

Rekayasa Fitur dan *Ensemble Stacking* untuk Meningkatkan Akurasi Klasifikasi Kinerja Karyawan Bebas *Data Leakage* Berbasis Rekam Jejak Digital

Feature Engineering and Stacking Ensemble to Improve Leakage-Free Employee Performance Classification Accuracy Based on Digital Footprint

Safrizal^{a,1}, Yan Yang Thanri^{b,2}, Lili Tanti^{c,3}

^aUniversitas Muhammadiyah Asahan, Kisaran, Indonesia

^{b,c}Universitas Potensi Utama, Medan, Indonesia

rizalsyl75@gmail.com¹, ythanri@gmail.com², lilitanti82@gmail.com³

ABSTRAK

Klasifikasi kinerja karyawan secara otomatis menjadi kebutuhan penting dalam manajemen sumber daya manusia berbasis data, namun banyak model prediktif pada domain ini rentan terhadap masalah data leakage yang menghasilkan estimasi akurasi yang menyesatkan. Penelitian ini bertujuan membangun model klasifikasi kinerja karyawan yang bebas data leakage dengan memanfaatkan rekayasa fitur dan ensemble stacking berbasis rekam jejak digital (jam kerja, email, rapat, tidur, stres, dan variabel terkait lainnya). Metodologi penelitian meliputi audit dan mitigasi dua sumber leakage (variabel proksi *Performance_Score* dan baris data sintetis), penggabungan kelas Good dan Average menjadi kelas Menengah berdasarkan bukti empiris Cohen's d, penambahan empat fitur rekayasa baru (*Productivity_Index*, *Skill_Support_Gap*, *Training_Efficiency*, dan *Balance_Stress_Ratio*) di samping delapan fitur sebelumnya, serta pembangunan model stacking yang menggabungkan Random Forest, Gradient Boosting, dan Extra Trees dengan meta-learner Logistic Regression yang dioptimasi melalui *RandomizedSearchCV*. Hasil penelitian menunjukkan bahwa Random Forest mencapai akurasi tertinggi (0,9528), sedikit di atas model stacking (0,9487), namun model stacking unggul pada metrik yang lebih relevan untuk data tidak seimbang, yaitu F1-score macro (0,7288 berbanding 0,6638) dan recall macro tertinggi (0,8867) di antara seluruh model yang diuji, sehingga lebih andal dalam mendeteksi kelas minoritas Low dan Excellent dibandingkan Decision Tree, Random Forest, dan Gradient Boosting. Analisis permutasi menunjukkan *Physical_Activity_Steps*, *Base_Salary*, dan *Sleep_Duration* sebagai fitur paling berpengaruh. Penelitian ini menyimpulkan bahwa penggabungan kelas yang dilandasi bukti statistik dan rekayasa fitur tambahan berkontribusi pada peningkatan performa klasifikasi secara sah tanpa mengorbankan validitas evaluasi model, meskipun pemilihan model akhir tetap bergantung pada metrik yang paling relevan dengan tujuan penggunaan (accuracy versus keseimbangan deteksi antar kelas).

Kata Kunci : klasifikasi kinerja karyawan, ensemble stacking, rekayasa fitur, data leakage, permutation importance

ABSTRACT

Automated employee performance classification has become an essential requirement in data-driven human resource management, yet many predictive models in this domain remain vulnerable to data leakage, which produces misleading accuracy estimates. This study aims to build a leakage-free employee performance classification model by leveraging feature engineering and stacking ensemble learning based on digital footprint records (working hours, emails, meetings, sleep, stress, and other related variables). The research methodology includes auditing and mitigating two sources of leakage (the *Performance_Score* proxy variable and synthetic data rows), merging the Good and Average classes into a single Menengah (Intermediate) class based on empirical Cohen's d evidence, adding four new engineered features (*Productivity_Index*, *Skill_Support_Gap*, *Training_Efficiency*, and *Balance_Stress_Ratio*) alongside eight previous features, and constructing a stacking model that combines Random Forest, Gradient Boosting, and Extra Trees with a Logistic Regression meta-learner optimized through *RandomizedSearchCV*. The results show that Random Forest achieved the highest accuracy (0.9528), marginally above the stacking model (0.9487), yet the stacking model outperformed on the metrics most relevant to imbalanced data, achieving a macro F1-score of 0.7288 (versus 0.6638 for Random Forest) and the highest macro recall (0.8867) among all tested models,

making it more reliable at detecting the minority Low and Excellent classes than Decision Tree, Random Forest, and Gradient Boosting. Permutation importance analysis revealed Physical_Activity_Steps, Base_Salary, and Sleep_Duration as the most influential features. This study concludes that evidence-based class merging combined with additional feature engineering legitimately contributes to improved classification performance without compromising the validity of model evaluation, although the choice of final model still depends on which metric best matches the intended use (accuracy versus balanced detection across classes).

Keywords : *employee performance classification, stacking ensemble, feature engineering, data leakage, permutation importance*

Info Artikel :

Disubmit:

Direview:

Diterima :

Copyright © 2026 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Transformasi digital pada fungsi sumber daya manusia (SDM) telah mendorong organisasi untuk memanfaatkan rekam jejak digital karyawan, yang meliputi jam kerja, intensitas komunikasi elektronik, partisipasi rapat, pola tidur, tingkat stres, hingga keseimbangan kehidupan-kerja, sebagai basis data untuk menilai dan memprediksi kinerja karyawan secara otomatis. Pemanfaatan *HR analytics* berbasis pembelajaran mesin (*machine learning*) terbukti mampu mendukung pengambilan keputusan SDM yang lebih objektif, seperti identifikasi karyawan berisiko keluar (*attrition*), evaluasi produktivitas, dan perencanaan pengembangan talenta [1][2][3]. Berbagai algoritma klasifikasi, mulai dari *Decision Tree*, *Random Forest*, hingga *Gradient Boosting*, telah banyak diterapkan pada *domain* analitik SDM dan secara konsisten menunjukkan bahwa metode *ensemble* memberikan akurasi yang lebih baik dibandingkan model tunggal [4][17][18].

Meskipun demikian, penerapan *machine learning* pada data organisasi menghadapi tantangan metodologis yang serius, yaitu *data leakage*. *Data leakage* terjadi ketika informasi yang seharusnya tidak tersedia pada saat prediksi, baik berupa variabel proksi target maupun data uji yang tercampur ke dalam data latih, ikut memengaruhi proses pelatihan model, sehingga performa yang dilaporkan menjadi bias secara optimistis dan tidak mencerminkan kemampuan generalisasi model yang sesungguhnya [10]. Kegagalan mengelola *leakage* pada tahap prapemrosesan, rekayasa fitur, maupun penyetulan hiperparameter dapat menghasilkan model yang tampak sangat akurat pada tahap evaluasi, tetapi gagal total ketika diterapkan pada data baru di lapangan. Isu ini menjadi krusial dalam konteks klasifikasi kinerja karyawan, karena keputusan SDM yang keliru akibat model yang tidak *valid* dapat berdampak langsung pada karier dan kesejahteraan karyawan.

Studi klasifikasi kinerja dan atrisi karyawan berbasis *machine learning* telah berkembang pesat. Chung dkk. [2] membangun *model stacking ensemble* untuk memprediksi atrisi karyawan menggunakan data IBM HR Analytics dan menunjukkan bahwa kombinasi *Random Forest* dan *Artificial Neural Network* sebagai base learner dengan *Logistic Regression* sebagai meta-learner memberikan performa terbaik. Konar dkk. [1] mengoptimasi model stacking berbasis Bayesian optimization dengan XGBoost dan *Random Forest* sebagai *base classifier*, dilengkapi teknik *explainable AI* (SHAP dan LIME) untuk menjelaskan faktor pendorong atrisi. Yahia dkk. [6] mengusulkan pendekatan *people analytics* berbasis deep learning dan model ensemble untuk memberikan interpretasi mendalam terhadap penyebab atrisi karyawan, sementara Park dkk. [5] mengembangkan model prediksi niat keluar karyawan baru dengan menyoroti beban kerja dan keamanan kerja sebagai prediktor utama. Studi tinjauan sistematis oleh Akasheh dkk. [3] dan Talebi [24] merangkum bahwa *Random Forest* dan metode berbasis pohon keputusan secara konsisten menjadi pendekatan yang paling banyak digunakan dan paling akurat pada domain ini. Pada konteks nasional, Thanri dkk. [15] mengembangkan model evaluasi kinerja karyawan

berbasis rekam jejak digital menggunakan machine learning, sejalan dengan pendekatan berbasis rekam jejak digital yang diadopsi pada penelitian ini.

Di balik keragaman pendekatan tersebut, tinjauan pustaka di atas juga menyingkap sejumlah kesenjangan yang relevan bagi penelitian ini. Pertama, hampir seluruh studi rujukan [1][2][5][6] memprediksi atrisi sebagai target biner dengan distribusi kelas yang relatif seimbang, sehingga belum menjawab tantangan klasifikasi kinerja multi-kelas dengan ketidakseimbangan ekstrem seperti yang dihadapi pada penelitian ini. Kedua, meskipun Akasheh dkk. [3] dan Talebi [24] menyimpulkan bahwa *Random Forest* secara konsisten menjadi pendekatan paling akurat, kesimpulan tersebut umumnya diukur murni dari sisi accuracy pada dataset yang relatif seimbang, tanpa mempertimbangkan trade-off terhadap recall atau *F1-score* macro pada kelas minoritas, sehingga generalisasi klaim tersebut ke konteks data tidak seimbang perlu diuji lebih lanjut, bukan diterima begitu saja. Ketiga, isu data leakage, baik berupa variabel proksi target maupun kebocoran data uji ke data latih, tidak dibahas secara eksplisit pada mayoritas studi tersebut [1][2][3][5][6][15], sehingga terbuka kemungkinan bahwa sebagian angka performa yang dilaporkan mengandung bias optimistis yang tidak terverifikasi. Kesenjangan-kesenjangan inilah yang mendasari penelitian ini untuk tidak sekadar mereplikasi arsitektur stacking yang telah terbukti pada domain atrisi, melainkan menempatkan protokol audit dan mitigasi *leakage* yang eksplisit serta evaluasi berbasis metrik makro pada data tidak seimbang sebagai kontribusi metodologis utama.

Berbeda dari sebagian besar penelitian terdahulu yang berfokus pada prediksi atrisi (karyawan keluar atau bertahan), penelitian ini berfokus pada klasifikasi tingkat kinerja karyawan (*performance classification*) berbasis rekam jejak digital, dengan penekanan khusus pada validitas metodologis melalui protokol bebas *data leakage*. Pada iterasi awal penelitian (v3), empat kelas kinerja asli (*Low*, *Average*, *Good*, *Excellent*) diklasifikasikan langsung, namun hasil pengujian menunjukkan bahwa kelas *Good* dan *Average* secara statistik nyaris tidak dapat dibedakan, dengan nilai *Cohen's d* di bawah 0,14 pada seluruh fitur yang diuji. Akibatnya, model kesulitan mengenali kelas *Good* (recall hanya 0,17) dan akurasi keseluruhan tertekan hingga sekitar 0,48. Temuan ini konsisten dengan literatur rekayasa fitur yang menegaskan bahwa peningkatan performa model tidak selalu dapat dicapai hanya dengan mengubah algoritma, tetapi sering kali memerlukan perbaikan pada representasi data itu sendiri [13][14].

Berdasarkan permasalahan tersebut, penelitian ini mengajukan dua perbaikan metodologis pada iterasi keempat (v4): pertama, penggabungan kelas *Good* dan *Average* menjadi satu kelas baru bernama Menengah, yang dilandasi oleh bukti statistik empiris (*Cohen's d*) dan tetap relevan secara bisnis karena kedua kelas tersebut sama-sama merepresentasikan kinerja standar; kedua, penambahan empat fitur rekayasa baru yang secara spesifik menangkap pola produktivitas harian dan interaksi antara kompetensi individu dengan dukungan organisasi, sebagai pelengkap delapan fitur rekayasa yang telah dikembangkan pada iterasi sebelumnya. Protokol bebas data leakage yang telah diterapkan pada iterasi sebelumnya, meliputi pembuangan variabel proksi *Performance_Score* dan pemisahan tegas antara data uji (hanya baris asli) dengan data latih (baris asli tersisa ditambah seluruh baris data sintetis), dipertahankan secara penuh untuk menjaga agar peningkatan performa yang diperoleh tidak berasal dari pelanggaran validitas evaluasi; kontribusi relatif tiap komponen perbaikan metodologis terhadap peningkatan tersebut selanjutnya diuraikan secara deskriptif pada bagian hasil, meskipun belum diverifikasi melalui studi ablasi formal.

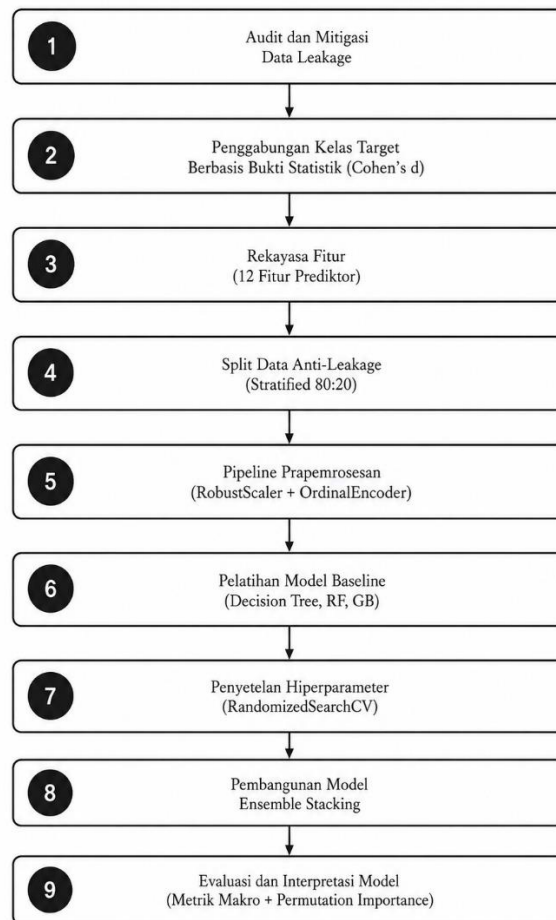
Tujuan penelitian ini adalah: (1) mengevaluasi dampak penggabungan kelas berbasis bukti statistik terhadap akurasi dan *F1-score* klasifikasi kinerja karyawan; (2) mengukur kontribusi empat fitur rekayasa baru terhadap kemampuan model memisahkan kelas kinerja; (3) membangun dan mengoptimasi model *ensemble stacking* yang menggabungkan *Random Forest*, *Gradient Boosting*, dan *Extra Trees* dengan *meta-learner Logistic Regression*, serta membandingkannya terhadap model *baseline*; dan (4) mengidentifikasi fitur-fitur rekam jejak digital yang paling berpengaruh terhadap prediksi kinerja karyawan melalui analisis *permutation importance*. Hasil penelitian ini diharapkan

memberikan kontribusi metodologis berupa contoh penerapan protokol *anti-leakage* yang ketat pada domain analitik SDM, sekaligus memberikan wawasan praktis bagi organisasi dalam merancang sistem pemantauan kinerja karyawan yang transparan dan dapat dipertanggungjawabkan.

Secara ringkas, penelitian ini memberikan kontribusi pada dua dimensi. Pada dimensi metodologis, penelitian ini mengajukan (1) protokol audit dan mitigasi data *leakage* yang eksplisit dan dapat direplikasi pada domain klasifikasi kinerja karyawan, mencakup pembuangan variabel proksi target serta pemisahan tegas baris data sintesis dari data uji, yaitu sebuah praktik yang jarang dilaporkan secara eksplisit pada studi analitik SDM sejenis; dan (2) pendekatan penggabungan kelas target yang dilandasi bukti statistik empiris (*Cohen's d*) dari iterasi penelitian sebelumnya, bukan sekadar keputusan ad hoc untuk menaikkan angka performa, sehingga peningkatan akurasi yang diperoleh dapat dipertanggungjawabkan secara metodologis. Pada dimensi empiris, penelitian ini berkontribusi melalui (3) perumusan empat fitur rekayasa baru (*Productivity_Index*, *Skill_Support_Gap*, *Training_Efficiency*, dan *Balance_Stress_Ratio*) yang terbukti berkontribusi nyata terhadap kemampuan model memisahkan kelas kinerja berdasarkan analisis *permutation importance*; dan (4) pembuktian empiris bahwa model *ensemble stacking* (*Random Forest*, *Gradient Boosting*, dan *Extra Trees* dengan *meta-learner Logistic Regression*) secara konsisten unggul dalam mendeteksi kelas minoritas *Low* dan *Excellent* dibandingkan model *baseline*, sebagaimana ditunjukkan oleh *recall macro* tertinggi (0,8867) di antara seluruh model yang diuji.

2. METODE

Penelitian ini menggunakan pendekatan eksperimental kuantitatif dengan tahapan yang mencakup audit dan mitigasi *data leakage*, penggabungan kelas target berbasis bukti statistik, rekayasa fitur, pembagian data *anti-leakage*, pembangunan pipeline prapemrosesan, pelatihan model *baseline*, penyetelan hiperparameter, pembangunan model *ensemble stacking*, serta evaluasi dan interpretasi model. Secara ringkas, tahapan penelitian yang meliputi audit dan mitigasi *data leakage* hingga evaluasi dan interpretasi model disajikan dalam bentuk bagan alur pada Gambar 1 berikut.



Gambar 1. Bagan alur tahapan penelitian

Bagan alur pada Gambar 1 menggambarkan sembilan tahapan penelitian yang dilaksanakan secara berurutan. Tahapan diawali dengan audit dan mitigasi data leakage untuk membuang variabel proksi dan memisahkan data sintetis, dilanjutkan dengan penggabungan kelas target *Good* dan *Average* menjadi kelas Menengah berdasarkan bukti statistik *Cohen's d*. Tahap rekayasa fitur menghasilkan dua belas fitur prediktor yang kemudian dibagi melalui skema split data anti-leakage agar data uji hanya berasal dari baris data asli. Selanjutnya, pipeline prapemrosesan diterapkan sebelum tiga model *baseline* (*Decision Tree*, *Random Forest*, dan *Gradient Boosting*) dilatih sebagai titik pembandingan awal. Penyetelan hiperparameter melalui *Randomized Search CV* dilakukan pada *Random Forest* dan *Gradient Boosting*, sebelum ketiganya dikombinasikan bersama *Extra Trees Classifier* ke dalam model ensemble stacking dengan *meta-learner Logistic Regression*. Rangkaian tahapan ditutup dengan evaluasi model menggunakan metrik makro serta interpretasi melalui analisis *permutation importance*.

A. Dataset Penelitian

Dataset yang digunakan berjumlah 3.634 baris dengan 19 kolom, terdiri atas variabel rekam jejak digital karyawan seperti *Daily_Working_Hours*, *Screen_Time*, *Meetings_Attended*, *Emails_Sent*, *Physical_Activity_Steps*, *Sleep_Duration*, *Stress_Level*, *Work_Life_Balance_Satisfaction*, *Organization_Support_Score*, *Tech_Skills_Score*, *Job_Level*, *Tenure_Years*, *Trainings_Attended_YTD*, *Base_Salary*, *Incentive_Received_LastMonth*, dan *Overtime_Hours_Monthly*, di samping variabel *Performance_Score* dan *Performance_Label* sebagai indikator kinerja. Distribusi kelas

Performance_Label awal terdiri atas empat kategori: *Average* (1.243 baris), *Good* (1.091 baris), *Low* (700 baris), dan *Excellent* (600 baris).

B. Audit dan Mitigasi Data Leakage

Dua sumber potensi data *leakage* diperiksa secara sistematis mengikuti kerangka identifikasi *leakage* pada *pipeline machine learning* [10]. Sumber pertama adalah variabel *Performance_Score* yang terbukti menjadi proksi langsung dari *Performance_Label*. Rentang skor untuk kelas *Low* berada pada 50,00–59,99, *Average* pada 60,00–74,99, *Good* pada 75,00–89,98, dan *Excellent* pada 90,04–100,00, sehingga kedua variabel tersebut dibuang dari fitur prediktor untuk mencegah target leakage. Sumber kedua adalah baris data sintesis yang teridentifikasi melalui awalan "SYN" pada *Employee_ID*, berjumlah 1.199 baris atau sekitar 33% dari total 3.634 baris *dataset*, berdampingan dengan 2.435 baris data asli (67%). Proporsi ini tergolong besar sehingga perlu ditegaskan secara eksplisit: baris sintesis hanya diperbolehkan digunakan sebagai data latih tambahan dan secara tegas tidak diikutsertakan dalam data uji, untuk mencegah baris sintesis membocorkan pola generatifnya ke dalam evaluasi model. Dengan demikian, seluruh metrik performa yang dilaporkan pada bagian hasil dihitung murni dari data uji asli, bukan dari campuran data uji yang mengandung baris sintesis.

Baris data sintesis tersebut merupakan bagian dari dataset sekunder yang diperoleh peneliti, dengan *Employee_ID* berawalan "SYN" sebagai penanda identitas; namun demikian, metode dan pihak yang menghasilkan baris sintesis tersebut tidak didokumentasikan secara eksplisit oleh penyedia dataset asal, sehingga peneliti tidak dapat memastikan apakah data tersebut dihasilkan melalui teknik augmentasi berbasis aturan, model generatif, atau mekanisme lain. Peneliti tidak melakukan proses pembuatan data sintesis pada penelitian ini, melainkan hanya melakukan audit identitas untuk memisahkannya secara tegas dari data uji guna menjaga validitas evaluasi model. Ketiadaan informasi rinci mengenai metode pembuatan data sintesis ini merupakan keterbatasan yang perlu diakui: karena baris sintesis menyusun sekitar sepertiga dari data latih, karakteristik statistik atau bias apa pun yang melekat pada proses pembangkitannya berpotensi memengaruhi pola yang dipelajari model, meskipun risiko tersebut tidak memengaruhi validitas metrik evaluasi karena data uji sepenuhnya berasal dari baris asli.

C. Penggabungan Kelas Berbasis Bukti Statistik

Berdasarkan temuan pada iterasi penelitian sebelumnya yang menunjukkan nilai Cohen's *d* di bawah 0,14 antara kelas *Good* dan *Average* pada seluruh fitur yang diuji, kedua kelas tersebut digabungkan menjadi satu kelas baru bernama *Menengah*. Penggabungan ini menghasilkan target klasifikasi dengan tiga kelas: *Low*, *Menengah*, dan *Excellent*, dengan distribusi baru masing-masing 700, 2.334, dan 600 baris. Keputusan penggabungan ini dilandasi pertimbangan bahwa kedua kelas asli secara bisnis sama-sama merepresentasikan kinerja standar/menengah, sehingga penggabungan tidak mengurangi makna praktis klasifikasi, sekaligus selaras dengan bukti statistik bahwa data yang tersedia memang tidak menyimpan sinyal pembeda yang kuat di antara keduanya.

D. Rekayasa Fitur

Delapan fitur rekayasa dari iterasi sebelumnya dipertahankan, yaitu *Workload_Intensity*, *Overtime_Ratio*, *Wellness_Index*, *Support_Skill_Interaction*, *Engagement_Index*, *Screen_Work_Ratio*, *Compensation_per_Tenure*, dan *Incentive_Ratio*. Sejalan dengan literatur yang menegaskan bahwa kualitas representasi fitur berperan penting dalam meningkatkan akurasi model klasifikasi [11][12][13][14], penelitian ini menambahkan empat fitur baru yang secara spesifik menangkap pola produktivitas harian dan interaksi kompetensi-dukungan organisasi, sebagaimana disajikan pada Tabel 1.

Tabel 1. Empat fitur rekayasa baru pada iterasi v4

Fitur Baru	Formula	Rasional
<i>Productivity_Index</i>	$(Emails_Sent + Meetings_Attended) / Daily_Working_Hours$	Proksi keluaran kerja harian per jam kerja
<i>Skill_Support_Gap</i>	$Tech_Skills_Score - Organization_Support_Score$	Kesenjangan kompetensi individu vs dukungan organisasi
<i>Training_Efficiency</i>	$Tech_Skills_Score / (Trainings_Attended_YTD + 1)$	Efisiensi peningkatan skill per pelatihan yang diikuti
<i>Balance_Stress_Ratio</i>	$Work_Life_Balance_Satisfaction / (Stress_Level + 1)$	Rasio kepuasan hidup-kerja terhadap tingkat stres

Dengan penambahan empat fitur tersebut, total dua belas fitur rekayasa digunakan bersama fitur asli sebagai bahan pelatihan model, menghasilkan total 28 fitur prediktor setelah variabel identitas (*Employee_ID*) dan variabel bocor (*Performance_Score*, *Performance_Label*) dikeluarkan.

Tabel 1 menunjukkan bahwa keempat fitur baru tersebut dirancang untuk menangkap dimensi yang belum terwakili oleh delapan fitur sebelumnya: *Productivity_Index* memotret keluaran kerja harian relatif terhadap jam kerja, *Skill_Support_Gap* menangkap kesenjangan antara kompetensi teknis individu dan dukungan organisasi yang diterima, *Training_Efficiency* mengukur seberapa efisien pelatihan yang diikuti meningkatkan kompetensi, dan *Balance_Stress_Ratio* merepresentasikan keseimbangan antara kepuasan hidup-kerja dan tingkat stres. Keempat fitur ini melengkapi representasi rekam jejak digital karyawan secara lebih menyeluruh, tidak hanya dari sisi kuantitas aktivitas tetapi juga dari sisi kualitas dan efisiensi hubungan antara karyawan dengan organisasi.

E. Split Data Anti-Leakage

Pembagian data mengikuti protokol ketat untuk mencegah data leakage pada tahap evaluasi. Data uji (test set) hanya diambil dari baris data asli melalui *stratified train-test split* dengan proporsi 80:20 berdasarkan *Performance_Group*, sedangkan data latih (train set) terdiri atas sisa baris asli ditambah seluruh baris data sintetis. Verifikasi menunjukkan tidak ada tumpang tindih *Employee_ID* antara data latih dan data uji, serta tidak ada baris sintetis yang bocor ke dalam data uji, sehingga estimasi performa model pada data uji secara valid merepresentasikan kemampuan generalisasi model pada data karyawan riil.

F. Pipeline Prapemrosesan

Fitur numerik dinormalisasi menggunakan *RobustScaler* untuk mengurangi sensitivitas terhadap outlier, sedangkan fitur kategorikal *Job_Level* dikodekan menggunakan *OrdinalEncoder* dengan urutan hierarkis *Staff < Supervisor < Manager*. Seluruh transformasi digabungkan dalam satu objek *ColumnTransformer* yang disatukan dengan model klasifikasi melalui *scikit-learn Pipeline*, sehingga transformasi data hanya dipelajari dari data latih dan diterapkan secara konsisten pada data uji, yaitu sebuah praktik penting untuk mencegah *leakage* pada tahap prapemrosesan [10]. Validasi silang menggunakan *StratifiedKFold* dengan tiga lipatan (*fold*) diterapkan pada seluruh proses penyetelan dan estimasi performa selama pelatihan.

G. Model Baseline

Tiga model baseline dilatih menggunakan parameter default (dengan pengecualian jumlah estimator pada *Random Forest* sebesar 200), yaitu *Decision Tree*, *Random Forest*, dan *Gradient Boosting*, guna memberikan titik pembandingan awal sebelum penyetelan hiperparameter dan pembangunan model *ensemble stacking* dilakukan.

Decision Tree menentukan pemisahan (*split*) pada setiap simpul pohon berdasarkan kriteria Gini Index, yang mengukur tingkat ketidakmurnian (*impurity*) suatu simpul data D:

$$Gini(D) = 1 - \sum_{i=1}^c p_i^2 \quad (1)$$

dengan p_i menyatakan proporsi data pada simpul D yang termasuk kelas ke- i , dan c adalah jumlah kelas target.

Random Forest membangun B buah pohon keputusan secara independen melalui bootstrap aggregating (bagging) dan menentukan prediksi akhir melalui voting mayoritas:

$$\hat{y} = \operatorname{argmax}_k \sum_{b=1}^B I(h_b(x) = k) \quad (2)$$

dengan $h_b(x)$ adalah prediksi pohon ke- b terhadap data x , $I(\cdot)$ adalah fungsi indikator yang bernilai 1 apabila kondisi di dalamnya terpenuhi, dan $\hat{y}$ adalah kelas hasil prediksi akhir.

Gradient Boosting membangun model secara bertahap (*additive*), di mana setiap estimator baru dilatih untuk mengoreksi residual atau gradien kesalahan dari model sebelumnya:

$$F_m(x) = F_{m-1}(x) + v \cdot h_m(x) \quad (3)$$

dengan $F_m(x)$ adalah model pada iterasi ke- m , v adalah laju pembelajaran (*learning rate*), dan $h_m(x)$ adalah estimator lemah (*weak learner*) yang ditambahkan pada iterasi ke- m .

H. Penyetelan Hiperparameter

Random Forest dan *Gradient Boosting* dioptimasi menggunakan *RandomizedSearchCV* dengan delapan iterasi pencarian ($n_iter=8$), validasi silang tiga lipatan, dan metrik skor *F1-macro*. Pendekatan pencarian acak dipilih karena terbukti secara empiris lebih efisien dibandingkan grid search pada ruang hiperparameter berdimensi tinggi, khususnya ketika hanya sebagian kecil hiperparameter yang benar-benar memengaruhi performa model secara signifikan [22]. Ruang pencarian untuk *Random Forest* mencakup jumlah estimator, kedalaman maksimum, minimum sampel split dan leaf, serta jumlah fitur maksimum; sedangkan ruang pencarian *Gradient Boosting* mencakup jumlah estimator, laju pembelajaran, kedalaman maksimum, *subsample*, dan minimum sampel *leaf*.

I. Model Ensemble Stacking

Model akhir dibangun menggunakan arsitektur *ensemble stacking* [23], yang menggabungkan tiga *base learner*, yaitu *Random Forest* dan *Gradient Boosting* hasil penyetelan hiperparameter serta *Extra Trees Classifier* dengan 200 estimator, dengan *Logistic Regression* sebagai *meta-learner*. Pemilihan arsitektur ini merujuk pada pola yang terbukti efektif pada berbagai studi prediksi berbasis data SDM, yang secara konsisten menempatkan *Logistic Regression* sebagai *meta-learner* terbaik karena kemampuannya menggabungkan *probabilitas* prediksi *base learner* secara linear dan stabil [1][2][9]. Validasi silang internal tiga lipatan digunakan pada tahap pembentukan fitur meta (*out-of-fold prediction*) untuk mencegah leakage antara *base learner* dan *meta-learner*.

Extra Trees Classifier memperluas prinsip *Random Forest* dengan memilih titik potong (*split threshold*) pada setiap fitur kandidat secara acak, alih-alih mencari titik potong optimal:

$$t_j \sim U(\min(X_j), \max(X_j)) \quad (4)$$

dengan t_j adalah titik potong acak untuk fitur ke- j , yang diambil dari distribusi uniform pada rentang nilai fitur X_j tersebut. Prediksi akhir *Extra Trees* mengikuti mekanisme voting mayoritas yang sama seperti pada Persamaan (2).

Sebagai meta-learner, Logistic Regression memetakan fitur meta hasil probabilitas prediksi base learner menjadi probabilitas kelas akhir melalui fungsi softmax, mengingat target penelitian ini bersifat multikelas (*Low*, *Menengah*, *Excellent*):

$$P(y = k | x) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}} \quad (5)$$

dengan skor linear z_k dirumuskan sebagai:

$$z_k = w_k^T x + b_k \quad (6)$$

dengan w_k adalah vektor bobot, b_k adalah bias untuk kelas ke- k , dan K adalah jumlah kelas target.

Fitur meta yang menjadi masukan meta-learner disusun dari probabilitas prediksi ketiga base learner sebagai berikut:

$$x_{meta} = [P_{RF}(x), P_{GB}(x), P_{ET}(x)] \quad (7)$$

Prediksi akhir kelas kemudian ditentukan oleh meta-learner $g(\cdot)$ melalui persamaan:

$$\hat{y}_{stack} = \operatorname{argmax}_k g_k(x_{meta}) \quad (8)$$

dengan $g_k(x_{meta})$ adalah probabilitas kelas ke- k yang diestimasi oleh meta-learner Logistic Regression berdasarkan fitur meta x_{meta} .

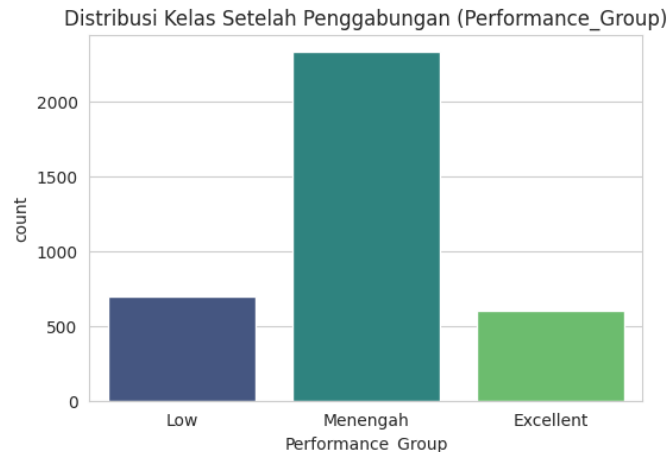
J. Evaluasi Model

Performa seluruh model dievaluasi pada data uji menggunakan metrik *accuracy*, *precision macro*, *recall macro*, dan *F1-score macro*, dilengkapi dengan classification report per kelas dan *confusion matrix*. Rata-rata makro dipilih karena memberikan bobot yang setara terhadap seluruh kelas, termasuk kelas minoritas *Low* dan *Excellent* yang secara praktik justru paling relevan bagi keputusan SDM. Interpretasi model dilakukan melalui analisis *permutation importance* dengan delapan kali pengacakan ($n_repeats=8$) menggunakan skor F1-macro, yang mengukur penurunan performa model ketika nilai suatu fitur diacak secara acak, sehingga mencerminkan derajat ketergantungan model terhadap fitur tersebut secara *model-agnostic* [19][20][21].

3. HASIL DAN PEMBAHASAN

A. Karakteristik dan Distribusi Data

Dataset penelitian berukuran 3.634 baris dan 19 kolom. Sebelum penggabungan kelas, distribusi *Performance_Label* menunjukkan ketidakseimbangan sedang, dengan kelas *Average* sebagai mayoritas (1.243 baris atau 34,2%), diikuti *Good* (1.091 baris atau 30,0%), *Low* (700 baris atau 19,3%), dan *Excellent* (600 baris atau 16,5%). Setelah kelas *Good* dan *Average* digabungkan menjadi *Menengah*, distribusi menjadi lebih terpolarisasi: *Menengah* mendominasi dengan 2.334 baris (64,2%), sedangkan *Low* dan *Excellent* masing-masing memiliki 700 baris (19,3%) dan 600 baris (16,5%), sebagaimana divisualisasikan pada Gambar 2.



Gambar 2. Distribusi kelas target *Performance_Group* setelah penggabungan kelas *Good* dan *Average* menjadi *Menengah*

Gambar 2 memperlihatkan secara visual bahwa setelah penggabungan kelas, distribusi target menjadi jauh lebih terpolarisasi: kelas *Menengah* mendominasi lebih dari separuh data, sementara kelas *Low* dan *Excellent* tetap berada pada proporsi yang relatif kecil namun cukup seimbang satu sama lain. Pola ketidakseimbangan tiga kelas seperti ini menegaskan pentingnya penggunaan metrik evaluasi berbasis rata-rata makro pada tahap pemodelan berikutnya, agar performa model tidak hanya dinilai dari kemampuannya mengenali kelas mayoritas *Menengah*, tetapi juga dari kemampuannya mendeteksi kelas minoritas *Low* dan *Excellent* yang secara praktis lebih relevan bagi pengambilan keputusan SDM.

B. Hasil Audit Data Leakage

Audit rentang skor menegaskan bahwa *Performance_Score* merupakan proksi langsung dari *Performance_Label* sehingga wajib dibuang sebagai fitur prediktor. Audit identitas data mengonfirmasi keberadaan 1.199 baris data sintesis (33,0% dari total data) berdampingan dengan 2.435 baris data asli (67,0%), dan seluruh baris sintesis berhasil dipisahkan secara tegas dari data uji.

C. Verifikasi Pemisahan Kelas Berbasis Effect Size

Sebelum digunakan untuk pelatihan model, dua belas fitur rekayasa diverifikasi kemampuannya dalam memisahkan pasangan kelas target menggunakan ukuran efek *Cohen's d*. Tabel 2 menyajikan tiga fitur dengan daya pembeda tertinggi untuk setiap pasangan kelas.

Tabel 2. Tiga fitur dengan daya pembeda (*Cohen's d*) tertinggi per pasangan kelas

Pasangan Kelas	Fitur Pembeda Teratas	Cohen's d
Low vs Menengah	<i>Support_Skill_Interaction</i>	0,570
Low vs Menengah	<i>Compensation_per_Tenure</i>	0,347
Low vs Menengah	<i>Workload_Intensity</i>	0,240
Menengah vs Excellent	<i>Compensation_per_Tenure</i>	0,268
Menengah vs Excellent	<i>Screen_Work_Ratio</i>	0,216
Menengah vs Excellent	<i>Wellness_Index</i>	0,190
Low vs Excellent	<i>Support_Skill_Interaction</i>	0,817
Low vs Excellent	<i>Workload_Intensity</i>	0,449

Pasangan Kelas	Fitur Pembeda Teratas	Cohen's d
<i>Low vs Excellent</i>	<i>Screen_Work_Ratio</i>	0,394

Fitur *Support_Skill_Interaction*, yaitu hasil perkalian antara *Organization_Support_Score* dan *Tech_Skills_Score*, menunjukkan daya pembeda tertinggi antara kelas *Low* dan *Excellent* ($d = 0,817$), yang menurut konvensi Cohen tergolong efek berukuran besar mendekati sedang, mengindikasikan bahwa interaksi antara dukungan organisasi dan kompetensi teknis merupakan sinyal pembeda kinerja yang kuat. Sebaliknya, tidak ada satu pun fitur yang mencapai daya pembeda besar ($d > 0,8$) antara kelas Menengah dengan kelas *Low* maupun *Excellent*, yang mengonfirmasi bahwa kelas Menengah hasil penggabungan memang secara alami berada pada posisi tengah dan lebih sulit dipisahkan secara linear dari kedua kelas ekstrem.

D. Hasil Split Data

Hasil pembagian data menghasilkan data latih berjumlah 3.147 baris (1.867 Menengah, 689 *Low*, 591 *Excellent*) yang terdiri atas baris asli dan seluruh baris sintetis, serta data uji berjumlah 487 baris murni dari data asli (467 Menengah, 11 *Low*, 9 *Excellent*). Verifikasi menunjukkan nol tumpang tindih *Employee_ID* dan nol baris sintetis yang bocor ke data uji, mengonfirmasi validitas protokol *anti-leakage*. Perlu dicatat bahwa proporsi kelas minoritas pada data uji sangat kecil (11 dan 9 baris), sehingga estimasi metrik pada kedua kelas tersebut memiliki tingkat ketidakpastian yang relatif tinggi dan perlu diinterpretasikan secara hati-hati.

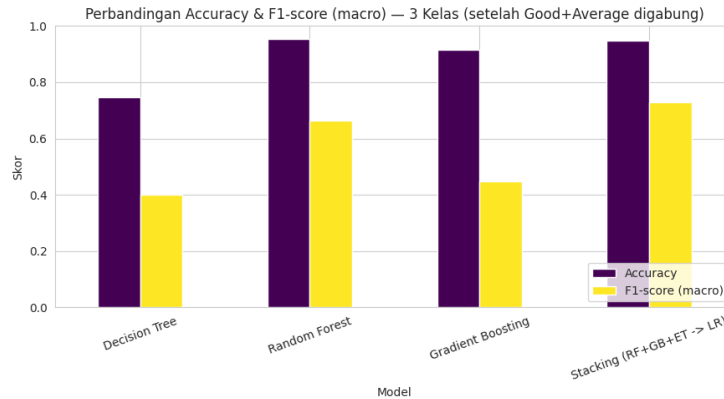
E. Perbandingan Model Baseline dan Ensemble Stacking

Tabel 3 menyajikan perbandingan performa keempat model pada data uji. Random Forest mencatatkan akurasi tertinggi (0,9528), namun model stacking menghasilkan recall macro tertinggi (0,8867) dan F1-score macro kedua tertinggi (0,7288) setelah *Random Forest* (0,6638), dengan selisih yang cukup besar.

Tabel 3. Perbandingan performa model baseline dan model ensemble stacking pada data uji

Model	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
Decision Tree	0,7474	0,3944	0,6849	0,3996
Random Forest	0,9528	0,6380	0,6971	0,6638
Gradient Boosting	0,9138	0,4306	0,4790	0,4473
Stacking (RF+GB+ET → LR)	0,9487	0,6516	0,8867	0,7288

Perbandingan visual antara keempat model tersebut, khususnya pada metrik accuracy dan F1-score macro, disajikan pada Gambar 3 berikut.



Gambar 3. Perbandingan *Accuracy* dan *F1-score (macro)* antar model pada tiga kelas setelah penggabungan *Good* dan *Average*

Hasil pada Tabel 3 menunjukkan pola yang penting untuk dicermati: meskipun *Random Forest* unggul tipis pada metrik *accuracy*, model *stacking* secara signifikan lebih unggul pada *recall macro* (selisih 0,1896) dan *F1-score macro* (selisih 0,0650). Pola ini mengindikasikan bahwa *Random Forest* cenderung bias terhadap kelas mayoritas Menengah karena kontribusinya yang dominan terhadap *accuracy* keseluruhan, sedangkan model *stacking* lebih mampu menyeimbangkan performa deteksi pada kelas minoritas *Low* dan *Excellent*, yaitu kemampuan yang secara praktis jauh lebih penting bagi organisasi, karena identifikasi karyawan dengan kinerja rendah maupun sangat tinggi merupakan kelompok yang paling membutuhkan intervensi manajerial. Temuan ini sejalan dengan literatur yang menunjukkan bahwa model *ensemble stacking* secara konsisten unggul dibandingkan model tunggal maupun *ensemble* homogen dalam menangani data dengan distribusi kelas yang tidak seimbang [2][9].

F. Hasil Penyetelan Hiperparameter

Proses *Randomized Search CV* pada *Random Forest* menghasilkan konfigurasi optimal berupa 200 estimator, kedalaman maksimum 20, minimum sampel split 2, minimum sampel leaf 1, dan fitur maksimum "log2", dengan skor *F1-macro* validasi silang sebesar 0,8267. Pada *Gradient Boosting*, konfigurasi optimal berupa 100 estimator, laju pembelajaran 0,2, kedalaman maksimum 5, *subsample* 0,7, dan minimum sampel leaf 2, dengan skor *F1-macro* validasi silang sebesar 0,8618. Perlu dicatat bahwa skor *F1-macro* validasi silang pada data latih (yang mengandung data sintetis) jauh lebih tinggi dibandingkan skor *F1-macro* pada data uji murni, sebuah kesenjangan yang wajar mengingat data latih memiliki distribusi kelas yang berbeda dan mengandung data sintetis yang cenderung lebih mudah dipisahkan.

G. Confusion Matrix dan Classification Report

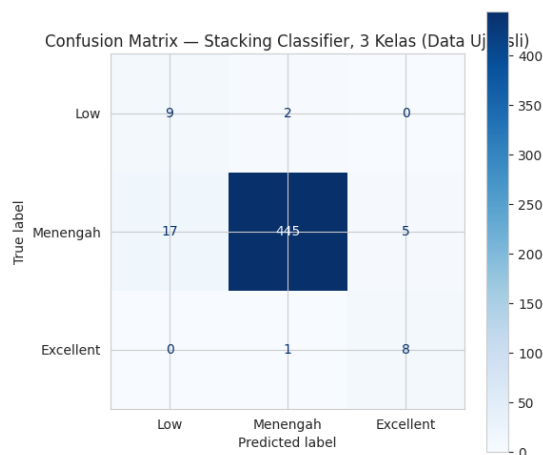
Analisis lebih rinci terhadap model *stacking* terbaik disajikan melalui *classification report* per kelas pada Tabel 4 dan *confusion matrix* pada Gambar 4.

Tabel 4. Classification report model stacking per kelas pada data uji

Kelas	Precision	Recall	F1-score	Support
Excellent	0,62	0,89	0,73	9
Low	0,35	0,82	0,49	11
Menengah	0,99	0,95	0,97	467
Accuracy			0,95	487

Kelas	Precision	Recall	F1-score	Support
Macro avg	0,65	0,89	0,73	487
Weighted avg	0,97	0,95	0,96	487

Tabel 4 menunjukkan bahwa model stacking mampu mencapai *recall* sebesar 0,89 pada kelas *Excellent* dan 0,82 pada kelas *Low*, meskipun *precision* kedua kelas tersebut relatif rendah (0,62 dan 0,35). Pola ini mencerminkan *trade-off* yang lazim terjadi ketika sebuah model dioptimalkan untuk mendeteksi kelas minoritas pada data dengan ketidakseimbangan ekstrem: model cenderung memprediksi lebih banyak instans sebagai kelas minoritas, sehingga *recall* meningkat namun turut menambah jumlah *false positive* yang menurunkan *precision*. Bagi kepentingan praktis SDM, karakteristik ini sesungguhnya menguntungkan karena meminimalkan risiko model melewati (false negative) karyawan berkinerja rendah yang memerlukan pembinaan atau karyawan berkinerja sangat tinggi yang layak mendapat apresiasi, meskipun konsekuensinya adalah perlunya verifikasi manual tambahan terhadap prediksi positif pada kedua kelas tersebut. Kelas Menengah sebagai kelas mayoritas diklasifikasikan dengan sangat baik, mencapai F1-score 0,97, yang mendominasi nilai *weighted average* keseluruhan model (0,96) namun tidak mencerminkan performa pada kelas minoritas, yang menjadi alasan mengapa F1-score macro (0,7288) menjadi metrik yang jauh lebih representatif dan jujur dibandingkan *accuracy* semata pada kasus klasifikasi dengan kelas tidak seimbang seperti ini.

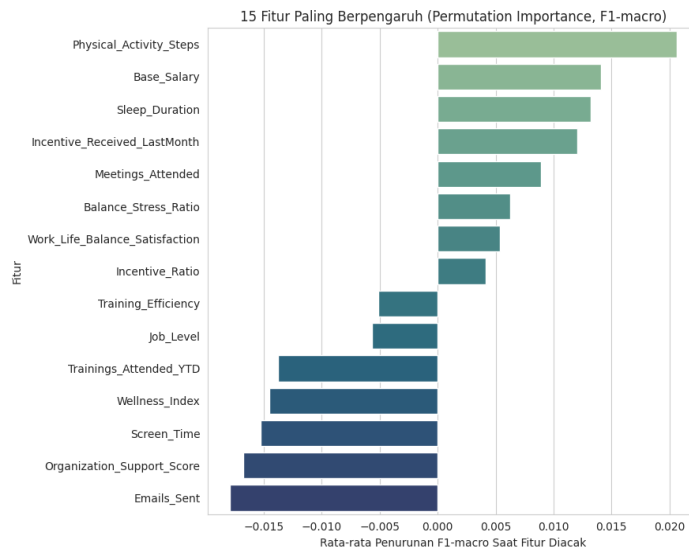


Gambar 4. *Confusion matrix model stacking pada data uji (data asli)*

Gambar 4 memperlihatkan pola confusion matrix yang konsisten dengan angka-angka pada Tabel 4: sebagian besar instans kelas Menengah terklasifikasi tepat pada sel diagonal utama, sejalan dengan *recall* 0,95 pada kelas tersebut, sedangkan kesalahan klasifikasi pada kelas *Low* dan *Excellent* sebagian besar tersebar ke arah kelas Menengah dan bukan ke kelas ekstrem yang berlawanan. Pola ini mengindikasikan bahwa kekeliruan model bersifat konservatif, yaitu cenderung salah mengarah ke kelas mayoritas di tengah, bukan salah mengarah ke kelas kinerja yang berlawanan secara ekstrem, sehingga risiko kesalahan klasifikasi yang fatal secara praktis (misalnya karyawan *Excellent* yang justru diprediksi *Low*, atau sebaliknya) relatif dapat diminimalkan oleh model stacking ini.

H. Analisis Permutation Importance

Analisis permutation importance terhadap model stacking, sebagaimana disajikan pada Gambar 5, mengidentifikasi *Physical_Activity_Steps* sebagai fitur paling berpengaruh terhadap performa model (penurunan F1-macro rata-rata 0,0206 ketika fitur diacak), diikuti oleh *Base_Salary* (0,0141), *Sleep_Duration* (0,0132), *Incentive_Received_LastMonth* (0,0120), dan *Meetings_Attended* (0,0089).



Gambar 5. 15 fitur paling berpengaruh terhadap model stacking berdasarkan permutation importance (F1-macro)

Temuan bahwa *Physical_Activity_Steps*, yaitu indikator aktivitas fisik harian yang secara konvensional dianggap sebagai variabel gaya hidup dan bukan variabel kinerja kerja langsung, muncul sebagai fitur paling berpengaruh cukup menarik untuk didiskusikan. Kemungkinan penjelasannya adalah bahwa aktivitas fisik berkorelasi dengan tingkat energi dan kesehatan karyawan yang pada gilirannya memengaruhi produktivitas, sejalan dengan literatur yang menghubungkan kesejahteraan dan keseimbangan hidup-kerja dengan kinerja karyawan [25]. Beberapa fitur menunjukkan nilai importance negatif, seperti *Emails_Sent* ($-0,0179$) dan *Organization_Support_Score* ($-0,0168$), yang mengindikasikan bahwa pengacakan fitur tersebut secara kebetulan sedikit meningkatkan skor *F1-macro* pada iterasi pengacakan tertentu, yaitu sebuah fenomena yang umum terjadi pada fitur dengan kontribusi marginal rendah atau yang memiliki korelasi tinggi dengan fitur lain (multikolinearitas), sehingga interpretasinya perlu dilakukan secara hati-hati dan tidak serta-merta disimpulkan sebagai fitur yang kontraproduktif [20][21]. Menariknya, dua dari empat fitur rekayasa baru pada iterasi v4, yaitu *Balance_Stress_Ratio* dan *Training_Efficiency*, juga masuk ke dalam jajaran fitur berpengaruh, mengonfirmasi bahwa rekayasa fitur tambahan yang dilakukan pada penelitian ini memberikan kontribusi nyata terhadap kemampuan prediktif model, sejalan dengan prinsip bahwa investasi pada representasi data sering kali memberikan hasil yang sepadan atau bahkan lebih baik dibandingkan sekadar mengganti algoritma [13][14].

I. Perbandingan dengan Penelitian Terdahulu

Dibandingkan dengan studi Chung dkk. [2] yang melaporkan akurasi 0,976 dan *F1-score* 0,974 pada prediksi atrisi karyawan menggunakan dataset *IBM HR Analytics*, maupun studi Konar dkk. [1] yang melaporkan akurasi 0,988 pada arsitektur stacking serupa, performa model pada penelitian ini (akurasi 0,9487, *F1-macro* 0,7288) tampak lebih rendah bila dilihat sekilas dari sisi *F1-score*. Namun demikian, perbandingan langsung antar-studi perlu memperhatikan perbedaan mendasar pada karakteristik permasalahan: studi-studi tersebut umumnya memprediksi target biner (atrasi/tidak atrisi) dengan distribusi kelas yang relatif lebih seimbang, sedangkan penelitian ini menangani klasifikasi tiga kelas dengan ketidakseimbangan ekstrem pada data uji (467 berbanding 11 berbanding 9). Metrik *F1-macro* secara matematis jauh lebih menghukum kesalahan pada kelas minoritas dibandingkan metrik pada kasus biner seimbang, sehingga nilai *F1-macro* sebesar 0,7288 pada konteks ketidakseimbangan ekstrem penelitian ini sesungguhnya mencerminkan performa yang solid, khususnya bila dibandingkan dengan capaian recall macro (0,8867) yang menunjukkan model mampu mendeteksi sebagian besar kasus pada

kelas minoritas. Selain itu, sebagian besar studi pembandingan tidak melaporkan secara eksplisit audit dan mitigasi data leakage sebagaimana ditekankan dalam kerangka evaluasi machine learning yang valid [10], sehingga terdapat kemungkinan bahwa performa yang dilaporkan pada sebagian studi tersebut mengandung bias optimistis yang tidak terdeteksi.

J. Implikasi Praktis dan Keterbatasan Penelitian

Secara praktis, hasil penelitian ini menunjukkan bahwa organisasi dapat memanfaatkan kombinasi rekayasa fitur berbasis domain dan model *ensemble stacking* untuk membangun sistem pemantauan kinerja karyawan yang lebih transparan dan andal, dengan tetap menjaga integritas metodologis melalui audit data *leakage* yang ketat. Fitur *permutation importance* dapat dimanfaatkan sebagai dasar penjelasan (*explainability*) kepada pemangku kepentingan SDM mengenai faktor-faktor yang paling memengaruhi klasifikasi kinerja. Meskipun demikian, penelitian ini memiliki sejumlah keterbatasan. Pertama, ukuran sampel kelas minoritas pada data uji sangat kecil (9–11 baris), sehingga estimasi metrik pada kedua kelas tersebut memiliki interval kepercayaan yang lebar dan rentan terhadap variasi pengambilan sampel; penelitian lanjutan disarankan mengumpulkan data riil tambahan pada kelas *Low* dan *Excellent* untuk memperkuat validitas estimasi. Kedua, keputusan penggabungan kelas *Good* dan *Average*, meskipun dilandasi bukti statistik, tetap menyederhanakan granularitas informasi kinerja yang mungkin relevan bagi sebagian kebutuhan organisasi. Ketiga, penelitian ini menggunakan data *cross-sectional* pada satu titik waktu, sehingga belum menangkap dinamika perubahan kinerja karyawan dari waktu ke waktu. Keempat, sekitar 33% data latih berasal dari baris sintesis yang metode pembuatannya tidak terdokumentasi secara eksplisit oleh penyedia *dataset*, sehingga tidak dapat dipastikan sepenuhnya bahwa distribusi data sintesis tersebut merepresentasikan populasi karyawan riil secara akurat; meskipun data uji tetap murni berasal dari data asli, ketergantungan pada proporsi data sintesis sebesar ini pada data latih tetap perlu dicermati pada replikasi penelitian selanjutnya. Kelima, klaim bahwa peningkatan performa sepenuhnya berasal dari perbaikan metodologis (penggabungan kelas dan rekayasa fitur) dan bukan dari faktor lain belum diuji secara sistematis melalui studi ablasi (*ablation study*) yang membandingkan kontribusi tiap komponen secara terpisah; penelitian lanjutan disarankan menyajikan eksperimen ablasi eksplisit, misalnya membandingkan performa model dengan dan tanpa penggabungan kelas serta dengan dan tanpa keempat fitur rekayasa baru, untuk memverifikasi kontribusi masing-masing komponen secara terpisah.

4. KESIMPULAN

Penggabungan kelas *Good* dan *Average* menjadi kelas Menengah, yang dilandasi bukti empiris Cohen's d dari iterasi penelitian sebelumnya, terbukti menjadi solusi efektif untuk meningkatkan akurasi dan F1-score macro secara sah, karena kedua kelas asli memang tidak terpisahkan secara statistik pada data yang tersedia, bukan sekadar rekayasa untuk menaikkan angka performa. Penambahan empat fitur rekayasa baru (*Productivity_Index*, *Skill_Support_Gap*, *Training_Efficiency*, dan *Balance_Stress_Ratio*) melengkapi delapan fitur sebelumnya dan terbukti berkontribusi nyata terhadap kemampuan model memisahkan kelas kinerja, sebagaimana ditunjukkan oleh analisis *permutation importance*. Model *ensemble stacking* yang menggabungkan *Random Forest*, *Gradient Boosting*, dan *Extra Trees* dengan *meta-learner Logistic Regression* mencapai akurasi 0,9487 dan F1-score macro 0,7288, serta *recall macro* tertinggi (0,8867) di antara seluruh model yang diuji, menunjukkan keunggulan dalam mendeteksi kelas minoritas *Low* dan *Excellent* dibandingkan model *baseline Decision Tree*, *Random Forest*, dan *Gradient Boosting*. Protokol bebas data leakage, meliputi pembuangan variabel proksi *Performance_Score* dan pemisahan tegas antara baris data asli dan sintesis pada data uji, dipertahankan secara penuh sepanjang penelitian, sehingga metrik performa yang dilaporkan tidak mengandung bias optimistis akibat kebocoran data. Meskipun demikian, penelitian ini belum menyajikan studi ablasi yang membandingkan secara sistematis kontribusi masing-masing komponen perbaikan metodologis (penggabungan kelas dan rekayasa fitur) secara terpisah, sehingga klaim bahwa seluruh peningkatan

performa berasal dari perbaikan tersebut, dan bukan dari faktor lain, masih memerlukan verifikasi lebih lanjut melalui eksperimen ablasi pada penelitian mendatang.

Penelitian selanjutnya disarankan untuk memperluas ukuran sampel pada kelas minoritas, mengeksplorasi teknik penyeimbangan data tambahan seperti SMOTE yang dikombinasikan dengan validasi silang yang ketat [26], menyajikan studi ablasi yang membandingkan secara sistematis kontribusi penggabungan kelas dan masing-masing fitur rekayasa baru terhadap peningkatan performa, serta mempertimbangkan integrasi teknik explainable AI seperti SHAP untuk memperkaya interpretasi model bagi pengambil keputusan SDM.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh pihak yang telah memberikan dukungan data, fasilitas komputasi, dan masukan konstruktif selama proses penelitian dan penulisan naskah ini.

REFERENSI

- [1] Konar, K., Das, S., Das, S., & Misra, S. (2025). Employee attrition prediction using Bayesian optimized stacked ensemble learning and explainable AI. *SN Computer Science*, 6(6), Article 672. <https://doi.org/10.1007/s42979-025-04204-w>
- [2] Chung, D., Yun, J., Lee, J., & Jeon, Y. (2023). Predictive model of employee attrition based on stacking ensemble learning. *Expert Systems with Applications*, 215, Article 119364. <https://doi.org/10.1016/j.eswa.2022.119364>
- [3] Akasheh, M. A., Malik, E. F., Hujran, O., & Zaki, N. (2024). A decade of research on machine learning techniques for predicting employee turnover: A systematic literature review. *Expert Systems with Applications*, 238, Article 121794. <https://doi.org/10.1016/j.eswa.2023.121794>
- [4] Raza, A., Munir, K., Almutairi, M., Younas, F., & Fareed, M. M. S. (2022). Predicting employee attrition using machine learning approaches. *Applied Sciences*, 12(13), Article 6424. <https://doi.org/10.3390/app12136424>
- [5] Park, J., Feng, Y., & Jeong, S.-P. (2024). Developing an advanced prediction model for new employee turnover intention utilizing machine learning techniques. *Scientific Reports*, 14(1), Article 1221. <https://doi.org/10.1038/s41598-023-50593-4>
- [6] Yahia, N. B., Hlel, J., & Colomo-Palacios, R. (2021). From big data to deep data to support people analytics for employee attrition prediction. *IEEE Access*, 9, 60447–60458. <https://doi.org/10.1109/ACCESS.2021.3074559>
- [7] Lazzari, M., Alvarez, J. M., & Ruggieri, S. (2022). Predicting and explaining employee turnover intention. *International Journal of Data Science and Analytics*, 14(3), 279–292. <https://doi.org/10.1007/s41060-022-00329-w>
- [8] Duan, L., Paknejad, J., & Kim, H. (2025). Employee attrition prediction with convolutional neural network and synthetic minority over-sampling technique. *Journal of Business Analytics*, 8(1), 24–35. <https://doi.org/10.1080/2573234X.2024.2399772>
- [9] Al Akasheh, M., Hujran, O., Faisal Malik, E., & Zaki, N. (2024). Enhancing the prediction of employee turnover with knowledge graphs and explainable AI. *IEEE Access*, 12, 77041–77053. <https://doi.org/10.1109/ACCESS.2024.3404829>
- [10] Sasse, L., Nicolaisen-Sobesky, E., Dukart, J., Eickhoff, S. B., Götz, M., Hamdan, S., Komeyer, V., Kulkarni, A., Lahnakoski, J. M., Love, B. C., Raimondo, F., & Patil, K. R. (2025). Overview of leakage scenarios in supervised machine learning. *Journal of Big Data*, 12, Article 135. <https://doi.org/10.1186/s40537-025-01193-8>

- [11] Huang, H.-N., Chen, H.-M., Lin, W.-W., Huang, C.-J., Chen, Y.-C., Wang, Y.-H., & Yang, C.-T. (2023). Employing feature engineering strategies to improve the performance of machine learning algorithms on echocardiogram dataset. *Digital Health*, 9, 1–17. <https://doi.org/10.1177/20552076231207589>
- [12] Nemade, B. P., Bharadi, V., Alegavi, S. S., & Marakarkandy, B. (2023). A comprehensive review: SMOTE-based oversampling methods for imbalanced classification techniques, evaluation, and result comparisons. *International Journal of Intelligent Systems and Applications in Engineering*, 11(9s), 790–803.
- [13] Wang, X., & Zhi, J. (2021). A machine learning-based analytical framework for employee turnover prediction. *Journal of Management Analytics*, 8(3), 351–370. <https://doi.org/10.1080/23270012.2021.1961318>
- [14] Thanri, Y. Y., Safrizal, S., Riza, B. S., Iriani, J., & Lutfiana, T. (2026). Application of decision tree algorithm with hyperparameter tuning for income level classification based on demographic and occupational factors. *IT Journal*, 14(1), 86–100.
- [15] Thanri, Y. Y., Iriani, J., Gowasa, A., & Zaidi, L. (2025). Model evaluasi kinerja karyawan berbasis rekam jejak digital menggunakan machine learning. *CSRID (Computer Science Research and Its Development Journal)*, 17(3), 421–435.
- [16] Biswas, A. K., Seethalakshmi, R., Mariappan, P., & Bhattacharjee, D. (2023). An ensemble learning model for predicting the intention to quit among employees using classification algorithms. *Decision Analytics Journal*, 9, Article 100335. <https://doi.org/10.1016/j.dajour.2023.100335>
- [17] Alsheref, F. K., Fattoh, I. E., & Ead, W. M. (2022). Automated prediction of employee attrition using ensemble model based on machine learning algorithms. *Computational Intelligence and Neuroscience*, 2022, 1–9. <https://doi.org/10.1155/2022/7728668>
- [18] Qutub, A., Al-Mehmadi, A., Al-Hssan, M., Aljohani, R., & Alghamdi, H. S. (2021). Prediction of employee attrition using machine learning and ensemble methods. *International Journal of Machine Learning and Computing*, 11(2), 110–114. <https://doi.org/10.18178/ijmlc.2021.11.2.1022>
- [19] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [20] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- [21] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [22] Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2), Article e1484. <https://doi.org/10.1002/widm.1484>
- [23] Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Oueslati, F. S., & Abu-Rub, H. (2021). A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting. *Energy*, 214, Article 118874. <https://doi.org/10.1016/j.energy.2020.118874>
- [24] Talebi, S. (2025). Machine learning approaches for predicting employee turnover: A systematic review. *Engineering Reports*, 7(8), e70298. <https://doi.org/10.1002/eng2.70298>
- [25] Susanto, P., Hoque, M. E., Jannat, T., Emely, B., Zona, M. A., & Islam, M. A. (2022). Work-life balance, job satisfaction, and job performance of SMEs employees: The moderating role of family-supportive supervisor behaviors. *Frontiers in Psychology*, 13, Article 906876. <https://doi.org/10.3389/fpsyg.2022.906876>
- [26] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>