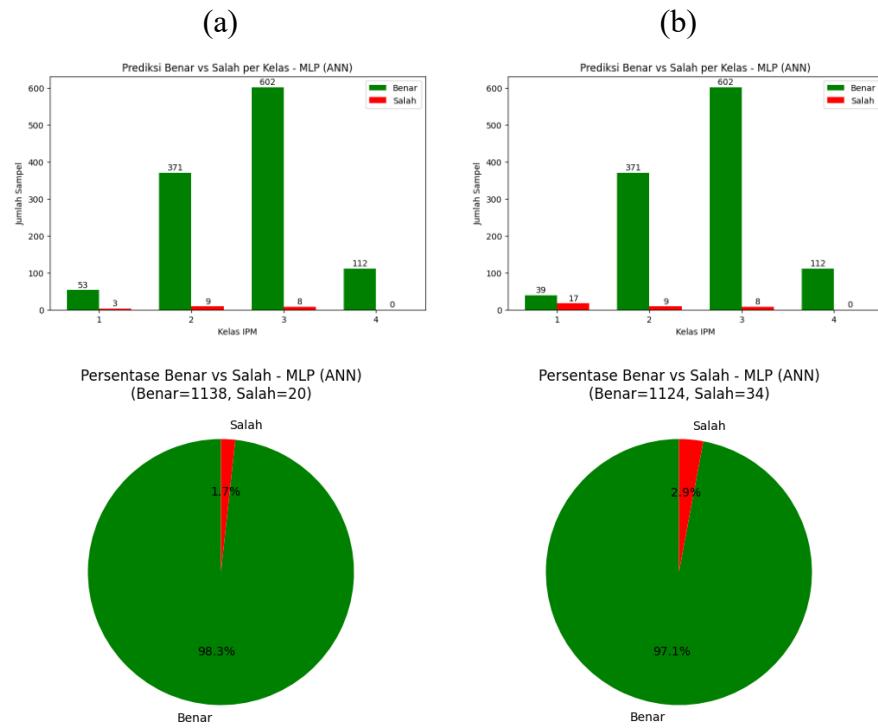
Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.930	0.946	0.938	1.000	0.696	0.821
2	0.989	0.976	0.983	0.989	0.976	0.983
3	0.987	0.987	0.987	0.965	0.987	0.976
4	0.966	1.000	0.982	0.933	1.000	0.966

Tabel 3.6 menunjukkan bahwa model ANN memiliki performa yang sangat baik pada seluruh kelas baik dengan maupun tanpa SMOTE, namun penerapan SMOTE memberikan peningkatan yang lebih jelas pada recall, terutama pada kelas 1 yang meningkat signifikan dari 0,696 menjadi 0,946 dan diikuti oleh kenaikan F1-score dari 0,821 menjadi 0,938, yang menandakan peningkatan keseimbangan antara presisi dan recall dalam mengenali kelas minoritas, sementara pada kelas 2 dan kelas 3 performa kedua model relatif identik dengan nilai metrik yang sangat tinggi, dan pada kelas 4 SMOTE meningkatkan recall hingga 1,000 meskipun presisi sedikit menurun namun tetap berada pada tingkat yang sangat baik, sehingga secara keseluruhan SMOTE membantu membuat performa ANN lebih konsisten antar kelas.



Gambar 3.11 Matriks Evaluasi ANN: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.11 menunjukkan Matriks Evaluasi Klasifikasi model ANN dengan dan tanpa SMOTE, di mana kedua model secara umum memiliki performa yang sangat baik, namun penerapan SMOTE memberikan peningkatan yang lebih jelas pada kelas 1 dengan jumlah prediksi benar meningkat dari 39 pada model non-SMOTE menjadi 53 pada model SMOTE, yang menegaskan efektivitas SMOTE dalam membantu ANN mengenali kelas minoritas, sementara pada kelas 2 dan kelas 3 performa kedua model hampir identik dengan tingkat kesalahan yang sangat rendah, dan pada kelas 4 kedua model menghasilkan jumlah prediksi benar yang sama, yaitu 112, sehingga menunjukkan bahwa ANN sudah mampu memisahkan kelas mayoritas dengan sangat baik tanpa memerlukan tambahan data sintetis.



Gambar 3.12 Visualisasi Prediksi ANN: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.12 menunjukkan bahwa penerapan SMOTE pada model ANN menghasilkan jumlah prediksi benar yang lebih tinggi, yaitu 1.138 dibandingkan 1.124 pada model non-SMOTE, serta tingkat kesalahan yang lebih rendah sebesar 1,7% dibandingkan 2,9%, yang menandakan peningkatan akurasi keseluruhan, terutama terlihat pada kelas 1 di mana model SMOTE mampu memprediksi lebih banyak sampel dengan benar, sementara pada kelas lainnya performa kedua model relatif serupa, sehingga secara keseluruhan visualisasi ini menegaskan bahwa SMOTE memberikan manfaat nyata bagi ANN dalam meningkatkan

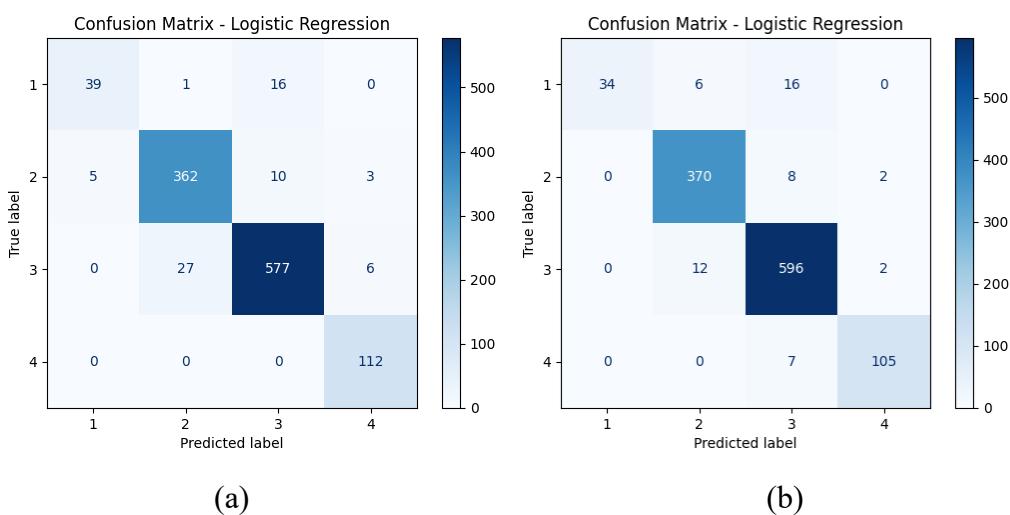
sensitivitas terhadap kelas minoritas, mengurangi kesalahan prediksi, dan menjaga kestabilan kinerja antar kelas.

### 3.5.7 Logistic Regression

Tabel 3.7 Laporan Klasifikasi Logistic Regression: SMOTE vs Non-SMOTE

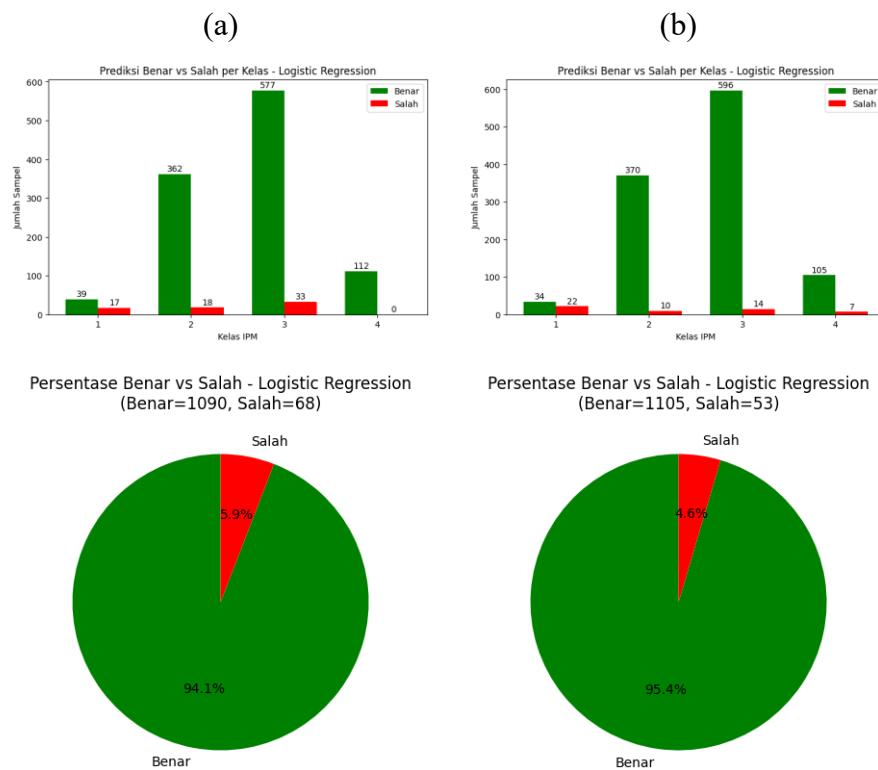
Kelas	SMOTE			Non-SMOTE		
	Presisi	Recall	F1-Score	Presisi	Recall	F1-Score
1	0.886	0.696	0.780	1.000	0.607	0.756
2	0.928	0.953	0.940	0.954	0.974	0.964
3	0.957	0.946	0.951	0.951	0.977	0.964
4	0.926	1.000	0.961	0.963	0.938	0.950

Tabel 3.7 menunjukkan model Logistic Regression memiliki performa yang stabil pada seluruh kelas baik dengan maupun tanpa SMOTE, namun penerapan SMOTE memberikan peningkatan yang jelas pada recall, khususnya pada kelas 1 yang meningkat dari 0,607 menjadi 0,696 dan pada kelas 4 yang mencapai nilai sempurna 1, sehingga menunjukkan peningkatan model dalam menangkap kelas minoritas meskipun presisi pada beberapa kelas sedikit menurun, sementara pada kelas 2 dan kelas 3 kedua model tetap menunjukkan performa sangat baik dengan perbedaan kecil antara presisi dan recall, yang menegaskan bahwa kontribusi utama SMOTE pada Logistic Regression adalah peningkatan sensitivitas model. Secara keseluruhan SMOTE memberikan peningkatan terutama pada kelas minor, sedangkan model non-SMOTE unggul dalam hal presisi di beberapa kelas.



Gambar 3.13 Matriks Evaluasi Logistic Regression: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.13 menunjukkan Matriks Evaluasi Klasifikasi Logistic Regression dengan dan tanpa SMOTE, di mana kedua model secara umum memiliki pola prediksi yang baik, namun penerapan SMOTE memberikan peningkatan yang lebih jelas pada kelas minoritas, khususnya kelas 1 dengan peningkatan prediksi benar dari 34 menjadi 39 serta kelas 4 dengan recall sempurna tanpa kesalahan prediksi, sementara pada kelas 2 dan kelas 3 performa kedua model relatif serupa dengan model non-SMOTE menunjukkan ketepatan sedikit lebih tinggi pada kelas mayoritas, sehingga secara keseluruhan hasil ini menegaskan bahwa SMOTE terutama meningkatkan sensitivitas Logistic Regression terhadap kelas minor, sedangkan model tanpa SMOTE tetap unggul dalam presisi pada beberapa kelas, selaras dengan temuan pada Tabel 3.7 dan menunjukkan bahwa pemilihan penggunaan SMOTE perlu disesuaikan dengan tujuan analisis.



Gambar 3.14 Visualisasi Prediksi Logistic Regression: SMOTE (a) vs Non-SMOTE (b)

Gambar 3.14 menunjukkan bahwa performa Logistic Regression dengan dan tanpa SMOTE relatif serupa, di mana model dengan SMOTE menghasilkan 1.090 prediksi benar dengan tingkat kesalahan 5,9%, sedangkan model tanpa SMOTE sedikit lebih baik dengan 1.105 prediksi benar dan kesalahan 4,6%, yang mengindikasikan bahwa Logistic Regression sudah bekerja optimal pada

distribusi data asli tanpa perlu oversampling, meskipun SMOTE memberikan sedikit peningkatan pada pengenalan kelas minor seperti kelas 1, sementara model non-SMOTE tetap unggul pada kelas mayoritas seperti kelas 2 dan 3, sehingga secara keseluruhan SMOTE hanya memberikan keuntungan terbatas dan Logistic Regression cenderung lebih stabil ketika dilatih pada data dengan distribusi asli.

Evaluasi di atas telah menunjukkan bahwa model dengan SMOTE lebih baik daripada yang tidak menggunakan SMOTE. Namun, evaluasi tersebut perlu juga melihat akurasinya dan log-loss, berikut tabel akurasi, log-loss, dan Weighted F1-score:

Tabel 3.8 Akurasi, Log-loss, Weighted-F1: SMOTE vs Non-SMOTE

Nama Model	SMOTE			Non-SMOTE		
	Accuracy	Weighted F1	Log-loss	Accuracy	Weighted F1	Log-loss
Decision Tree	0.911054	0.910501	3.205955	0.894646	0.894350	3.797345
Random Forest	0.933506	0.932519	0.259283	0.928325	0.927374	0.244688
Gradient Boosting	0.943869	0.942804	0.208718	0.938687	0.937415	0.226510
AdaBoost	0.815199	0.794440	1.340290	0.823834	0.805929	1.338794
SVM	0.963731	0.962741	0.201838	0.966321	0.965048	0.198663
ANN	0.982729	0.982749	0.081766	0.970639	0.969554	0.106151
Logistic Regression	0.941278	0.940399	0.443743	0.954231	0.952238	0.530161

Tabel 3.8 menunjukkan bahwa SMOTE memberikan peningkatan performa yang cukup konsisten pada sebagian besar model. Model-model tersebut memperoleh kenaikan akurasi dan weighted F1-score, yang menunjukkan bahwa augmentasi data sintetis membantu memperbaiki kemampuan model dalam mengenali kelas minoritas. Tabel 3.8 inilah menjadi patokan terakhir dalam mengevaluasi model. Dari sini dapat diketahui, bahwa Decision Tree yang awalnya kelihatan bagus, tetapi memiliki probabilitas yang buruk. Hanya 2 model saja yang mengalami overfitting, yaitu Decision

Tree dan AdaBoost. Model paling terbaik adalah ANN dengan SMOTE. Sisanya, model masih bagus dan dapat digunakan.

ANN memang bagus, karena dia membentuk aturan sendiri melalui jaringan saraf yang terhubung pada lapisan masukkan, lapisan tersembunyi, sampai dengan lapisan keluaran. Sebab itu, ANN memberikan performa yang sesuai. Dari semua evaluasi model, didapatkan tabel pemeringkatan sebagai berikut.

Tabel 3.9 Tabel Peringkat Model

Rank	Model	Keterangan
1	ANN - SMOTE	Performa terbaik; akurasi, F1, dan log-loss paling unggul.
2	ANN - Non-SMOTE	Sangat stabil; hanya sedikit di bawah versi SMOTE.
3	SVM - Non-SMOTE	Akurasi dan F1 tinggi; sangat konsisten tanpa SMOTE.
4	SVM - SMOTE	Performa kuat; peningkatan kecil dari SMOTE.
5	Gradient Boosting - SMOTE	Akurasi dan F1 tinggi; cocok untuk pola kompleks.
6	Gradient Boosting - Non-SMOTE	Performa hampir sama dengan SMOTE; sangat stabil.
7	Logistic Regression - Non-SMOTE	Akurasi baik; probabilitas relatif stabil.
8	Logistic Regression - SMOTE	Sedikit lebih rendah karena log-loss meningkat.
9	Random Forest - SMOTE	Performa baik; namun masih di bawah Gradient Boosting.
10	Random Forest - Non-SMOTE	Performa mirip SMOTE; peningkatan minim.
11	Decision Tree - SMOTE	Akurasi meningkat, tetapi probabilitas kurang stabil.
12	Decision Tree - Non-SMOTE	Performa rendah; cenderung overfit.
13	AdaBoost - Non-SMOTE	Akurasi dan F1 rendah; kurang baik untuk data tidak seimbang.
14	AdaBoost - SMOTE	Performa terendah; tidak cocok untuk dataset ini.

Pemeringkatan ini tidak hanya didasari loss dan akurasi saja, tetapi semua yang telah dilakukan. Loss yang rendah bukan berarti itu bagus, tetapi yang bagus adalah loss rendah, akurasi tinggi, Weighted F1-Scorenya juga tinggi. Loss tinggi tapi tidak adanya indikasi underfit maupun overfit, tetapi F1-Scorenya tinggi, recall tinggi, akurasi tinggi, dan metrik lainnya tinggi, maka itu yang dipilih.

Dari rangkaian evaluasi tersebut disimpulkan bahwa ANN merupakan model yang paling layak digunakan untuk klasifikasi IPM dalam penelitian ini. ANN tidak hanya memberikan akurasi dan F1-score tertinggi, tetapi juga menghasilkan nilai log-loss paling rendah, sehingga model ini lebih dapat diandalkan dalam melakukan prediksi dengan tingkat kepercayaan tinggi. Selain itu, performa ANN yang konsisten pada kelas mayoritas maupun minoritas menjadikan model ini sebagai pilihan paling optimal untuk pengembangan sistem prediksi berkelanjutan pada penelitian ini.

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Eksperimen yang dilakukan memberikan banyak informasi dan pengetahuan. Informasi dan pengetahuan yang dapat disimpulkan adalah sebagai berikut:

- 4.1.1 Variabel kesehatan, pendidikan, dan kesejahteraan terbukti memiliki korelasi yang kuat dengan IPM.
- 4.1.2 Permasalahan utama dataset adalah imbalance kelas IPM, dan penggunaan SMOTE terbukti meningkatkan performa sebagian besar model untuk mempelajari kelas minoritas.
- 4.1.3 Perbandingan tujuh model Machine Learning menunjukkan bahwa Artificial Neural Network (ANN) dengan SMOTE merupakan model yang paling optimal.
- 4.1.4 SMOTE terbukti efektif, tetapi tidak cocok untuk semua model.
- 4.1.5 Hasil pemeringkatan akhir (1-14) konsisten dengan evaluasi numerik dan analisis visual (confusion matrix dan distribusi salah–benar), sehingga model terbaik sudah ditentukan secara komprehensif.
- 4.1.6 Model paling buruk adalah AdaBoost, yang menunjukkan kondisi overfit.

Secara keseluruhan, hasil eksperimen ini memberikan gambaran yang jelas mengenai perilaku tiap model dalam memprediksi tingkat pembangunan manusia, sekaligus menunjukkan pentingnya penanganan ketidakseimbangan data dalam meningkatkan kualitas prediksi. Temuan ini tidak hanya memperkuat validitas pemilihan model terbaik, tetapi juga memberikan dasar yang kuat untuk pengembangan sistem prediksi IPM yang lebih akurat dan andal di masa mendatang.

#### **4.2 Saran**

Berdasarkan hasil penelitian, evaluasi model, serta analisis performa dengan dan tanpa SMOTE, beberapa saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

- 4.2.1 Menerapkan teknik penanganan imbalance lain selain SMOTE
- 4.2.2 Menguji dan mengubah arsitektur dan model yang lebih kompleks
- 4.2.3 Melakukan *hyperparameter tuning* secara menyeluruh
- 4.2.4 Mengintegrasikan model ke dalam sistem prediksi berbasis web atau dashboard
- 4.2.5 Mempertimbangkan penggunaan ensemble berbasis voting atau stacking

## 4.3 Lampiran

### 4.3.1 Link BPS:

- 4.3.1.1 IPM: <https://www.bps.go.id/id/statistics-table/2/MjIwNyMy/-metode-baru--indeks-pembangunan-manusia-menurut-provinsi--umur-harapan-hidup-hasil-long-form-sp2020-.html>
  - 4.3.1.2 UHH: <https://www.bps.go.id/id/statistics-table/2/NDE0IzI=/-metode-baru--umur-harapan-hidup-saat-lahir--uhh---tahun-.html>
  - 4.3.1.3 HLS: <https://www.bps.go.id/id/statistics-table/2/NDE3IzI=/-metode-baru--harapan-lama-sekolah--tahun-.html>
  - 4.3.1.4 RLS: <https://www.bps.go.id/id/statistics-table/2/NDE1IzI=/-metode-baru--rata-rata-lama-sekolah--tahun-.html>
  - 4.3.1.5 PPP: <https://www.bps.go.id/id/statistics-table/2/NDE2IzI=/-metode-baru--pengeluaran-per-kapita-disesuaikan--ribu-rupiah-orang-tahun-.html>
- 4.3.2 Link Github: <https://github.com/wilhelm213/IPMClassification1.git>

## DAFTAR PUSTAKA

- Aggarwal, C. C. (2023). *Neural Networks and Deep Learning*. Springer.
- Badan Pemeriksa Keuangan Republik Indonesia. (2022). *Undang-Undang Nomor 16 Tahun 2022 tentang Pembentukan Provinsi Papua Pegunungan*.  
<https://peraturan.bpk.go.id/Details/217799/uu-no-16-tahun-2022>
- Badan Pusat Statistik. (n.d.). *Indeks Pembangunan Manusia Metode Baru*. Badan Pusat Statistik.
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly.
- Han, J., Kamber, M., & Pei, J. (2021). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann.
- JDIH Kota Tanjungpinang. (2022). *Undang-Undang Nomor 29 Tahun 2022 tentang Pembentukan Provinsi Papua Barat Daya*.  
<https://jdih.tanjungpinangkota.go.id/cariprodukhukum/1736>
- Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). *Chapter 15: Exploratory Data Analysis*. In MIT Critical Data (Ed.), *Secondary Analysis of Electronic Health Records* (pp. 185–203). Springer. [https://doi.org/10.1007/978-3-319-43742-2\\_15](https://doi.org/10.1007/978-3-319-43742-2_15)
- Pedregosa, F., et al. (2024). *Comparative study on the performance of categorical variable encoders in classification and regression tasks*. arXiv:2401.09682.  
<https://arxiv.org/abs/2401.09682>
- Zhu, J., Zou, H., Rosset, S., & Hastie, T. (2020). Multi-class AdaBoost. *Journal of Statistical Software*, 93(3), 1–20.