

PREDIKSI KEMENANGAN MPL INDONESIA SEASON 16 BERDASARKAN HERO DRAFT MENGGUNAKAN RANDOM FOREST

Adam Nurwahid Alfarizi^{1*}, Duwi Cahya Putri Buani²

^{1,2}Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Nusa Mandiri; Jl. Pemuda No.9, Rawamangun, Kec. Pulo Gadung, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13220; (021-28534471)

Keywords:

Random Forest;
Machine learning;
Hero draft;
Mobile Legends.

Correspondent Email:

12220065@nusamandiri.ac.id

Abstrak. Mobile Legends: Bang Bang (MLBB) memiliki kompetisi profesional melalui Mobile Legends Professional League (MPL). Prediksi kemenangan berdasarkan fase *hero draft* menjadi salah satu masalah yang dapat dianalisis menggunakan *machine learning*. Penelitian ini menerapkan algoritma *Random Forest* untuk memprediksi kemenangan pertandingan MPL Indonesia Season 16 menggunakan 207 data pertandingan yang diperoleh dari Liquipedia. Tahapan penelitian meliputi pengumpulan data, *preprocessing*, pembangunan model pada Altair AI Studio (RapidMiner), *tuning parameter*, dan evaluasi menggunakan *10-fold Cross Validation*. Konfigurasi akhir menggunakan 1200 *trees*, *criterion Gini Index*, *maximal depth* 20, *subset ratio* 0,6, serta *prepruning*. Hasil pengujian menunjukkan akurasi $70,07\% \pm 8,83\%$ dengan *micro average* 70,05%, meningkat dari konfigurasi awal 57,02%. Hasil tersebut menunjukkan bahwa *Random Forest* dapat digunakan untuk memprediksi kemenangan berdasarkan atribut yang digunakan, meskipun faktor di luar dataset tetap dapat memengaruhi hasil pertandingan.

Abstract. Mobile Legends: Bang Bang (MLBB) features a professional competition through the Mobile Legends Professional League (MPL). Predicting match outcomes from the hero draft phase is a problem that can be analyzed using machine learning. This study applies the Random Forest algorithm to predict MPL Indonesia Season 16 match outcomes using 207 matches collected from Liquipedia. The research stages include data collection, preprocessing, model development in Altair AI Studio (RapidMiner), parameter tuning, and evaluation using 10-fold Cross Validation. The final configuration uses 1,200 trees, the Gini Index criterion, a maximum depth of 20, a subset ratio of 0.6, and pre-pruning. The model achieved $70.07\% \pm 8.83\%$ accuracy with a 70.05% micro average, improving from the initial 57.02% configuration. These findings indicate that Random Forest can be used to predict match outcomes based on the attributes analyzed, although factors outside the dataset may still influence match results.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

1. PENDAHULUAN

Perkembangan teknologi informasi turut mendorong pertumbuhan industri hiburan digital dan *esports*. Mobile Legends: Bang Bang (MLBB) menjadi salah satu permainan *Multiplayer Online Battle Arena* (MOBA) yang memiliki basis pemain besar serta ekosistem kompetitif profesional. Mobile Legends Professional League (MPL) berkembang menjadi kompetisi utama di Indonesia dan berkontribusi terhadap perkembangan industri *esports*, peluang ekonomi, serta pengembangan talenta [1], [2].

Salah satu tahap penting dalam pertandingan MLBB adalah fase *hero draft*, ketika kedua tim secara bergantian memilih (*pick*) dan melarang (*ban*)

hero. Fase ini menjadi dasar penyusunan strategi sebelum pertandingan dimulai. Namun, hasil pertandingan tidak hanya dipengaruhi oleh komposisi *hero*, tetapi juga oleh performa pemain, strategi selama pertandingan, penguasaan objektif, serta kondisi permainan. Penelitian terdahulu menunjukkan bahwa data *draft pick* tetap dapat digunakan sebagai dasar prediksi, walaupun akurasi dapat berbeda dibandingkan penggunaan data *in-game* [3].

Kajian prediksi kemenangan pada permainan MOBA telah memanfaatkan berbagai algoritma *machine learning*. Tangniyom dan Boonma menunjukkan bahwa hubungan antar*hero* dan pendekatan *machine learning* dapat digunakan

untuk memprediksi pemenang pada permainan MOBA [4]. Penelitian lain pada konteks *esports* juga menunjukkan bahwa metode *ensemble* seperti *Random Forest* memiliki performa kompetitif dalam tugas klasifikasi. Hal tersebut mendukung pemilihan *Random Forest* sebagai algoritma utama karena model dibangun dari sejumlah *decision tree* dan menggabungkan prediksi setiap pohon untuk menghasilkan keputusan yang lebih stabil.

Pada konteks MPL Indonesia, proses analisis *hero draft* sering dilakukan secara manual dan dapat dipengaruhi subjektivitas. Selain itu, penelitian yang secara khusus menggunakan data pertandingan MPL Indonesia Season 16 untuk membangun model prediksi berbasis *hero draft* masih terbatas. Kesenjangan ini menjadi dasar penelitian untuk menerapkan *Random Forest* terhadap data pertandingan musim tersebut dan mengevaluasi kemampuan model melalui pengujian terstruktur.

Penelitian ini bertujuan menerapkan algoritma *Random Forest* untuk membangun model prediksi kemenangan pertandingan berdasarkan atribut yang tersedia pada data *hero draft* MPL Indonesia Season 16 serta mengevaluasi tingkat akurasi. Fokus penelitian ditempatkan pada kinerja *Random Forest* setelah proses *tuning parameter*. Perbandingan dengan *Decision tree*, *K-Nearest Neighbor*, dan *Naive Bayes* digunakan sebagai analisis tambahan untuk memberi konteks terhadap hasil model utama.

2. TINJAUAN PUSTAKA

2.1 Machine learning dan Random Forest

Machine learning merupakan bagian dari kecerdasan buatan yang memungkinkan sistem mempelajari pola dari data dan pengalaman untuk melakukan prediksi atau klasifikasi tanpa aturan eksplisit untuk setiap kondisi [5], [6]. Dalam penelitian ini, masalah prediksi kemenangan diperlakukan sebagai klasifikasi terawasi karena data pertandingan memiliki label pemenang yang digunakan sebagai target model.

Random Forest merupakan algoritma *ensemble* yang membangun banyak *decision tree* dari variasi sampel dan atribut, kemudian menggabungkan hasil prediksi pohon-pohon tersebut. Mekanisme ini membantu mengurangi ketergantungan pada satu *decision tree* dan dapat meningkatkan stabilitas model [7], [8]. Implementasi klasifikasi dan pengolahan data pada penelitian dilakukan menggunakan Altair AI Studio (RapidMiner), yang menyediakan operator untuk seleksi atribut, konversi data, *Cross Validation*, pembangunan model, penerapan model, serta pengukuran performa [9].

2.2 Evaluasi Model Klasifikasi

Evaluasi model dilakukan menggunakan akurasi, *precision*, *recall*, dan *confusion matrix*. Akurasi digunakan untuk mengukur proporsi

prediksi yang benar terhadap keseluruhan data pengujian. *Precision* menunjukkan ketepatan prediksi pada suatu kelas, sedangkan *recall* menunjukkan kemampuan model mengenali data aktual pada kelas tersebut. *Confusion matrix* digunakan untuk melihat distribusi hasil prediksi benar dan salah pada setiap kelas tim [10]. Dalam penelitian multi-kelas ini, nilai *class precision* dan *class recall* dianalisis untuk melihat variasi kemampuan model terhadap sembilan tim.

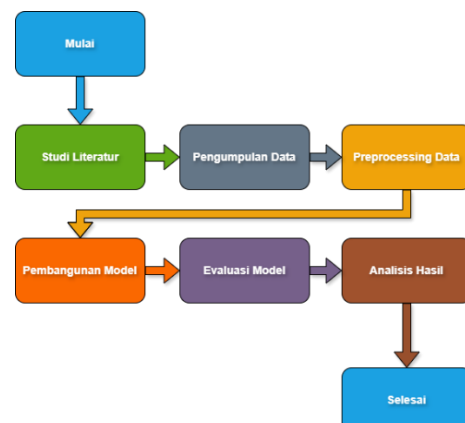
2.3 Penelitian Terkait

Penelitian pada bidang *esports* menunjukkan bahwa karakteristik data dan pemilihan algoritma dapat menghasilkan tingkat akurasi yang berbeda. Dikananda dkk. membandingkan *Decision tree*, *Naive Bayes*, dan *Random Forest* pada klasifikasi turnamen *esports* dan menunjukkan kemampuan *Random Forest* sebagai metode *ensemble* [11]. Valenzuela dkk. menggunakan *machine learning* untuk prediksi kemenangan Dota 2 dengan data pra-pertandingan dan menekankan aspek interpretabilitas model [12]. Tyran dan Chomatek menunjukkan bahwa kualitas data, termasuk keberadaan outlier, dapat memengaruhi efektivitas prediksi kemenangan pada permainan MOBA [13]. Arik juga membandingkan beberapa model *machine learning* untuk prediksi pemenang League of Legends dan memperoleh variasi performa antaralgoritma [14]. Penelitian-penelitian tersebut menjadi dasar bahwa prediksi kemenangan MOBA dapat dilakukan menggunakan data pra-pertandingan dengan evaluasi yang disesuaikan terhadap karakteristik dataset.

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian disusun secara berurutan mulai dari studi literatur, pengumpulan data, *preprocessing*, pembangunan model, evaluasi model, hingga analisis hasil. Alur ini digunakan agar proses pemodelan dapat dilaksanakan secara sistematis dan hasil pengujian dapat ditelusuri kembali.



Gambar 3.1 Tahapan Penelitian

3.2 Pengumpulan Data dan Karakteristik Dataset

Penelitian menggunakan data sekunder pertandingan MPL Indonesia Season 16 yang diperoleh dari Liquipedia [15]. Dataset mencakup 207 game, terdiri atas 170 game Regular Season dan 37 game Playoff, dengan sembilan tim. Setiap baris merepresentasikan satu game dan dataset awal memiliki 26 atribut yang mencakup Match_ID, Game, Phase, Team_1, Team_2, sepuluh atribut *hero ban*, sepuluh atribut *hero pick*, serta Winner sebagai label klasifikasi.

Tabel 3.1 Ringkasan Dataset MPL Indonesia Season 16

Keterangan	Jumlah
Total Game	207
Regular Season	170
Playoff	37
Jumlah Tim	9
Jumlah Atribut	26

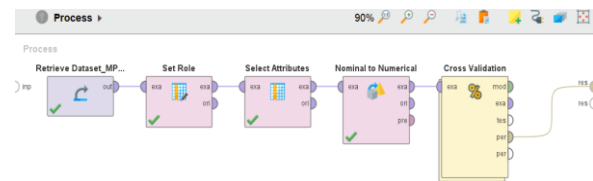
3.3 Preprocessing Data

Tahap *preprocessing* dilakukan agar data sesuai dengan kebutuhan pemodelan. Atribut Match_ID, Game, dan Phase dikeluarkan karena berfungsi sebagai identitas dan konteks pertandingan, bukan sebagai prediktor. Atribut yang dipertahankan adalah Team_1, Team_2, seluruh atribut *Hero Pick*, seluruh atribut *Hero Ban*, serta Winner. Atribut Winner kemudian ditetapkan sebagai label klasifikasi, sedangkan atribut kategori dikonversi ke bentuk numerik menggunakan operator Nominal to Numerical.

Pembagian data dilakukan menggunakan *10-fold Cross Validation*. Dataset dibagi menjadi sepuluh *fold*; pada setiap iterasi satu *fold* digunakan sebagai data pengujian dan sembilan *fold* lainnya sebagai data pelatihan. Proses diulang hingga seluruh *fold* memperoleh giliran sebagai data pengujian. Pendekatan ini memungkinkan seluruh data digunakan dalam proses *training* dan *testing* secara bergantian serta memberikan estimasi kinerja yang lebih konsisten pada dataset yang jumlahnya terbatas.

3.4 Pembangunan dan Tuning Model

Alur pembangunan model pada Altair AI Studio (RapidMiner) terdiri atas Retrieve, Set Role, Select Attributes, Nominal to Numerical, Cross Validation, Random Forest, Apply Model, dan Performance. Pada sisi *training*, Random Forest digunakan untuk membangun model. Pada sisi *testing*, Apply Model menerapkan model pada *fold* pengujian, sedangkan Performance menghitung metrik evaluasi.

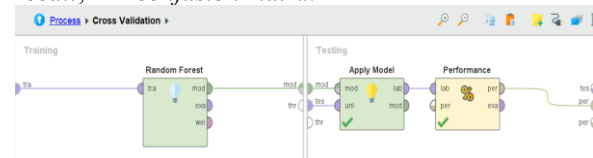


Gambar 3.2 Alur Pembangunan Model di RapidMiner

Tuning parameter dilakukan secara bertahap dengan menguji *number of trees*, *maximal depth*, *criterion*, *subset ratio*, dan *prepruning*. Konfigurasi awal menggunakan 200 *trees*, *maximal depth* 10, *criterion* Gain Ratio, serta *subset ratio* default. Percobaan selanjutnya meningkatkan jumlah pohon, mengubah *criterion* menjadi Gini Index, menyesuaikan *maximal depth*, dan mengatur *subset ratio*. Konfigurasi final menggunakan 1200 *trees*, *criterion* gini_index, *maximal depth* 20, *subset ratio* 0,6, *prepruning* aktif, *minimal gain* 0,01, *minimal leaf size* 1, *minimal size for split* 4, dan *number of prepruning alternatives* 3.

3.5 Evaluasi Model

Evaluasi model dilakukan menggunakan *10-fold Cross Validation*. Pada setiap iterasi, sembilan *fold* digunakan sebagai data pelatihan dan satu *fold* sebagai data pengujian. Model Random Forest dibangun pada sisi *training*, kemudian diterapkan pada sisi *testing* menggunakan Apply Model dan dievaluasi menggunakan operator Performance. Metrik yang dianalisis meliputi *accuracy*, *precision*, *recall*, dan *confusion matrix*.



Gambar 3.3 Alur Kerja di dalam Cross Validation

4. HASIL DAN PEMBAHASAN

4.1 Deskripsi dan Distribusi Data

Distribusi kelas pada dataset tidak sepenuhnya seimbang. ONIC memiliki jumlah kemenangan tertinggi sebanyak 39 game, sedangkan GEEK memiliki jumlah kemenangan terendah sebanyak 12 game. Perbedaan jumlah data pada masing-masing kelas dapat memengaruhi kemampuan model dalam mempelajari pola kemenangan setiap tim, sehingga evaluasi tidak hanya melihat akurasi keseluruhan, tetapi juga *class recall* dan *class precision*.

Tabel 4.1 Distribusi Data Pemenang per Tim

Tim	Jumlah Kemenangan
ONIC	39
AE	36
EVOS	30
BTR	26
DEWA	19
NAVI	17

Tim	Jumlah Kemenangan
RRQ	15
TLID	13
GEEK	12

4.2 Implementasi dan Pengujian *Random Forest*

Hasil tuning menunjukkan peningkatan akurasi secara bertahap. Konfigurasi awal menghasilkan akurasi 57,02%. Penggunaan 500 *trees* dengan *Gini Index* meningkatkan akurasi menjadi 65,21%, kemudian pengaturan *maximal depth* 20 menghasilkan 67,67%. Peningkatan jumlah *trees* menjadi 1000 menghasilkan 68,14%. Penurunan *subset ratio* menjadi 0,6 meningkatkan akurasi menjadi 69,12%, dan kombinasi 1200 *trees*, *maximal depth* 20, *Gini Index*, *subset ratio* 0,6, serta *prepruning* memberikan nilai tertinggi sebesar 70,07%.

Tabel 4.2 Hasil Tuning parameter *Random Forest*

Perc.	Konfigurasi	Akurasi
1	200 <i>trees</i> ; depth 10; <i>Gain Ratio</i> ; default	57,02%
2	500 <i>trees</i> ; depth -1; <i>Gini</i> ; 0,7	65,21%
3	500 <i>trees</i> ; depth 20; <i>Gini</i> ; 0,7	67,67%
4	1000 <i>trees</i> ; depth 20; <i>Gini</i> ; 0,7	68,14%
5	1000 <i>trees</i> ; depth 20; <i>Gini</i> ; 0,6	69,12%
6	1200 <i>trees</i> ; depth 20; <i>Gini</i> ; 0,6	70,07%

Peningkatan dari 57,02% menjadi 70,07% menunjukkan bahwa pemilihan parameter berpengaruh terhadap kinerja *Random Forest* pada dataset ini. Penambahan jumlah *trees* memberi model lebih banyak pohon untuk menghasilkan keputusan kolektif, sedangkan pembatasan *maximal depth* dan penerapan *prepruning* membantu mengendalikan kompleksitas pohon. *Subset ratio* 0,6 memberikan hasil yang lebih baik dibandingkan 0,7 pada konfigurasi yang diuji. Karena pengujian parameter dilakukan secara eksperimental pada dataset dan skenario penelitian ini, konfigurasi tersebut diperlakukan sebagai konfigurasi terbaik pada ruang percobaan yang dilakukan, bukan sebagai parameter universal untuk seluruh dataset MOBA.

4.3 Hasil Pengujian Model

Model akhir diuji menggunakan *10-fold Cross Validation* pada 207 game. Hasil pengujian memperoleh akurasi 70,07% $\pm$ 8,83% dengan *micro average* 70,05%. Nilai deviasi antar-fold menunjukkan bahwa performa model masih bervariasi pada pembagian data yang berbeda. Walaupun demikian, akurasi akhir telah meningkat 13,05 poin persentase dibandingkan konfigurasi awal 57,02%.

accuracy: 70.07% $\pm$ 8.83% (micro average: 70.05%)

	true ONIC	true EVOS	true GEEK	true TLID	true AE	true RRQ	true NAVI	true BTR	true DEWA	class pred...
pred. ONIC	36	3	0	0	6	0	1	0	0	78.26%
pred. EVOS	2	20	0	1	2	1	1	2	2	64.52%
pred. GEEK	0	0	9	1	0	0	0	0	0	90.00%
pred. TLID	0	2	2	7	1	0	2	0	1	46.67%
pred. AE	0	4	0	2	22	0	2	4	0	64.71%
pred. RRQ	0	0	0	0	1	12	1	2	0	75.00%
pred. NAVI	1	0	0	1	1	0	9	1	0	69.23%
pred. BTR	0	1	0	0	3	2	1	15	1	65.22%
pred. DEWA	0	0	1	1	0	0	0	2	15	78.95%
class recall	92.31%	66.67%	75.00%	53.85%	61.11%	80.00%	52.94%	57.69%	78.95%	

Gambar 4.1 Hasil Pengujian Model *Random Forest*

Akurasi di atas 70% menunjukkan bahwa atribut yang digunakan mampu memberikan informasi yang dapat dipelajari *Random Forest* untuk melakukan klasifikasi pemenang. Namun, hasil tersebut tidak dapat diinterpretasikan bahwa *hero draft* menjadi satu-satunya penentu kemenangan. Dataset juga memuat identitas tim sebagai atribut input, sedangkan faktor lain seperti performa pemain, strategi selama pertandingan, penguasaan objektif, dan kondisi permainan tidak digunakan. Oleh karena itu, hasil model merepresentasikan kemampuan prediksi berdasarkan atribut yang tersedia pada dataset penelitian.

4.3.1 Analisis Confusion matrix

Analisis per kelas menunjukkan variasi yang cukup besar. ONIC memperoleh *class recall* tertinggi sebesar 92,31%, yang berarti sebagian besar pertandingan dengan kemenangan aktual ONIC dapat dikenali model. RRQ memperoleh *recall* 80,00%, DEWA 78,95%, dan GEEK 75,00%. Sebaliknya, NAVI memperoleh *recall* 52,94% dan TLID 53,85%, menunjukkan bahwa model lebih sering gagal mengenali kemenangan aktual pada dua kelas tersebut.

Dari sisi *precision*, GEEK memperoleh nilai tertinggi sebesar 90,00%, disusul DEWA 78,95%, ONIC 78,26%, dan RRQ 75,00%. TLID memperoleh *precision* terendah 46,67%. Perbedaan *precision* dan *recall* ini menunjukkan bahwa akurasi keseluruhan tidak cukup untuk menggambarkan kinerja pada setiap tim. Ketidakseimbangan distribusi kemenangan, variasi pola draft, dan karakteristik pertandingan masing-masing tim dapat berkontribusi pada perbedaan performa tersebut.

Tabel 4.3 Class recall dan Class precision per Tim

Tim	Recall / Precision
ONIC	92,31% / 78,26%
EVOS	66,67% / 64,52%
GEEK	75,00% / 90,00%
TLID	53,85% / 46,67%
AE	61,11% / 64,71%
RRQ	80,00% / 75,00%
NAVI	52,94% / 69,23%

Tim	Recall / Precision
BTR	57,69% / 65,22%
DEWA	78,95% / 78,95%

4.4 Pembahasan

Sebagai analisis tambahan, *Random Forest* dibandingkan dengan *Decision tree*, *K-Nearest Neighbor*, dan *Naive Bayes* menggunakan dataset serta metode evaluasi *10-fold Cross Validation* yang sama. *Random Forest* menghasilkan akurasi 70,07%, lebih tinggi dibandingkan *Decision tree* 54,50%, *K-Nearest Neighbor* 44,90%, dan *Naive Bayes* 32,50%.

Tabel 4. 4 Perbandingan Kinerja Algoritma

Algoritma	Akurasi
<i>Random Forest</i>	70,07%
<i>Decision tree</i>	54,50%
<i>K-Nearest Neighbor</i>	44,90%
<i>Naive Bayes</i>	32,50%

Perbedaan tersebut mendukung penggunaan *Random Forest* sebagai algoritma utama dalam penelitian ini. Secara konseptual, penggunaan banyak *decision tree* memungkinkan *Random Forest* mempelajari pola yang lebih beragam dibandingkan satu pohon keputusan. Hasil ini sejalan dengan penelitian Dikananda dkk. yang menunjukkan kemampuan *Random Forest* pada klasifikasi terkait *esports* [11]. Namun, hasil penelitian tidak dimaksudkan untuk menyimpulkan bahwa *Random Forest* selalu unggul pada seluruh masalah prediksi MOBA, karena performa algoritma sangat dipengaruhi karakteristik data, atribut, tahap *preprocessing*, serta skenario evaluasi.

4.4.1 Perbandingan dengan Penelitian Terdahulu.

Hasil penelitian memperkuat temuan bahwa data pra-pertandingan atau draft dapat dimanfaatkan untuk prediksi kemenangan pada permainan MOBA. Firmansyah dkk. melaporkan bahwa prediksi pada fase *draft pick* menghasilkan akurasi yang lebih rendah dibandingkan prediksi yang memasukkan informasi *in-game* [3]. Pola serupa dapat menjelaskan mengapa akurasi penelitian ini berhenti pada 70,07% walaupun telah dilakukan *tuning parameter*: informasi *hero draft* dan identitas tim hanya menggambarkan kondisi sebelum pertandingan, sedangkan dinamika pertandingan setelah draft tidak dimasukkan ke model.

Temuan tersebut juga berkaitan dengan penelitian Valenzuela dkk. yang menunjukkan bahwa data pemilihan karakter sebelum pertandingan dapat digunakan untuk prediksi kemenangan Dota 2 [12]. Hal ini menunjukkan bahwa informasi pra-pertandingan memiliki nilai prediktif pada permainan MOBA, tetapi besarnya kemampuan prediksi dapat berbeda antarpermainan

dan dataset. Sementara itu, Tyran dan Chomatek menekankan pengaruh kualitas data terhadap efektivitas model [13]. Pada penelitian ini, distribusi kelas yang tidak seimbang dan jumlah data yang hanya 207 game menjadi keterbatasan yang perlu diperhatikan ketika menginterpretasikan hasil.

Dibandingkan model pada penelitian Arik yang menggunakan metrik performa dan karakteristik lain pada League of Legends [14], penelitian ini memiliki ruang atribut yang lebih terbatas karena fokus pada data yang tersedia pada fase *hero draft* dan identitas tim. Perbedaan jenis atribut tersebut menunjukkan bahwa penambahan informasi yang relevan berpotensi meningkatkan kemampuan model. Oleh karena itu, pengembangan selanjutnya dapat menggunakan beberapa musim MPL Indonesia, menambahkan statistik *head-to-head* tim, *win rate hero* pada patch tertentu, performa pemain, atau atribut pertandingan lain yang tersedia sebelum maupun selama pertandingan.

Dari sisi implementasi, hasil tuning juga menunjukkan pentingnya pengaturan parameter *Random Forest*. Perubahan jumlah pohon, kedalaman maksimum, *subset ratio*, dan *prepruning* menghasilkan peningkatan akurasi yang konsisten pada eksperimen yang dilakukan. Hal ini menunjukkan bahwa penggunaan konfigurasi *default* atau konfigurasi awal belum tentu memberikan performa terbaik. Meskipun demikian, proses tuning pada penelitian masih terbatas pada beberapa kombinasi parameter sehingga belum mencakup pencarian ruang parameter yang luas. Pengujian lanjutan dapat menggunakan pendekatan optimasi parameter yang lebih sistematis serta *feature selection* atau *feature engineering*.

4.4.2 Analisis Performa Setiap Kelas Tim.

Pada kelas ONIC, *recall* sebesar 92,31% merupakan nilai tertinggi pada seluruh kelas. Hal ini menunjukkan bahwa model relatif konsisten mengenali pertandingan yang secara aktual dimenangkan ONIC. *Precision* sebesar 78,26% juga menunjukkan bahwa sebagian besar prediksi ONIC sebagai pemenang sesuai dengan kelas aktual. Kombinasi *recall* dan *precision* tersebut menempatkan ONIC sebagai salah satu kelas yang paling mudah dikenali oleh model pada dataset penelitian.

Pada kelas EVOS, *recall* sebesar 66,67% dan *precision* sebesar 64,52% memperlihatkan kemampuan klasifikasi yang berada pada tingkat menengah. Model mampu mengenali sekitar dua pertiga kemenangan aktual EVOS, tetapi masih terdapat prediksi EVOS yang tidak sesuai dengan kelas aktual. Kondisi tersebut menunjukkan bahwa pola atribut pada pertandingan EVOS belum sepenuhnya terpisah dari pola tim lain.

GEEK memiliki karakteristik yang berbeda. *Recall* sebesar 75,00% menunjukkan bahwa sebagian besar kemenangan aktual GEEK dapat dikenali, sedangkan *precision* 90,00% merupakan nilai tertinggi. Artinya, ketika model memberikan prediksi GEEK sebagai pemenang, prediksi tersebut sangat sering benar. Walaupun jumlah kemenangan GEEK pada dataset merupakan yang paling sedikit, yaitu 12 game, pola atribut pada sebagian pertandingan GEEK tetap dapat dikenali dengan cukup spesifik oleh model.

TLID menjadi salah satu kelas dengan performa terendah, dengan *recall* 53,85% dan *precision* 46,67%. Nilai tersebut menunjukkan bahwa hampir setengah prediksi TLID sebagai pemenang tidak sesuai dengan kelas aktual dan sebagian kemenangan aktual TLID juga tidak berhasil dikenali. Hasil ini dapat berkaitan dengan jumlah kemenangan TLID yang hanya 13 game dan kemungkinan adanya pola *hero draft* yang tumpang tindih dengan tim lain.

AE memperoleh *recall* 61,11% dan *precision* 64,71%. Meskipun AE memiliki 36 kemenangan dan menjadi salah satu kelas dengan jumlah data yang besar, kinerja klasifikasinya tidak setinggi ONIC. Hal ini menegaskan bahwa jumlah data per kelas bukan satu-satunya faktor yang memengaruhi kemampuan model; variasi pola atribut dan kemiripan strategi draft antarpertandingan juga dapat memengaruhi proses klasifikasi.

RRQ memperoleh *recall* 80,00% dan *precision* 75,00%. Nilai *recall* yang tinggi menunjukkan bahwa model dapat mengenali sebagian besar kemenangan aktual RRQ. NAVI memiliki *recall* 52,94% dan *precision* 69,23%, sedangkan BTR memperoleh *recall* 57,69% dan *precision* 65,22%. Pada dua kelas tersebut, *precision* lebih tinggi daripada *recall*, yang berarti prediksi ketika model memilih kelas tersebut relatif lebih tepat dibandingkan kemampuan model dalam menemukan seluruh kemenangan aktualnya.

DEWA memperoleh *recall* dan *precision* yang sama, yaitu 78,95%. Kesamaan ini menunjukkan keseimbangan antara kemampuan model mengenali kemenangan aktual DEWA dan ketepatan saat memprediksi DEWA sebagai pemenang. Secara keseluruhan, variasi antar kelas menunjukkan pentingnya penggunaan *confusion matrix* pada masalah klasifikasi multi-kelas karena satu nilai akurasi tidak dapat menggambarkan keberhasilan model pada setiap tim secara terpisah.

5. KESIMPULAN

Penelitian berhasil membangun model prediksi kemenangan MPL Indonesia Season 16 menggunakan algoritma *Random Forest* pada Altair AI Studio (RapidMiner). Model menggunakan data

207 game dengan *preprocessing* atribut, konversi data nominal, dan evaluasi *10-fold Cross Validation*. Konfigurasi akhir terdiri atas 1200 *trees*, *criterion gini_index*, *maximal depth* 20, *subset ratio* 0,6, serta *prepruning* yang diaktifkan.

Model akhir memperoleh akurasi 70,07% $\pm$ 8,83% dengan *micro average* 70,05%, meningkat dari konfigurasi awal 57,02%. *Random Forest* juga menghasilkan akurasi lebih tinggi dibandingkan *Decision tree*, *K-Nearest Neighbor*, dan *Naive Bayes* pada skenario pengujian yang sama. Kinerja per kelas masih bervariasi, dengan *recall* tertinggi pada ONIC sebesar 92,31% dan *precision* tertinggi pada GEEK sebesar 90,00%.

Keterbatasan penelitian terletak pada jumlah data, distribusi kelas yang tidak seimbang, dan belum digunakannya faktor lain di luar atribut dataset. Penelitian berikutnya dapat menggunakan beberapa musim MPL, menambahkan atribut performa pemain dan tim, serta melakukan pengujian parameter, *feature selection*, dan *feature engineering* yang lebih luas untuk memperoleh model yang lebih konsisten.

1. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, serta pihak-pihak yang telah mendukung pelaksanaan penelitian dan penyusunan naskah ini.

2. DAFTAR PUSTAKA

- [1] S. S. Sanjaya, A. K. Jaya, R. Candra, and S. Zang, "Sentiment Analysis of Mobile Legends Game using Naïve Bayes, K-Nearest Neighbors and Support Vector Machine Algorithm," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 5, no. 2, pp. 2252–2260, Feb. 2026, doi: 10.59934/JAIEA.V5I2.1840.
- [2] A. Maulida Roziana Putri et al., "Pengembangan E-Sport Di Indonesia, Peluang, Tantangan, Dan Masa Depan Industri: Studi Kasus Mobile Legends Profesional League (MPL)," *Jurnal Media Akademik (JMA)*, vol. 2, no. 10, pp. 3031–5220, Oct. 2024, doi: 10.62281/V2I10.795.
- [3] D. F. Firmansyah, A. P. Kuncoro, and R. Riyanto, "Match Outcome Prediction in Draft pick and In-game Phases of MSC 2025 Mobile Legends using Random Forest and XGBoost," *Journal of Applied Informatics and Computing*, vol. 9, no. 6, pp. 3892–3903, Dec. 2025, doi: 10.30871/JAIC.V9I6.11658.
- [4] N. Tangniyom and P. Boonma, "A Study on Using Machine learning to Predict Winner in Multiplayer Online Battle Arena (MOBA) Game," *ASEAN Journal of Scientific and*

- Technological Reports, vol. 27, no. 5, pp. e252289–e252289, Sep. 2024, doi: 10.55164/AJSTR.V27I5.252289.
- [5] Dr. Feri Sulianta, Dasar dan Konsep Machine learning. Feri Sulianta, 2025. Accessed: Apr. 29, 2026. [Online]. Available: https://www.google.co.id/books/edition/Dasar_dan_Konsep_Machine_Learning/15BwEQA_AQBAJ?hl=id&gbpv=0
- [6] S. Kom. M. M. Asrul et al., Machine learning Teori, Model, Dan Implementasi-Sebuah Pendekatan Komprehensif. Yash Media, 2025. Accessed: Apr. 29, 2026. [Online]. Available: https://www.google.co.id/books/edition/MACHINE_LEARNING_TEORI_MODEL_DAN_IMPLERME/afXEEQAAQBAJ?hl=id&gbpv=0
- [7] Muhammad Reza Romahdoni, Mengenal Algoritma Random Forest: Cara Mudah Menerapkan Machine learning di Bisnis UMKM. Serasi Media Teknologi, 2025. Accessed: Apr. 29, 2026. [Online]. Available: https://www.google.co.id/books/edition/Mengenal_Algoritma_Random_Forest_Cara_Mu/d1SaEQAAQBAJ?hl=id&gbpv=0
- [8] Muhammad Syahputra Novelan and Solly Aryza, Belajar Mengenal Dasar Machine learning. Serasi Media Teknologi, 2025. Accessed: Apr. 29, 2026. [Online]. Available: https://www.google.co.id/books/edition/Belajar_Mengenal_Dasar_Machine_Learning/TeVOEQAAQBAJ?hl=id&gbpv=0
- [9] Deny Jollyta, Prihandoko, Alyauma Hajjah, Elin Haerani, and Muhammad Siddik, Algoritma Klasifikasi untuk Pemula Solusi Python dan RapidMiner. Deepublish, 2023. Accessed: Apr. 29, 2026. [Online]. Available: https://www.google.co.id/books/edition/Algoritma_Klasifikasi_untuk_Pemula_Solus/y84TEQAAQBAJ?hl=id&gbpv=0
- [10] S. K. M. K. Rahmadden, S. K. M. K. Denok Wulandari, M. Renova, A. M. Gilang Ramadhan, and R. Sari, Machine learning. Serasi Media Teknologi, 2024. Accessed: May 01, 2026. [Online]. Available: <https://books.google.co.id/books?id=owoOEQAAQBAJ>
- [11] A. R. Dikananda, I. Ali, Fathurrohman, R. A. Rinaldi, and Iin, “Genre E-Sport Gaming Tournament Classification Using Machine learning Technique Based On Decision tree, Naïve Bayes, And Random Forest Algorithm,” IOP Conf. Ser. Mater. Sci. Eng., vol. 1088, no. 1, p. 012037, Feb. 2021, doi: 10.1088/1757-899X/1088/1/012037.
- [12] O. Valenzuela et al., “The Explainability of Machine learning Algorithms for Victory Prediction in the Video Game Dota 2,” Computer Sciences & Mathematics Forum 2025, Vol. 11, Page 26, vol. 11, no. 1, p. 26, Aug. 2025, doi: 10.3390/CMSF2025011026.
- [13] J. Tyran and L. Chomatek, “Influence of Outliers in MOBA Games Winner Prediction,” Procedia Comput. Sci., vol. 192, pp. 1973–1981, Jan. 2021, doi: 10.1016/J.PROCS.2021.08.203.
- [14] K. Arik, “Machine learning Models on MOBA Gaming: League of Legends Winner Prediction,” Acta Infologica, vol. 7, no. 1, pp. 139–151, Jan. 2024, doi: 10.26650/ACIN.1180583.
- [15] “MPL Indonesia Season 16 - Liquipedia Mobile Legends: Bang Bang Wiki.” Accessed: May 21, 2026. [Online]. Available: https://liquipedia.net/mobilelegends/MPL/Indonesia/Season_16