Klasifikasi Tingkat *Engagement* Pemain Game Online menggunakan Algoritma *Random Forest*

Classification of Online Game Player Engagement Levels using the Random Forest Algorithm

[1]Febrian Joseph Laia*, [2]Charitas Fibriani
[1,2]Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
[1,2]Jl. Dr. O. Notohamidjojo No. 1-10, Blotongan, Sidorejo, Salatiga, Jawa Tengah 501715 Indonesia
*e-mail: charitas.fibriani@uksw.edu

(***received*:** 2 June 2026, ***revised*:** 21 July 2026, ***accepted*:** 22 July 2026)

Abstrak

Peningkatan jumlah pemain game *online* menghasilkan akumulasi data perilaku yang dapat dimanfaatkan untuk memetakan tingkat keterlibatan (*engagement*) pemain. Permasalahan utama yang dihadapi adalah ketidakseimbangan kelas pada data *engagement* serta minimnya analisis faktor penentu di balik prediksi model. Penelitian ini bertujuan untuk mengembangkan model klasifikasi tingkat *engagement* pemain game online menggunakan algoritma *Random Forest* dengan penanganan data tidak seimbang melalui *Synthetic Minority Over-sampling Technique (SMOTE)*, sekaligus mengidentifikasi fitur-fitur yang paling berkontribusi melalui *Feature Importance*. Dataset yang digunakan adalah *Online Gaming Behavior Dataset* dari *Kaggle* sebanyak 40.034 baris. Model *Random Forest* dioptimasi dengan *RandomizedSearchCV* dan validasi silang 5-*fold* untuk memperoleh konfigurasi hyperparameter terbaik. Hasil pengujian menunjukkan akurasi model mencapai 92%, dengan *recall* untuk kelas *Low*, *Medium*, dan *High* masing-masing sebesar 90%, 94%, dan 89%. Analisis *Feature Importance* dengan pendekatan *impurity-based* dan *permutation* secara konsisten mengidentifikasi *SessionsPerWeek* (0,4375; 0,4309) dan *AvgSessionDurationMinutes* (0,3371; 0,3540) sebagai dua fitur paling dominan yang menjelaskan lebih dari 77% keputusan model, sedangkan fitur demografis memberikan pengaruh minimal. Kebaruan penelitian ini terletak pada integrasi *Random Forest*, *SMOTE*, dan *Feature Importance* sebagai upaya menghasilkan model yang akurat dan juga interpretatif.. Temuan ini diharapkan mampu memberikan dasar strategis bagi pengembang dalam merancang program retensi pemain, seperti pemberian *daily login rewards* untuk meningkatkan frekuensi sesi dan *time-limited events* untuk memperpanjang durasi bermain.

Kata kunci: engagement pemain, game online, random forest, SMOTE, feature importance.

Abstract

The rapid growth in the number of online game players has generated large volumes of behavioral data that can be leveraged to analyze player engagement levels. However, the primary challenges include class imbalance in engagement data and the limited interpretability of predictive models regarding the factors influencing their decisions. This study aims to develop a classification model for online game player engagement levels using the Random Forest algorithm, while addressing class imbalance through the Synthetic Minority Over-sampling Technique (SMOTE) and identifying the most influential features using Feature Importance analysis. The study utilized the Online Gaming Behavior Dataset from Kaggle, comprising 40,034 records. The Random Forest model was optimized using RandomizedSearchCV with five-fold cross-validation to determine the optimal hyperparameter configuration. The experimental results demonstrate that the proposed model achieved an overall accuracy of 92%, with recall values of 90%, 94%, and 89% for the Low, Medium, and High engagement classes, respectively. Feature Importance analysis using both impurity-based and permutation approaches consistently identified SessionsPerWeek (0.4375 and 0.4309) and AvgSessionDurationMinutes (0.3371 and 0.3540) as the two most influential features, jointly accounting for more than 77% of the model's predictive decisions, whereas demographic features contributed only marginally. The novelty of this study lies in the integration of Random Forest,

 ISSN:2302-8149
e-ISSN:2540-9719

SMOTE, and Feature Importance to develop a classification model that is not only highly accurate but also interpretable. These findings provide valuable insights for game developers in designing evidence-based player retention strategies, such as implementing daily login rewards to increase session frequency and time-limited events to encourage longer gameplay sessions.

Keywords: *player engagement, online game, random forest, SMOTE, feature importance.*

1 Pendahuluan

Industri game *online* telah berkembang pesat menjadi salah satu sektor unggulan dalam ekonomi kreatif, dengan peran yang semakin krusial dalam mendukung pertumbuhan ekonomi digital. Nilai pasar di Indonesia sendiri dalam bidang industri game tercatat mencapai USD 1,92 miliar, dan menjadi penyumbang utama ekonomi digital nasional [1]. Secara global, pengguna video game telah melampaui 3,38 miliar orang [2]. Pertumbuhan ini menghasilkan volume data pemain yang sangat besar. Data seperti durasi bermain, frekuensi sesi, dan pencapaian pemain kini dimanfaatkan untuk mengklasifikasikan tingkat keterikatan (*engagement*) ke dalam kategori *Low*, *Medium*, dan *High* guna merancang strategi retensi yang tepat sasaran [3].

Pemain dengan *engagement* tinggi menunjukkan loyalitas dan potensi monetisasi besar, sedangkan pemain *engagement* rendah berisiko tinggi berhenti bermain (*churn*). *Engagement* terbukti menjadi mediator utama antara pengalaman bermain dan niat pemain untuk terus bermain, membeli item virtual, serta merekrut pemain baru [4]. Prediksi churn menggunakan machine learning juga ditegaskan sebagai komponen vital dalam strategi retensi industri game modern [5]. Kemampuan mengklasifikasikan pemain berdasarkan engagement semakin krusial seiring meningkatnya kompleksitas industri game. Qiu et al. [6] menunjukkan bahwa pemain dapat dikelompokkan ke dalam lima segmen berbeda berdasarkan pola perilaku, yang setiap segmennya memerlukan pendekatan retensi yang berbeda [7].

Beberapa penelitian telah memanfaatkan *Online Gaming Behavior* Dataset untuk mengklasifikasikan *engagement* pemain, namun masih berfokus pada pencapaian akurasi tertinggi [3], [8], [9]. Analisis *Feature Importance* yang menyediakan ukuran kuantitatif tentang kontribusi setiap fitur terhadap keputusan model masih jarang dilakukan. Padahal, informasi inilah yang dapat menerjemahkan label klasifikasi menjadi strategi retensi yang tepat sasaran. Kebaruan penelitian ini terletak pada integrasi *Random Forest* dengan *SMOTE* untuk menghasilkan model yang akurat sekaligus interpretatif melalui *Feature Importance*. Efektivitas *SMOTE* akan dievaluasi melalui perbandingan performa model dengan dan tanpa teknik *oversampling*.

Rumusan masalah penelitian ini ada tiga. Pertama, bagaimana membangun model *Random Forest* yang mampu mengelompokkan pemain ke dalam kategori *Low*, *Medium*, dan *High engagement* secara akurat. Kedua, seberapa besar peningkatan *recall* pada kelas minoritas yang dihasilkan oleh *SMOTE*. Ketiga, fitur perilaku apa yang paling dominan menentukan engagement berdasarkan *Feature Importance*. Penelitian ini bertujuan mengembangkan model klasifikasi *engagement* pemain yang akurat dan interpretatif, serta memberikan rekomendasi strategis bagi pengembang game.

2 Tinjauan Literatur

Klasifikasi berbasis data perilaku merupakan pendekatan *machine learning* untuk mengelompokkan pemain berdasarkan pola interaksi dengan game. Data seperti frekuensi sesi, durasi bermain, dan pencapaian pemain terbukti efektif untuk mengidentifikasi tingkat *engagement* secara otomatis, serta mendukung strategi retensi pengembang [3], [8].

Distribusi kelas yang tidak merata merupakan masalah umum pada data *engagement* pemain. Model cenderung mempelajari pola mayoritas dan mengabaikan minoritas, menghasilkan akurasi tinggi namun *recall* kelas minoritas rendah. *SMOTE* mengatasi ini dengan menghasilkan sampel sintetis baru melalui interpolasi, efektif meningkatkan recall tanpa mengubah distribusi data uji [9], [10].

Random Forest merupakan algoritma *ensemble* berbasis bagging yang memberikan performa konsisten pada berbagai tugas klasifikasi, termasuk pada data tidak seimbang. Durachman dan Rahman [11] menunjukkan bahwa *Random Forest* mengungguli *Logistic Regression* dalam

 ISSN:2302-8149
e-ISSN:2540-9719

memprediksi rekomendasi pemain. Model ini tetap memiliki bias terhadap kelas mayoritas ketika distribusi sangat timpang, karena jumlah pohon keputusan pada kelas minoritas menjadi jauh lebih sedikit [3], [8], [10].

Keunggulan *Random Forest* yang belum banyak dimanfaatkan adalah kemampuannya menyediakan *Feature Importance*, yang memungkinkan identifikasi faktor-faktor paling menentukan *engagement* [10]. Interpretabilitas model telah menjadi perhatian utama dalam *machine learning* modern. Model tanpa kemampuan menjelaskan keputusannya dianggap kurang transparan untuk pengambilan keputusan strategis [12].

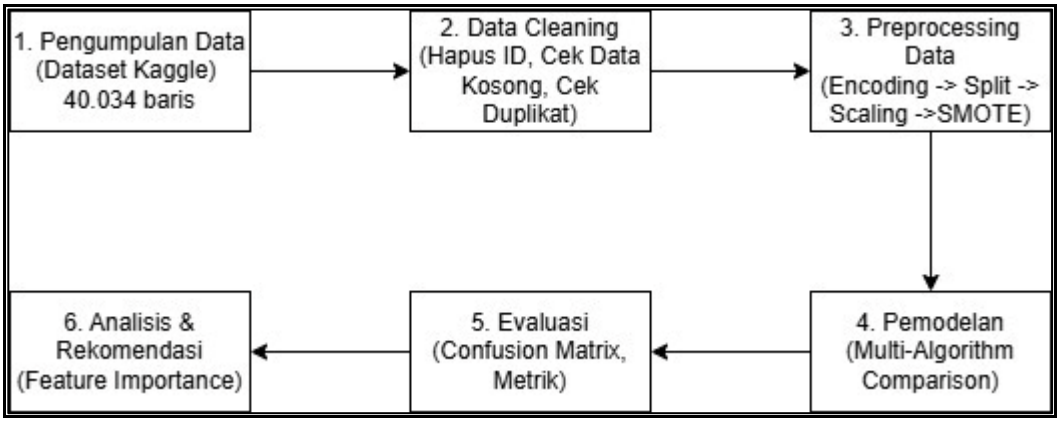
Feature Importance pada *Random Forest* dihitung berdasarkan rata-rata penurunan *impurity* di seluruh pohon [13]. Alzuhdi et al. [10] membuktikan analisis ini mampu mengidentifikasi atribut pemain paling menentukan pada game edukasi. Wawasan dari *Feature Importance* memungkinkan pengembang memfokuskan strategi retensi pada aspek perilaku yang terbukti paling memengaruhi *engagement*.

Berbagai teknik *balancing* telah diterapkan untuk meningkatkan sensitivitas model terhadap kelas minoritas, efektivitasnya pada data *engagement* pemain masih memerlukan kajian lebih mendalam, terutama ketika dikombinasikan dengan *Feature Importance* [9]. Studi sebelumnya berfokus pada peningkatan akurasi, sementara analisis faktor pendorong keputusan model masih jarang dilakukan [8], [10], [14].

Kondisi ini menyisakan ruang penelitian untuk mengevaluasi performa *Random Forest* dengan *SMOTE* pada dataset *engagement* pemain, penelitian ini menempatkan diri pada posisi untuk mengevaluasi performa *Random Forest* yang diintegrasikan dengan *SMOTE*, serta mengidentifikasi faktor-faktor dominan yang membedakan setiap kategori *engagement* melalui analisis *Feature Importance* berbasis *impurity* dan *permutation*. Pendekatan ini diharapkan mampu menghasilkan model yang akurat serta interpretatif, sehingga memberikan dasar strategis yang lebih kuat bagi pengembang game.

3 Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental untuk membangun model klasifikasi berbasis *Random Forest* yang mampu mengelompokkan pemain ke dalam tiga tingkat engagement. Pemilihan *Random Forest* didasarkan pada kemampuannya yang unggul dalam menangani data campuran (numerik dan kategorikal), ketahanannya terhadap *overfitting* melalui mekanisme *ensemble bagging*, serta kemampuannya menyediakan ukuran *Feature Importance*. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1 Alur penelitian

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder berupa *Online Gaming Behavior Dataset* yang diunduh dari platform Kaggle. Dataset ini disediakan oleh Rabie El Kharoua dan dirilis di bawah lisensi *Creative Commons Attribution 4.0 International* (CC BY 4.0), sehingga bebas digunakan untuk keperluan penelitian dengan menyertakan atribusi yang sesuai. Dataset ini terdiri

 ISSN:2302-8149
e-ISSN:2540-9719

dari 40.034 baris dan 13 kolom, yang merepresentasikan data perilaku pemain game online secara sintetis. Rincian struktur dataset disajikan pada Tabel 1.

Tabel 1 Struktur dataset

No	Column	Data Type	Deskripsi
1	PlayerID	int64	ID unik untuk setiap pemain
2	Age	int64	Usia pemain
3	Gender	object	Jenis kelamin pemain
4	Location	object	Lokasi geografis pemain
5	GameGenre	object	Genre game yang dimainkan
6	PlayTimeHours	float64	Rata-rata jam bermain per sesi
7	InGamePurchases	int64	Indikasi pembelian dalam game (0 = Tidak, 1 = Ya)
8	GameDifficulty	object	Tingkat kesulitan game
9	SessionsPerWeek	int64	Jumlah sesi bermain per minggu
10	AvgSessionDurationMinutes	int64	Durasi rata-rata setiap sesi bermain (menit)
11	PlayerLevel	int64	Level pemain saat ini dalam permainan
12	AchievementsUnlocked	int64	Jumlah pencapaian yang telah dibuka pemain
13	EngagementLevel	object	Kategori engagement pemain (Low, Medium, High)

3.2 Pembersihan Data (Data Cleaning)

Tahap pembersihan data dilakukan untuk memastikan kualitas data. Kolom *PlayerID* dihapus karena berisi pengenal unik yang tidak memiliki nilai informatif sebagai prediktor. Selanjutnya, pemeriksaan terhadap nilai kosong dan data duplikat dilakukan menggunakan pustaka *Pandas*. Hasil pemeriksaan menunjukkan bahwa data dalam keadaan lengkap (tidak ada *missing values*) dan tidak terdapat duplikasi, sehingga data dapat langsung digunakan tanpa perlu imputasi atau eliminasi baris.

3.3 Preprocessing Data

Tahap pra-pemrosesan dilakukan untuk mengubah data mentah pemain game *online* menjadi representasi numerik yang konsisten dan siap digunakan oleh algoritma *machine learning*. Proses ini melibatkan beberapa langkah utama, yaitu *encoding* variabel kategorikal, pembagian data, standarisasi fitur numerik, dan penanganan ketidakseimbangan kelas menggunakan SMOTE. Keputusan teknis pada setiap langkah didasarkan pada karakteristik algoritma *Random Forest* dan kebutuhan analisis interpretabilitas yang akan dilakukan pada tahap selanjutnya.

Proses *encoding* dilakukan dengan menggunakan *LabelEncoder* dari pustaka *Scikit-learn* untuk mengubah fitur kategorikal seperti *Gender*, *Location*, *GameGenre*, dan *GameDifficulty* menjadi representasi numerik. Variabel target *EngagementLevel* di-*encode* menggunakan *pd.Categorical* dengan urutan logis *Low* = 0, *Medium* = 1, dan *High* = 2. Penggunaan *LabelEncoder* untuk variabel nominal dipilih karena algoritma *Random Forest* tidak memanfaatkan asumsi ordinal atau jarak antarkategori, melainkan hanya menggunakan informasi pemisahan simpul berbasis nilai numerik. Pendekatan ini juga lebih efisien secara komputasi dan tidak meningkatkan dimensi data seperti halnya *One-Hot Encoding*. Tahapan ini memastikan seluruh data dalam dataset bersifat numerik dan dapat diproses oleh model.

Setelah *encoding*, dataset dipisahkan menjadi fitur (X) dan target (y). Selanjutnya, dilakukan pembagian data menjadi data latih dan data uji dengan proporsi 80:20 menggunakan fungsi *train_test_split*. Parameter *stratify=y* digunakan untuk memastikan proporsi setiap kelas pada data latih dan data uji tetap mencerminkan distribusi kelas asli. Parameter ini memastikan distribusi kelas pada kedua subset tetap seimbang sesuai data awal. Data uji diisolasi sepenuhnya dan tidak digunakan selama proses pelatihan maupun penyeimbangan data.

Langkah berikutnya adalah standarisasi fitur numerik menggunakan *StandardScaler*. Enam fitur numerik (*Age*, *PlayTimeHours*, *SessionsPerWeek*, *AvgSessionDurationMinutes*, *PlayerLevel*, dan

 ISSN:2302-8149
e-ISSN:2540-9719

AchievementsUnlocked) ditransformasi sehingga memiliki rata-rata nol dan standar deviasi satu. Proses standarisasi dinyatakan dengan rumus (1)

$$x' = \frac{x_{i\,-\,mean(x)}}{std(x)} \quad (1)$$

Nilai x_i adalah nilai asli, $mean(x)$ adalah rata-rata fitur pada data latih, $std(x)$ adalah standar deviasi fitur pada data latih, dan x'adalah nilai setelah standarisasi. Standarisasi dilakukan dengan tujuan menyamakan rentang nilai seluruh fitur numerik. Hal ini penting agar analisis *Permutation Importance* pada tahap selanjutnya tidak dipengaruhi oleh perbedaan satuan pengukuran antarfitur, sehingga kontribusi setiap fitur dapat dibandingkan secara adil.

Tahap terakhir adalah penanganan ketidakseimbangan kelas menggunakan *Synthetic Minority Over-sampling Technique* (SMOTE). Berdasarkan eksplorasi awal, kelas *Medium* mendominasi data latih dengan proporsi sekitar 48%, sementara kelas *Low* dan *High* masing-masing sekitar 26%. *SMOTE* bekerja dengan menghasilkan sampel sintetis baru pada kelas minoritas melalui interpolasi dari sampel yang sudah ada, bukan sekadar menduplikasi data. Proses interpolasi ini dinyatakan dalam rumus (2)

$$x_{new} = x_i + (x_{knn} - x_i) \times \delta \quad (2)$$

Nilai x_i adalah sampel asli dari kelas minoritas, x_{knn} adalah salah satu data yang memiliki jarak terdekat dengan x_i yang dipilih secara acak, δ merupakan bilangan acak antara 0 dan 1, dan x_{new} adalah sampel sintetis yang dihasilkan. Teknik ini diterapkan pada data latih, sedangkan data uji dibiarkan dalam distribusi aslinya untuk menjaga validitas pengujian. Untuk membuktikan efektivitas *SMOTE*, disiapkan dua skenario data latih: tanpa *SMOTE* dan dengan *SMOTE*. Perbandingan performa kedua skenario akan dievaluasi pada tahap pengujian model.

Hasil akhir dari seluruh tahapan pra-pemrosesan adalah dataset yang siap digunakan untuk pelatihan model. Struktur dataset setelah melalui proses *encoding* dan standarisasi dirangkum dalam Tabel 2.

Tabel 2 Struktur dataset setelah preprocessing

No	Column	Data Type	Deskripsi
1	Age	float64	Usia pemain
2	Gender	int64	Jenis kelamin pemain (0 = Female, 1 = Male)
3	Location	int64	Lokasi geografis pemain (0 = Asia, 1 = Europe, 2 = Other, 3 = USA)
4	GameGenre	int64	Genre game yang dimainkan (0 = Action, 1 = RPG, 2 = Simulation, 3 = Sports, 4 = Strategy)
5	PlayTimeHours	float64	Rata-rata jam bermain per sesi
6	InGamePurchases	int64	Indikasi pembelian dalam game (0 = Tidak, 1 = Ya)
7	GameDifficulty	int64	Tingkat kesulitan game (0 = Easy, 1 = Hard, 2 = Medium)
8	SessionsPerWeek	float64	Jumlah sesi bermain per minggu
9	AvgSessionDurationMinutes	float64	Durasi rata-rata setiap sesi bermain (menit)
10	PlayerLevel	float64	Level pemain saat ini dalam permainan
11	AchievementsUnlocked	float64	Jumlah pencapaian yang telah dibuka pemain
12	EngagementLevel	int64	Kategori engagement pemain (0 = Low, 1 = Medium, 2 = High)

 ISSN:2302-8149
e-ISSN:2540-9719

Selanjutnya, model *Random Forest* akan dilatih menggunakan data latih yang telah di-preprocessing. Optimasi hyperparameter akan dilakukan menggunakan *GridSearchCV* dengan validasi silang 5-*fold* untuk memperoleh kombinasi parameter terbaik, sebagaimana dijelaskan pada 3.4 Pelatihan Model. Analisis kontribusi fitur akan dilakukan melalui *Feature Importance* bawaan *Random Forest* dan *Permutation Importance* untuk memperkuat interpretabilitas model, sebagaimana diuraikan pada 3.6 Analisis *Feature Importance*.

3.4 Pelatihan Model

Pelatihan model dilakukan menggunakan *Random Forest* sebagai model utama, yang diimplementasikan dengan pustaka *Scikit-learn* pada bahasa pemrograman *Python*. *Random Forest* dipilih karena kemampuannya menangani data campuran numerik dan kategorikal , ketahanannya terhadap *overfitting* melalui mekanisme *ensemble bagging*, serta kemampuannya menyediakan ukuran *feature importance*.

Pelatihan model dilakukan pada dua skenario data latih. Pertama tanpa *SMOTE*, menggunakan data latih asli dengan distribusi kelas yang tidak seimbang; dan kedua dengan *SMOTE*, menggunakan data latih yang telah diseimbangkan melalui teknik *oversampling*. Kedua skenario dilatih menggunakan arsitektur model yang identik dan dievaluasi pada data uji yang sama untuk membuktikan efektivitas penggunaan *SMOTE*. Pendekatan komparatif ini memungkinkan pengukuran langsung dampak *SMOTE* terhadap performa model, khususnya peningkatan *recall* pada kelas minoritas.

Optimasi hyperparameter dilakukan menggunakan *RandomizedSearchCV* dengan validasi silang 5-*fold* dan 20 iterasi pencarian acak untuk memperoleh kombinasi parameter terbaik. Pendekatan ini dipilih karena efisiensi komputasi yang lebih tinggi dibandingkan *exhaustive grid search*, tanpa mengorbankan kualitas hasil optimasi secara signifikan. Ruang parameter yang dieksplorasi meliputi *n_estimators* (100, 200, 300), *max_depth* (10, 20, 30, None), *min_samples_split* (2, 5, 10), dan *min_samples_leaf* (1, 2, 4). Validasi silang 5-*fold* membagi data latih menjadi lima bagian sama besar, di mana secara bergantian empat bagian digunakan untuk pelatihan dan satu bagian untuk validasi. Proses ini diulang sebanyak lima kali hingga setiap bagian pernah menjadi data validasi. Rata-rata akurasi dari kelima iterasi digunakan sebagai dasar pemilihan kombinasi hyperparameter terbaik. Pendekatan ini meningkatkan reliabilitas hasil karena performa model tidak bergantung pada satu pembagian data tertentu.

Random Forest merupakan algoritma *ensemble* yang membangun sejumlah pohon keputusan selama proses pelatihan. Setiap pohon dilatih pada subset data yang diambil secara acak dengan pengembalian (*bootstrap sample*), dan pada setiap pemisahan simpul, sebagian fitur yang dipilih secara acak untuk dipertimbangkan (*random subspace*). Mekanisme ini memastikan keragaman antar pohon dan mengurangi korelasi di antara mereka. Prediksi akhir untuk setiap data diperoleh melalui mekanisme *hard voting*, yaitu kelas yang paling banyak dipilih oleh seluruh pohon keputusan dalam *forest*. Proses ini dinyatakan dalam rumus (3)

$$\hat{y} = mode\{h_1(x), h_2(x), \ldots, h_n(x)\} \quad (3)$$

Nilai $\hat{y}$ merupakan prediksi akhir, $h_1(x)$ merupakan prediksi dari pohon ke-*i*, dan *n* merupakan jumlah total pohon dalam *forest*.

Kombinasi hyperparameter terbaik yang dihasilkan oleh *GridSearchCV* digunakan untuk melatih model final pada seluruh data latih. Proses yang sama diterapkan pada kedua skenario (tanpa *SMOTE* dan dengan *SMOTE*). Model yang telah dilatih kemudian dievaluasi pada data uji. Hasil evaluasi kedua model dibandingkan untuk mengukur efektivitas *SMOTE*, khususnya pada metrik recall kelas *Low* dan *High*. Detail hasil evaluasi disajikan pada 3.5.

3.5 Evaluasi Model

Model yang telah dilatih diuji menggunakan data uji tanpa *SMOTE* dan data uji yang menggunakan *SMOTE*. Data uji ini tidak dilibatkan dalam proses pelatihan maupun penyeimbangan data, sehingga evaluasi memberikan gambaran objektif mengenai kemampuan generalisasi model. Perbandingan performa kedua model difokuskan pada peningkatan recall kelas *Low* dan *High* sebagai indikator efektivitas *SMOTE* dalam mengenali kelas minoritas.

Kinerja model diukur melalui metrik akurasi, *precision*, *recall*, dan *F1-score* yang dirangkum dalam *classification report*.

 ISSN:2302-8149
e-ISSN:2540-9719

Akurasi merupakan rasio antara jumlah prediksi yang benar terhadap keseluruhan data uji. Metrik ini memberikan gambaran umum tentang performa model, namun dapat menyesatkan ketika data memiliki distribusi kelas yang tidak seimbang. Secara matematis, akurasi dinyatakan dalam rumus (4)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

di mana *TP (True Positive)* adalah jumlah prediksi benar untuk kelas positif, *TN (True Negative)* adalah jumlah prediksi benar untuk kelas negatif, *FP (False Positive)* adalah jumlah prediksi salah untuk kelas positif, dan *FN (False Negative)* adalah jumlah prediksi salah untuk kelas negatif.

Precision mengukur seberapa tepat model dalam memprediksi kelas positif. Metrik ini menunjukkan proporsi prediksi positif yang benar-benar positif. Rumus (5) mendefinisikan *precision* sebagai berikut:

$$Precission = \frac{TP}{TP + FP} \quad (5)$$

Recall mengukur seberapa lengkap model dalam menangkap seluruh kelas positif yang sebenarnya. Metrik ini menunjukkan proporsi data positif yang berhasil diidentifikasi dengan benar. *Recall* dinyatakan dalam rumus (6):

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

F1-score merupakan rata-rata harmonik dari precision dan recall. Metrik ini memberikan keseimbangan antara kedua ukuran, terutama berguna ketika distribusi kelas tidak seimbang. Rumus (7) mendefinisikan *F1-score* sebagai berikut:

$$F1\ Score = 2 \times \frac{Recall \times Precission}{Recall + Precission} \quad (7)$$

Hasil evaluasi disajikan dalam bentuk *classification report* yang merangkum nilai *precision, recall, dan F1-score* untuk setiap kelas (*Low, Medium, High*), serta akurasi keseluruhan. *Confusion matrix* disajikan sebagai visualisasi pendukung untuk menganalisis pola kesalahan klasifikasi pada masing-masing kelas. Tabel perbandingan disusun untuk menampilkan metrik dari model tanpa *SMOTE* dan model dengan *SMOTE* secara berdampingan, sehingga peningkatan performa dapat diamati secara langsung. *Confusion matrix* disajikan sebagai visualisasi pendukung untuk menganalisis pola kesalahan klasifikasi pada masing-masing kelas di kedua skenario. Evaluasi ini bertujuan untuk membuktikan bahwa *SMOTE* secara efektif meningkatkan *recall* pada kelas minoritas tanpa mengorbankan performa kelas mayoritas.

3.6 Analisis Fitur Penting

Model *Random Forest* menyediakan informasi *Feature Importance* yang diekstrak untuk mengidentifikasi fitur-fitur yang paling berkontribusi dalam menentukan tingkat *engagement* pemain. Analisis ini bertujuan untuk menjawab pertanyaan penelitian ketiga, yaitu fitur perilaku apa yang paling dominan menentukan *engagement*. Agar kesimpulan lebih kokoh, digunakan dua pendekatan yang saling melengkapi: *impurity-based feature importance* dan *permutation importance*.

Impurity-based feature importance merupakan ukuran bawaan *Random Forest* yang dihitung berdasarkan rata-rata penurunan *impurity* (ketidakmurnian) yang dihasilkan oleh setiap fitur di seluruh pohon dalam *forest*. Fitur yang secara konsisten digunakan untuk membagi simpul pada posisi awal pohon dan menghasilkan penurunan *impurity* yang besar akan memiliki skor *importance* yang tinggi. Rumus (8) mendefinisikan *feature importance* untuk fitur sebagai berikut:

$$FI(j) = \frac{1}{N_{tree}} \sum_{k=1}^{N_{tree}} \sum_{t \in T_k} p(t)\, \Delta I(j,t) \quad (8)$$

di mana N_{tree} adalah jumlah total pohon dalam *forest*, T_k adalah himpunan simpul pada pohon ke-k yang menggunakan fitur j untuk membelah data, $p(t)$ adalah proporsi jumlah sampel yang melewati simpul t terhadap total sampel, dan $\Delta I(j,t)$ adalah penurunan *impurity* yang dihasilkan oleh fitur j pada simpul t.

Permutation importance digunakan sebagai pendekatan komplementer untuk mengatasi potensi bias pada *impurity-based importance* cenderung memberikan skor lebih tinggi pada fitur dengan

 ISSN:2302-8149
e-ISSN:2540-9719

banyak kategori unik, meskipun belum tentu fitur tersebut benar-benar penting bagi model. *Permutation importance* mengukur penurunan skor akurasi model ketika nilai suatu fitur diacak secara acak. Fitur yang penting akan menyebabkan penurunan akurasi yang besar saat nilainya diacak, karena model kehilangan informasi yang berguna dari fitur tersebut. Sebaliknya, fitur yang tidak penting akan menghasilkan perubahan akurasi yang kecil atau bahkan negatif. *Permutation importance* dinyatakan dalam rumus (9).

$$Permutation\ Importance(f) = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j} \quad (9)$$

di mana *s* adalah skor metrik acuan (misalnya akurasi) model pada data uji asli, *K* adalah jumlah pengulangan pengacakan, dan $s_{k,j}$ adalah skor model pada data uji dengan nilai fitur *j* yang telah diacak pada pengulangan ke-*k*.

Kedua pendekatan ini digunakan secara bersamaan. *Impurity-based importance* memberikan gambaran tentang fitur-fitur yang paling sering digunakan model dalam proses pelatihan, sedangkan *permutation importance* memberikan konfirmasi tentang seberapa besar dampak fitur tersebut terhadap performa prediksi model. Fitur yang memperoleh skor tinggi pada kedua ukuran akan diinterpretasikan sebagai fitur yang benar-benar dominan dalam menentukan *engagement* pemain. Hasil analisis ini kemudian menjadi dasar perumusan rekomendasi strategis bagi pengembang game.

4 Hasil dan Pembahasan

4.1 Hasil Pengumpulan Data

Dataset yang digunakan pada penelitian ini diambil dari platform Kaggle dengan judul *Online Gaming Behavior Dataset*. Dataset tersebut berisi 40.034 baris data dengan 13 kolom awal, yang terdiri dari satu kolom identitas (*PlayerID*), sebelas kolom fitur, dan satu kolom target (EngagementLevel). Fitur-fitur tersebut mencakup variabel demografis seperti *Age*, *Gender*, dan *Location*. Preferensi permainan seperti *GameGenre* dan *GameDifficulty*, serta metrik aktivitas bermain seperti *PlayTimeHours*, *InGamePurchases*, *SessionsPerWeek*, *AvgSessionDurationMinutes*, *PlayerLevel*, dan *AchievementsUnlocked*.

Data memiliki karakteristik yang tidak seimbang karena jumlah pemain pada tiap kategori engagement berbeda cukup signifikan. Ketidakseimbangan ini terlihat dari distribusi kelas target yang menunjukkan dominasi kategori Medium dengan 19.374 data (48,4%), diikuti High sebanyak 10.336 data (25,8%) dan Low sebanyak 10.324 data (25,8%). Seluruh 40.034 baris data digunakan dalam penelitian ini tanpa pengambilan sampel, mengingat ukuran dataset yang masih dalam batas kemampuan komputasi standar serta untuk menjaga representativitas seluruh pola perilaku pemain. Proses pembacaan data dilakukan menggunakan fungsi *read_csv()* dari pustaka *Pandas* pada bahasa pemrograman *Python*. Contoh data dari dataset yang digunakan ditunjukkan pada Tabel 3.

Tabel 3 Dataset awal

Player ID	Age	Gender	Location	Game Genre	PlayTime Hours	InGame Purchases	Game Difficulty	Sessions Per Week	Avg Session (min)	Player Level	Achievements Unlocked	Engagement Level
9000	43	Male	Other	Strategy	16.271119	0	Medium	6	108	79	25	Medium
9001	29	Female	USA	Strategy	5.525961	0	Medium	5	144	11	10	Medium
9002	22	Female	USA	Sports	8.223755	0	Easy	16	142	35	41	High
9003	35	Male	USA	Action	5.265351	1	Easy	9	85	57	47	Medium
9004	33	Male	Europe	Action	15.531945	0	Medium	2	131	95	37	Medium

4.2 Hasil Pembersihan Data

Proses pembersihan data diawali dengan penghapusan kolom *PlayerID*. Kolom ini berisi nomor identitas unik untuk setiap pemain dan tidak memiliki nilai informatif sebagai prediktor dalam proses klasifikasi. Penghapusan dilakukan menggunakan fungsi *drop()* dari pustaka *Pandas*.

Setelah penghapusan, dataset menyisakan 12 kolom yang terdiri dari 11 fitur dan 1 kolom target. Pemeriksaan terhadap kualitas data dilanjutkan dengan pengecekan keberadaan nilai kosong (*missing*

 ISSN:2302-8149
e-ISSN:2540-9719

values) dan baris data yang identik. Pemeriksaan nilai kosong dan duplikasi baris dilakukan menggunakan fungsi *sum()*.

Hasil pemeriksaan menunjukkan bahwa tidak terdapat nilai kosong pada seluruh kolom dan tidak ditemukan baris data yang identik, sehingga data dinyatakan lengkap dan seluruhnya dapat digunakan tanpa perlu proses imputasi, eliminasi, maupun penanganan duplikasi.

Pemeriksaan tipe data juga dilakukan untuk memastikan setiap kolom memiliki format yang sesuai. Kolom numerik (*Age, PlayTimeHours, InGamePurchases, SessionsPerWeek, AvgSessionDurationMinutes, PlayerLevel, AchievementsUnlocked*) telah bertipe *int64* atau *float64*, sedangkan kolom kategorikal (*Gender, Location, GameGenre, GameDifficulty, EngagementLevel*) bertipe *object*. Tidak ditemukan kesalahan format data, sehingga data dinyatakan konsisten. Dengan demikian, hasil pembersihan data menunjukkan bahwa dataset dalam kondisi lengkap, bebas duplikasi, dan memiliki tipe data yang sesuai. Dataset siap untuk memasuki tahap pra-pemrosesan lebih lanjut. Tabel 4 menunjukan hasil data *cleaning* pada dataset.

Tabel 4 Dataset setelah data cleaning

Age	Gender	Location	Game Genre	PlayTime Hours	InGame Purchases	Game Difficulty	Sessions Per Week	Avg Session (min)	Player Level	Achievements Unlocked	Engagement Level
43	Male	Other	Strategy	16.271119	0	Medium	6	108	79	25	Medium
29	Female	USA	Strategy	5.525961	0	Medium	5	144	11	10	Medium
22	Female	USA	Sports	8.223755	0	Easy	16	142	35	41	High
35	Male	USA	Action	5.265351	1	Easy	9	85	57	47	Medium
33	Male	Europe	Action	15.531945	0	Medium	2	131	95	37	Medium

4.3 Hasil Preprocessing Data

Dataset yang digunakan dalam penelitian ini adalah *Online Gaming Behavior* Dataset yang terdiri dari 40.034 baris data dan 12 fitur setelah kolom *PlayerID* dihapus. Sebelum proses pemodelan, dilakukan eksplorasi untuk memahami karakteristik data.

Tahap *preprocessing* diawali dengan proses *encoding* pada fitur kategorikal. Fitur *Gender, Location, GameGenre, dan GameDifficulty* diubah menjadi representasi numerik menggunakan *LabelEncoder*, sedangkan target *EngagementLevel* di-*encode* dengan urutan *Low* = 0, *Medium* = 1, dan *High* = 2. Setelah *encoding*, seluruh data telah berbentuk numerik dan siap diproses lebih lanjut. Contoh dari dataset setelah proses encoding ditunjukan pada Tabel 5.

Tabel 5 Dataset setelah encoding data

Age	Gender	Location	Game Genre	PlayTime Hours	InGame Purchases	Game Difficulty	Sessions Per Week	Avg Session (min)	Player Level	Achievements Unlocked	Engagement Level
43	1	2	4	16.271119	0	2	6	108	79	25	1
29	0	3	4	5.525961	0	2	5	144	11	10	1
22	0	3	3	8.223755	0	0	16	142	35	41	2
35	1	3	0	5.265351	1	0	9	85	57	47	1
33	1	1	0	15.531945	0	2	2	131	95	37	1

Dataset kemudian dipisahkan menjadi fitur (X) dan target (y), lalu dibagi ke dalam data latih (80%) dan data uji (20%) menggunakan *train_test_split* dengan parameter *stratify=y* untuk menjaga proporsi kelas pada kedua subset,.

Hasil pembagian menunjukkan data latih berjumlah 31.886 sampel dan data uji berjumlah 7.972 sampel. Hasil pembagian data masing-masing kelas *engagement* ditunjukkan pada Gambar 2.

```
Jumlah data latih SEBELUM SMOTE:
EngagementLevel
0     8220
1    15437
2     8229
Name: count, dtype: int64
```

Gambar 2 Kode pembagian data latih dan data uji

 ISSN:2302-8149
e-ISSN:2540-9719

Terlihat bahwa kelas *Medium* mendominasi data latih dengan jumlah hampir dua kali lipat dibandingkan kelas *Low* dan *High*. Mengatasi ketidakseimbangan ini, diterapkan *SMOTE* yang bekerja pada data latih.

Teknik ini menghasilkan sampel sintetis pada kelas *Low* dan *High* melalui interpolasi tetangga terdekat, sehingga masing-masing kelas memiliki jumlah sampel yang setara. Hasil pembagian data latih setelah diterapkan *SMOTE* dapat dilihat pada Gambar 3.

```
Jumlah data latih SETELAH SMOTE:
EngagementLevel
0    15437
1    15437
2    15437
Name: count, dtype: int64
```

Gambar 3 Jumlah pembagian data latih setelah SMOTE

Sebagai langkah terakhir, fitur numerik (*Age, PlayTimeHours, SessionsPerWeek, AvgSessionDurationMinutes, PlayerLevel, AchievementsUnlocked*) distandarisasi menggunakan *StandardScaler*. Proses ini mentransformasi setiap fitur sehingga memiliki rata-rata 0 dan standar deviasi 1, menyamakan skala seluruh fitur numerik.

Data latih kini memiliki jumlah sampel yang setara untuk setiap kelas. Data uji tetap dalam distribusi aslinya untuk menjaga validitas evaluasi. Standarisasi fitur numerik juga telah dilakukan dan menghasilkan fitur dengan rata-rata mendekati 0 dan standar deviasi mendekati 1. Tabel 6 menunjukkan hasil *preprocessing* data pada dataset.

Tabel 6 Dataset setelah data preprocessing data

Age	Gender	Location	Game Genre	PlayTime Hours	InGame Purchases	Game Difficulty	Sessions Per Week	Avg Session (min)	Player Level	Achievements Unlocked	Engagement Level
-0.398255	1	1	4	-0.715985	0	2	-1.471640	0.412814	-1.389655	0.867734	0
0.994159	1	0	2	-0.574392	1	1	-1.645374	-1.626444	1.270300	-0.518599	0
1.491449	1	1	0	0.529804	1	0	-0.950441	-1.483696	-0.024678	-0.033382	0
1.193075	1	3	2	1.725316	1	0	0.265691	1.432443	1.480297	-0.865182	2
-0.199339	1	3	1	-0.932776	0	0	0.091958	-0.035822	-1.319656	1.145000	1

4.4 Optimasi Hyperparameter

Optimasi hyperparameter dilakukan menggunakan *RandomizedSearchCV* dengan 20 iterasi dengan validasi silang 5-*fold* pada ruang parameter yang telah ditentukan di Bab 3. Pendekatan ini dipilih karena efisiensi komputasi yang lebih tinggi dibandingkan *exhaustive grid search*, tanpa mengorbankan kualitas hasil optimasi secara signifikan.

Proses optimasi mengeksplorasi ruang parameter yang meliputi *n_estimators* (100, 200, 300), *max_depth* (10, 20, 30, *None*), *min_samples_split* (2, 5, 10), dan *min_samples_leaf* (1, 2, 4). Kombinasi hyperparameter terbaik yang dihasilkan disajikan pada Tabel 7.

Tabel 7 Hyperparameter terbaik hasil randomizedSearchCV

No	Parameter	Nilai Terbaik
1	n_estimators	300
2	max_depth	20
3	min_samples_split	2
4	min_samples_leaf	1

Rata-rata akurasi validasi silang dengan kombinasi ini mencapai 91,57%. Konfigurasi tersebut kemudian digunakan untuk melatih model final pada kedua skenario: tanpa *SMOTE* dan dengan *SMOTE*.

4.5 Perbandingan Model

Model dilatih pada dua kondisi data latih: pertama, data latih asli tanpa penyeimbangan, dan kedua, data latih yang telah diseimbangkan dengan *SMOTE*. Kedua model menggunakan

 ISSN:2302-8149
e-ISSN:2540-9719

hyperparameter terbaik hasil optimasi (Tabel 7) dan diuji pada data uji yang identik untuk mengukur efektivitas *SMOTE*.

Model pertama dilatih menggunakan data latih asli tanpa penyeimbangan. Hyperparameter yang digunakan merupakan hasil terbaik dari optimasi *RandomizedSearchCV* (Tabel 7).

Setelah pelatihan selesai, model diuji menggunakan data latih. Evaluasi dilakukan dengan menghitung metrik akurasi, *precision*, *recall*, dan *F1-score* menggunakan fungsi *classification_report* dari *Scikit-learn*.

Hasil evaluasi model tanpa *SMOTE* yang ditampilkan Gambar 4 mencapai akurasi 91,57%. *Recall* untuk kelas *Medium* sangat tinggi (95%), namun *recall* untuk kelas *Low* dan *High* masing-masing hanya 88%. Ketimpangan ini menunjukkan bahwa model lebih banyak memprediksi data sebagai kelas mayoritas, sehingga pemain berisiko (*Low*) dan pemain loyal (*High*) kurang terdeteksi dengan baik.

```
Classification Report:
              precision    recall  f1-score   support

         Low       0.92      0.88      0.90      2065
      Medium       0.91      0.95      0.93      3875
        High       0.93      0.88      0.90      2067

    accuracy                           0.92      8007
   macro avg       0.92      0.90      0.91      8007
weighted avg       0.92      0.92      0.92      8007

Akurasi: 0.9156987635818659
```

Gambar 4 Hasil evaluasi model tanpa SMOTE

Model kedua dilatih menggunakan data latih yang telah diseimbangkan dengan SMOTE. Hyperparameter yang digunakan tetap sama dengan model pertama.

Setelah pelatihan selesai, model diuji menggunakan data latih. Evaluasi dilakukan dengan menghitung metrik akurasi, *precision*, *recall*, dan *F1-score* menggunakan fungsi *classification_report* dari *Scikit-learn*.

Model dengan *SMOTE* yang ditampilkan Gambar 5 mencapai akurasi 91,62%. *Recall* kelas *Low* meningkat menjadi 90%, sementara *recall* kelas *Medium* tetap sangat tinggi di 94%. *Recall* kelas *High* juga meningkat dari 88% menjadi 89%. *Precision* untuk kelas *High* tetap di 93%, menunjukkan bahwa prediksi untuk kategori ini tetap sangat akurat.

```
Classification Report:
              precision    recall  f1-score   support

         Low       0.91      0.90      0.90      2065
      Medium       0.92      0.94      0.93      3875
        High       0.93      0.89      0.90      2067

    accuracy                           0.92      8007
   macro avg       0.92      0.91      0.91      8007
weighted avg       0.92      0.92      0.92      8007

Akurasi: 0.9161983264643437
```

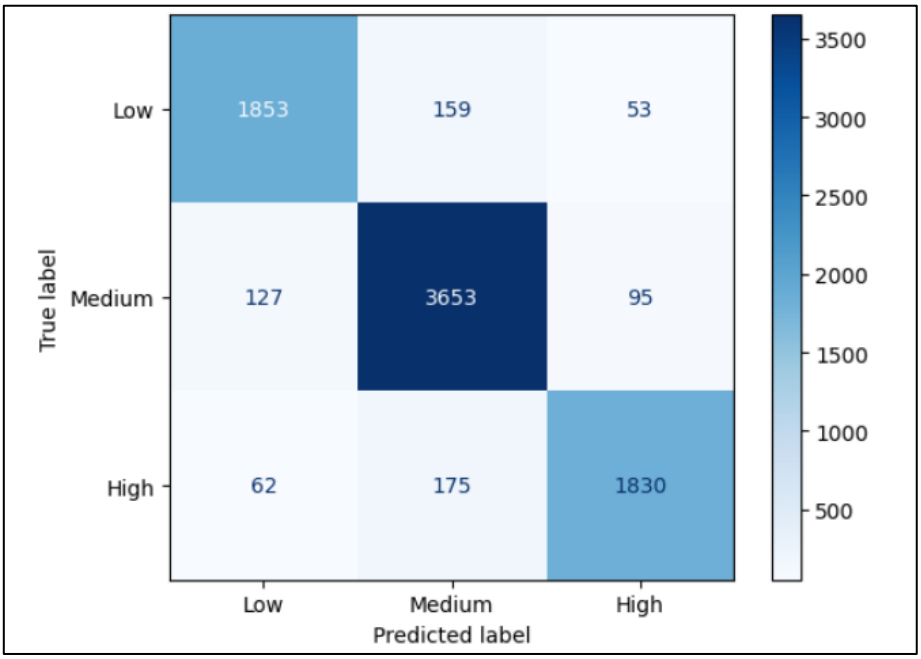
Gambar 5 Hasil evaluasi model dengan SMOTE

Peningkatan *recall* pada kelas *Low* dari 88% menjadi 90% dan pada kelas *High* dari 88% menjadi 89% menunjukkan bahwa *SMOTE* berhasil membantu model mengenali lebih banyak pemain di kedua kelas minoritas. Kemampuan ini penting agar pengembang dapat melakukan intervensi lebih dini, misalnya dengan memberikan insentif khusus kepada pemain yang mulai menunjukkan penurunan aktivitas atau mempertahankan loyalitas pemain *High*. *Recall* kelas *Medium* tetap tinggi di 94%, menegaskan bahwa penyeimbangan data tidak mengorbankan performa kelas mayoritas. Secara

 ISSN:2302-8149
e-ISSN:2540-9719

keseluruhan, *SMOTE* terbukti efektif dalam mengatasi ketidakseimbangan kelas pada dataset engagement pemain.

Pola kesalahan klasifikasi model dianalisis melalui *confusion matrix*. Visualisasi dihasilkan menggunakan fungsi *ConfusionMatrixDisplay* dari *Scikit-learn*. Hasil visualisasi *confusion matrix* ditampilkan pada Gambar 6.



Gambar 6 Confusion matrix model random forest dengan SMOTE

Data uji yang digunakan dengan total 8.007, model berhasil memprediksi dengan benar sebanyak 7.336 data. Rincian prediksi benar untuk setiap kelas adalah: 1.853 pemain *Low* diprediksi benar sebagai *Low*, 3.653 pemain *Medium* diprediksi benar sebagai *Medium*, dan 1.830 pemain *High* diprediksi benar sebagai *High*.

Kesalahan prediksi terbesar terjadi pada dua area perbatasan. Pertama, sebanyak 175 pemain *High* diprediksi sebagai *Medium*. Kedua, sebanyak 159 pemain *Low* diprediksi sebagai *Medium*. Kesalahan ekstrem, yaitu *Low* diprediksi sebagai *High* atau sebaliknya, relatif kecil dimana masing-masing hanya 53 dan 62 kasus.

Pola ini menegaskan bahwa model sangat andal dalam membedakan dua kelas yang berseberangan, yaitu *Low* dan *High*. Area "abu-abu" terletak di perbatasan antara *Low-Medium* dan *Medium-High*. Karakteristik ini wajar terjadi karena data perilaku pemain bersifat kontinu, bukan diskrit tegas. Pemain dengan *engagement* yang berada di sekitar batas antara dua kategori dapat memiliki karakteristik yang mirip dengan kedua kategori tersebut, sehingga model mengalami kesulitan pada area perbatasan ini.

4.6 Feature Importance

Model *Random Forest* menyediakan informasi *Feature Importance* yang menunjukkan kontribusi setiap fitur terhadap prediksi. *Impurity-based feature importance* merupakan ukuran bawaan *Random Forest* yang dihitung berdasarkan rata-rata penurunan *impurity* (ketidakmurnian) yang dihasilkan oleh setiap fitur di seluruh pohon dalam *forest*. Hasil *Impurity-Based Feature Importance* ditampilkan pada Gambar 7.

 ISSN:2302-8149
e-ISSN:2540-9719

```
Feature Importance (Random Forest):
SessionsPerWeek                : 0.4375
AvgSessionDurationMinutes      : 0.3371
PlayerLevel                    : 0.0480
AchievementsUnlocked           : 0.0464
PlayTimeHours                  : 0.0442
Age                            : 0.0372
GameGenre                      : 0.0163
Location                       : 0.0131
GameDifficulty                 : 0.0097
Gender                         : 0.0064
InGamePurchases                : 0.0041
```

Gambar 7 Feature importance model random forest

Berdasarkan Gambar 7, dua fitur paling dominan adalah *SessionsPerWeek* dengan skor 0,4375 dan *AvgSessionDurationMinutes* dengan skor 0,3371. Kedua fitur ini secara bersama-sama menjelaskan lebih dari 77% keputusan model. Fitur *PlayerLevel* (0,0480), *AchievementsUnlocked* (0,0464), dan *PlayTimeHours* (0,0442) memiliki kontribusi moderat. Sementara itu, fitur *Age* (0,0372), *GameGenre* (0,0163), *Location* (0,0131), *GameDifficulty* (0,0097), *Gender* (0,0064), dan *InGamePurchases* (0,0041) memberikan kontribusi yang jauh lebih kecil. Temuan ini mengindikasikan bahwa kebiasaan bermain seberapa sering dan seberapa lama pemain terlibat dalam permainan merupakan faktor utama yang membedakan tingkat *engagement* pemain.

Pendekatan kedua adalah *permutation importance* yang mengukur penurunan akurasi model ketika nilai suatu fitur diacak secara acak. *Permutation importance* dipilih karena lebih tidak bias terhadap fitur dengan banyak kategori unik dan memberikan gambaran yang lebih andal tentang kontribusi sebenarnya dari setiap fitur terhadap performa prediksi model.

```
Age                        | 0.0006 +/- 0.0007
Gender                     | 0.0004 +/- 0.0004
Location                   | 0.0006 +/- 0.0004
GameGenre                  | 0.0013 +/- 0.0004
PlayTimeHours              | 0.0012 +/- 0.0006
InGamePurchases            | 0.0006 +/- 0.0005
GameDifficulty             | 0.0004 +/- 0.0004
SessionsPerWeek            | 0.4309 +/- 0.0045
AvgSessionDurationMinutes  | 0.3540 +/- 0.0034
PlayerLevel                | 0.0302 +/- 0.0013
AchievementsUnlocked       | 0.0286 +/- 0.0014
```

Gambar 8 Hasil perhitungan permutation importance

Berdasarkan Gambar 8, *SessionsPerWeek* (0,4309) dan *AvgSessionDurationMinutes* (0,3540) tetap menjadi dua fitur paling dominan. Pengacakan kedua fitur ini menyebabkan penurunan akurasi yang paling besar. *PlayerLevel* (0,0302) dan *AchievementsUnlocked* (0,0286) memiliki kontribusi yang lebih kecil dibandingkan pada *impurity-based importance*, namun tetap menunjukkan pengaruh yang nyata. Seluruh fitur lainnya memiliki nilai permutation *importance* di bawah 0,0020. *PlayTimeHours* yang pada *impurity-based importance* tampak cukup penting (0,0442) ternyata hanya memiliki *permutation importance* sebesar 0,0012, mengonfirmasi bahwa informasi total jam bermain sudah tercakup oleh kombinasi *SessionsPerWeek* dan *AvgSessionDurationMinutes* sehingga model menganggapnya redundan.

Kedua pendekatan secara konsisten menempatkan *SessionsPerWeek* dan *AvgSessionDurationMinutes* sebagai fitur paling dominan. Konsistensi ini memperkuat keyakinan bahwa kedua fitur tersebut merupakan faktor penentu utama *engagement* pemain. Temuan ini memberikan dasar yang kokoh bagi pengembang game untuk memprioritaskan strategi yang mendorong peningkatan frekuensi sesi dan perpanjangan durasi bermain, seperti *daily login rewards*, misi harian, dan *time-limited events*.

 ISSN:2302-8149
e-ISSN:2540-9719

4.7 Pembahasan

Hasil penelitian menunjukkan bahwa model Random Forest mampu mengklasifikasikan tingkat engagement pemain dengan akurasi yang baik, yaitu 92%. Capaian ini melengkapi temuan Rismayanti et al. [3] yang menggunakan algoritma Gaussian Naive Bayes pada dataset yang sama. Pendekatan ensemble yang diterapkan dalam penelitian ini menunjukkan kemampuannya dalam menangani kompleksitas data campuran numerik dan kategorikal yang menjadi ciri khas data perilaku pemain. Hasil ini juga sejalan dengan studi Manju Usha Sree et al. [8] yang menemukan bahwa model ensemble memberikan performa yang andal pada dataset *Online Gaming Behavior*.

Penerapan *SMOTE* terlihat efektif dalam membantu model mengenali kelas-kelas yang awalnya memiliki jumlah data lebih sedikit. Sebelum penyeimbangan, kelas *Medium* mendominasi dengan proporsi sekitar 48%, sementara kelas *Low* dan *High* masing-masing sekitar 26%. Setelah *SMOTE* diterapkan, model mampu mengidentifikasi kelas *Low* dengan *recall* 90% dan kelas *High* dengan *recall* 89%. Temuan ini memperkuat hasil Prasetyaningrum et al. [9] yang mendemonstrasikan manfaat teknik *resampling* untuk meningkatkan kemampuan model dalam mengenali kelas minoritas. Penerapan *SMOTE* yang digunakan pada data latih juga memberikan gambaran performa yang realistis saat model dievaluasi pada data uji dengan distribusi asli.

Temuan bahwa *SessionsPerWeek* dan *AvgSessionDurationMinutes* merupakan dua fitur paling dominan memperkuat hasil Alzuhdi et al. [10] yang juga menempatkan frekuensi dan durasi bermain sebagai prediktor utama. Penelitian ini memperkuat temuan dengan menunjukkan bahwa kontribusi gabungan kedua fitur ini mencapai lebih dari 77%, menjadikannya faktor penentu utama. Sementara itu, fitur *InGamePurchases* yang sering dianggap sebagai indikator *engagement* justru memiliki kontribusi yang sangat kecil (0,4%). Hal ini mengindikasikan bahwa pembelian dalam game lebih merupakan akibat dari *engagement* yang sudah tinggi, bukan penyebabnya.

Kontribusi utama penelitian ini terletak pada beberapa aspek. Pertama, penelitian ini tidak sekadar mencapai akurasi tinggi, melainkan juga memberikan analisis mendalam mengenai faktor-faktor yang mendorong *engagement* melalui *Feature Importance* dengan dua pendekatan yang saling mengonfirmasi. Kedua, penerapan *SMOTE* yang dibatasi pada data latih dan evaluasi pada data uji yang tidak diubah memberikan gambaran performa yang realistis. Ketiga, penelitian ini menghasilkan rekomendasi strategis berbasis data yang dapat langsung ditindaklanjuti oleh pengembang game.

5 Kesimpulan

Penelitian ini berhasil mengembangkan model klasifikasi tingkat engagement pemain game online menggunakan algoritma *Random Forest* yang dioptimasi dengan *RandomizedSearchCV* dan validasi silang 5-*fold*. Model mencapai akurasi 92% dengan *recall* untuk kelas *Low*, *Medium*, dan *High* masing-masing sebesar 90%, 94%, dan 89%. Efektivitas *SMOTE* dibuktikan melalui eksperimen komparatif, di mana *recall* kelas *Low* meningkat dari 88% menjadi 90% dan *recall* kelas *High* dari 88% menjadi 89% setelah penyeimbangan data diterapkan. Analisis *Feature Importance* melalui pendekatan *impurity-based* dan *permutation* secara konsisten mengidentifikasi *SessionsPerWeek* dan *AvgSessionDurationMinutes* sebagai dua fitur paling dominan yang menjelaskan lebih dari 77% keputusan model. Temuan ini menegaskan bahwa kebiasaan bermain frekuensi dan durasi sesi merupakan kunci utama *engagement* pemain, menjawab ketiga rumusan masalah yang diajukan.

Kontribusi ilmiah penelitian ini terletak pada integrasi tiga aspek yang belum dilakukan secara simultan sebelumnya. Pertama, eksperimen komparatif *SMOTE*. Kedua, optimasi hyperparameter dengan validasi silang, dan ketiga, analisis interpretabilitas melalui dua pendekatan *feature importance* yang saling mengonfirmasi. Temuan ini mengindikasikan bahwa pengembang game dapat mempertimbangkan strategi retensi berbasis data, seperti *daily login rewards* dan *time-limited events*, sebagai langkah awal dalam meningkatkan *engagement* pemain. Keterbatasan penelitian terletak pada penggunaan dataset sintetis dan analisis statis tanpa mempertimbangkan aspek temporal. Penelitian selanjutnya disarankan menggunakan data riil dari game spesifik serta mengeksplorasi pendekatan *time-series* atau *deep learning* untuk menangkap dinamika perubahan *engagement* pemain secara longitudinal.

 ISSN:2302-8149
e-ISSN:2540-9719

Referensi

[1] L. Suciasih and S. Santoso, "*Analysis of Strategies for Strengthening the Online Game Industry using an Industrial Cluster Approach*," *Indonesian Journal of Business Analytics*, Vol. 4, No. 4, pp. 1629–1642, Aug. 2024, DOI: 10.55927/ijba.v4i4.10833.

[2] M. Mustofa, H. Purwanto, and F. Fandi Dwi Imaniawan, "Klasifikasi Tingkat Retensi Pemain Video Game Online," *Informatics and Computer Engineering Journal*, Vol. 4, No. 2, pp. 75–82, Aug. 2024, DOI: 10.31294/icej.v4i2.4910.

[3] N. Rismayanti, "*Predicting Online Gaming Behaviour using Machine Learning Techniques*," *Indonesian Journal of Data and Science*, Vol. 5, No. 2, Jul. 2024, DOI: 10.56705/ijodas.v5i2.166.

[4] Laurence, A. Hermawan, I. Bernarto, and F. Antonio, "*Video Game Engagement: A Passkey to the Intentions of Continue Playing, Purchasing Virtual Items, and Player Recruitment (3Ps)*," *International Journal of Computer Games Technology*, Vol. 2023, pp. 1–13, Apr. 2023, DOI: 10.1155/2023/2648097.

[5] R. Mulla *et al.*, "*Predicting Player Churn in the Gaming Industry: A Machine Learning Framework for Enhanced Retention Strategies*," *J. Curr. Sci. Technol.*, Vol. 15, No. 2, p. 103, Mar. 2025, DOI: 10.59796/jcst.V15N2.2025.103.

[6] Y. Qiu, Y. Gong, and G. Liu, "*User Behavior Analysis and Clustering in a MMO Mobile Game: Insights and Recommendations*," Sep. 2024, [Online]. Available: http://arxiv.org/abs/2407.11772

[7] A. Perišić and M. Pahor, "*Clustering Mixed-Type Player Behavior Data for Churn Prediction in Mobile Games*," *Cent. Eur. J. Oper. Res.*, Vol. 31, No. 1, pp. 165–190, Mar. 2023, DOI: 10.1007/s10100-022-00802-8.

[8] T. M. U. Sree, P. Sasikumar, S. Ayesha, M. S. Amzad Basha, and M. Martha Sucharitha, "*Predicting Player Engagement in Online Gaming: A Machine Learning Approach*," in *2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, IEEE, Sep. 2024, pp. 1–6. DOI: 10.1109/NKCon62728.2024.10774748.

[9] P. T. Prasetyaningrum, P. Purwanto, and A. F. Rochim, "*Enhancing Element Game Classification: Effective Techniques for Handling Imbalanced Classes*," *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 1, pp. 555–571, Feb. 2024, DOI: 10.22266/ijies2024.0229.47.

[10] A. V. Alzuhdi, H. Ar Rosyid, M. Y. Chuttur, and S. Nazir, "*Optimizing Random Forest Algorithm to Classify Player's Memorisation via in-game Data*," *Knowledge Engineering and Data Science*, Vol. 6, No. 1, pp. 103–113, Jul. 2023, DOI: 10.17977/um018v6i12023p103-113.

[11] Y. Durachman, "*Investigating the Impact of Gameplay Hours on Player Recommendations in Steam Games: A Comparative Analysis using Logistic Regression and Random Forest Classifiers*," *International Journal Research on Metaverse*, Vol. 2, No. 1, pp. 52–77, Mar. 2025, DOI: 10.47738/ijrm.v2i1.21.

[12] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed. Leanpub, 2022. Accessed: Jul. 10, 2026. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[13] L. Breiman, "*Random Forests*," *Mach. Learn.*, Vol. 45, No. 1, pp. 5–32, Oct. 2001, DOI: 10.1023/A:1010933404324.

[14] J. Choi and Y. Choi, "*Unlocking Player Engagement for Game Design*," 2025, pp. 278–287. DOI: 10.1007/978-3-031-94162-7_27.

[15] S. Heristian, R. S. Anwar, and H. A. Al Kautsar, "Klasifikasi Perilaku Pemain Game Online menggunakan *Naïve Bayes* berbasis *Particle Swarm Optimization*," *Computer Science (CO-SCIENCE)*, Vol. 4, No. 2, pp. 151–156, Jul. 2024, DOI: 10.31294/coscience.v4i2.4433.

[16] L. Peng, "*Research on Machine Learning Models for Predicting Player Churn*," in *Proceedings of the 1st International Conference on Modern Logistics and Supply Chain Management*, SCITEPRESS - Science and Technology Publications, 2024, pp. 112–121. DOI: 10.5220/0013234700004558.

 ISSN:2302-8149
e-ISSN:2540-9719

[17] T. Xie, "*Predicting Online Gaming Behavior: A Comparative Analysis of Random Forest and Gaussian Naive Bayes Models*," *Science and Technology of Engineering, Chemistry and Environmental Protection*, Vol. 1, No. 9, Oct. 2024, DOI: 10.61173/5hthrr73.
[18] J. Pan, "*Predict Players Engagement Level using KNN Method*," *Academic Journal of Science and Technology*, Vol. 19, No. 2, pp. 78–83, Jan. 2026, DOI: 10.54097/e7dscf70.