e-ISSN : 2722-4368

Penggunaan Algoritma Gaussian Naïve Bayes & Decision Tree Untuk Klasifikasi Tingkat Kemenangan Pada Game Mobile Legends

Yoga Naufal Ray Putro 1*, Aidil Afriansyah 2*, Radhinka Bagaskara 3*
* Teknik Informatika, Institut Teknologi Sumatera
Email coresponden yoga.14117099@student.itera.ac.id

Abstrak. *Perkembangan teknologi dan internet membuat popularitas game online seperti Mobile Legends semakin meningkat. Namun dalam kompetisi seringkali pemain mengalami kekalahan karena berbagai faktor, antara lain skill pemain, strategi tim, dan pemilihan hero yang tepat. Pemilihan hero yang tepat sangat penting untuk meningkatkan peluang kemenangan. Oleh karena itu, Mobile Legends Professional League (MPL) menjadi fokus tim-tim kompetitif di seluruh dunia. Penelitian ini bertujuan untuk mengetahui klasifikasi kemenangan pada pertandingan MPL berdasarkan draft pick. Gaussian Naïve Bayes dan Decision Tree digunakan sebagai model algoritma klasifikasi dalam penelitian ini. Proses dalam penelitian ini meliputi pembersihan data, transformasi data (pelabelan), penanganan data yang tidak seimbang, penskalaan, pemisahan, dan hyperparameter. Tahap evaluasi menggunakan matriks konfusi, data korelasi, dan kurva AU-ROC. Hasil penelitian ini menunjukkan bahwa metode Decision Tree memiliki kinerja yang lebih baik dibandingkan Gaussian Naïve Bayes dalam mengklasifikasikan data menggunakan matriks konfusi. Analisis AUC (area under the receiver operating character curve) menunjukkan bahwa pohon keputusan memiliki kinerja yang lebih baik dibandingkan Gaussian naif Bayes dalam memprediksi data positif dan negatif. Hal ini ditunjukkan dengan nilai AUC yang lebih tinggi untuk Decision Tree yaitu sebesar 0,67 dibandingkan dengan Gaussian Naïve Bayes yaitu sebesar 0,48. Model klasifikasi dengan nilai AUC yang lebih tinggi dapat membedakan data positif dan negatif dengan lebih akurat. Pada penelitian ini Decision Tree memiliki nilai AUC yang lebih tinggi dibandingkan Gaussian Naïve Bayes sehingga Decision Tree dapat mengklasifikasikan data kemenangan dan kekalahan dengan lebih akurat.*

Kata Kunci : Mobile Legends, Classification, Gaussian Naïve Bayes, Decision Tree, Draft Pick.

Abstract. *The development of technology and the internet has increased the popularity of online games, such as Mobile Legends. However, in competitions, players often experience defeat due to various factors, including player skills, team strategies, and the right hero selection. The right hero selection is very important to increase the chances of winning. Therefore, the Mobile Legends Professional League (MPL) has become a focus for competitive teams around the world. This study aimed to determine the classification of victory in MPL matches based on draft pick. Gaussian Naïve Bayes and Decision Tree were used as classification algorithm models in this study. The process in this study included cleaning data, data transformation (labeling), handling imbalanced data, scaling, splitting, and hyperparameter. The evaluation stage used confusion matrix, correlation data, and AU-ROC curve. The results of this study showed that the Decision Tree method had better performance than Gaussian Naïve Bayes in classifying data using the confusion matrix. The AUC (area under the receiver operating characteristic curve) analysis showed that the decision tree had better performance than Gaussian naive Bayes in predicting positive and negative data. This is indicated by the higher AUC value for the Decision Tree, which is 0.67 compared to Gaussian Naïve Bayes which is 0.48. Classification models with higher AUC values can more accurately distinguish between positive and negative data. In this study, the Decision Tree had a higher AUC value than Gaussian Naïve Bayes so the Decision Tree could more accurately classify victory and defeat data.*

Keyword : Mobile Legends, Classification, Gaussian Naïve Bayes, Decision Tree, Draft Pick.

PENDAHULUAN

Teknologi informasi saat ini berkembang sangat cepat, di mana kebutuhan manusia terpenuhi untuk mempermudah pekerjaan, memenuhi kebutuhan hiburan bahkan memenuhi kebutuhan manusia dalam hal olahraga. Bermacam produk teknologi informasi dikembangkan sebagai inovasi yang

menambah fasilitas maupun fitur pada produk teknologi yang sudah ada. Era Revolusi industri generasi 4.0. yaitu era revolusi industri yang ditandai dengan kemajuan teknologi yang sangat pesat pada setiap komponen kehidupan. Salah satu teknologi yang perkembangannya sangat pesat yaitu telepon genggam atau handphone berkembang menjadi smartphone[1]. Teknologi seperti ini juga telah menjadi kebutuhan setiap orang terlebih bagi mahasiswa. Teknologi bagi mahasiswa bukan hanya untuk mencari dan mendapatkan informasi, tetapi juga digunakan untuk kepentingan hiburan semata seperti bermain game online [2].

Dibandingkan jumlah pengguna game, jumlah pengembang lokal ternyata cenderung lebih lambat, sehingga pengguna game di Indonesia masih banyak menikmati game buatan dari luar Indonesia [6]. Pesatnya perkembangan Mobile Legends ditunjukkan dengan jumlah pemain terdaftar yang mencapai 1 miliar [7]. Namun seiring berjalannya waktu, game online kini tidak hanya tersedia di komputer atau laptop saja akan tetapi juga tersedia di smartphone, sehingga memungkinkan pengguna untuk lebih banyak menghabiskan waktunya untuk bermain game online. Beberapa jenis genre game online dalam smartphone banyak ditawarkan oleh para pengembang, diantaranya adalah Massively Multiplayer Online Role Playing Game (MMORPG), Massively Multiplayer Online Real Time Strategy (MMORTS), Massively Multiplayer Online First Person Shooter (MMOFPS), dan Multiplayer Online Battle Arena (MOBA) [8].

Mobile Legends Bang-Bang yang telah menjadi cabang turnamen pada Asian Games dan juaranya pun berasal dari Indonesia. Game yang dirilis oleh perusahaan pembuat game, 'Moonton', pada tahun 2016 ini sukses memikat hati para gamers Indonesia baik yang amatir maupun profesional. Melihat minat yang sangat tinggi dari masyarakat Indonesia terhadap game keluarannya, sebagai bentuk apresiasi 'Moonton' tidak tanggung-tanggung hingga merilis dua hero asal Indonesia, yakni 'Gatotkaca' yang berasal dari tokoh pewayangan dan 'Kadita' yang mengambil figur legenda Penguasa Laut Selatan [9]. Pada Mobile Legends ini banyak mode permainannya, terdapat beberapa mode permainan dalam game Mobile Legend ini yaitu: Classic, Brawl, VS Ai, Custom, Ranked, dan yang baru adalah Magic Chess. Pada beberapa kasus dalam permainan Ranked Match sering banyak pemain yang mengeluh dikarenakan peringkat game mereka terus menurun yang diakibatkan oleh kekalahan dalam permainan mode Ranked [10]. Mobile Legends Bang Bang atau disingkat 'MLBB' sendiri telah memiliki lebih dari 100 hero yang dapat dimainkan sehingga membuat game dengan sistem 5 versus 5 ini menjadi sangat bervariasi. Selain hero, pemain juga dapat menyusun set item dan emblem yang mampu meningkatkan performa dari hero yang dipakai. Dari keberagaman variasi tersebut maka terdapat banyak pula kombinasi yang mampu membawa tim menuju kemenangan, maka dari itu dibutuhkanlah sebuah aplikasi pendukung untuk menentukan penggunaan kombinasi role hero agar memiliki win rate atau kemungkinan menang yang tinggi [9].

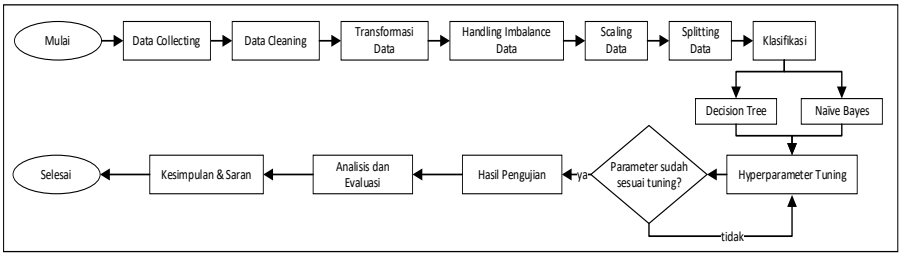
Beberapa kasus permainan Ranked Match banyak pemain yang mengeluh karena rank mereka terus turun akibat dari kekalahan dalam permainan. Beberapa faktor penyebab kekalahan dari permainan mereka. Contoh faktor permasalahan tersebut antara lain: penguasaan hero yang belum matang atau mencoba hero baru di ranked match, build item atau pemilihan equipment yang salah, koneksi jaringan yang kurang stabil namun dipaksa untuk bermain ranked match, dan susunan tim yang salah atau kurang optimal [13]. Kesulitan permainan justru meningkatkan pengguna karena developer game selalu memperbaharui tampilan, fitur dan menambahkan hero-hero yang baru, yang mana membuat pengguna tidak akan merasa bosan bahkan sebaliknya pengguna akan semakin sering memainkan permainan ini untuk mendapatkan hero-hero yang pemain inginkan [14]. Masalahnya kebanyakan pemain mengeluh karena ranked match mereka terus menurun akibat dari kekalahan dalam permainan. Berdasarkan kasus tersebut peneliti mendapatkan beberapa faktor penyebab kekalahan dari permainan mereka. Contoh permasalahan tersebut antara lain: tidak dapat memaksimalkan performa dari hero yang dimiliki, strategi kurang tepat, anggota tim banyak yang noob. Keberagaman variasi tersebut maka terdapat banyak kombinasi yang bisa membawa tim merebut kemenangan. Maka dari itu diperlukan suatu aplikasi pendukung untuk mengoptimalkan

susunan pemain agar memiliki kemungkinan menang yang tinggi, dan diharapkan dapat sebagai solusi bagi pemain pemula yang masih meraba-raba. Komposisi tim yang tepat atau untuk pengikut lomba, mereka dapat mengklasifikasi kemenangan dengan pemilihan hero dan susunan yang tepat.

Pohon klasifikasi dan pohon keputusan adalah teknik populer dalam data mining. Setelah proses klasifikasi, keputusan alami terbentuk, meskipun sering kali digunakan terpisah sebagai pohon klasifikasi atau pohon keputusan. Pohon keputusan digunakan untuk meramalkan kategori objek dengan mempertimbangkan nilai atribut mereka. Klasifikasi Gaussian Naïve Bayes yang didasarkan pada Teorema Bayes, telah menunjukkan kinerja yang sebanding dengan pohon keputusan dan beberapa jaringan saraf terpilih dalam studi perbandingan algoritma klasifikasi. Penggunaan klasifikasi Gaussian Naïve Bayes pada basis data besar juga telah terbukti memiliki akurasi dan kecepatan yang tinggi. Klasifikasi Gaussian Naïve Bayes mengoperasikan asumsi bahwa efek dari atribut pada suatu kelas adalah independen dari atribut lainnya, yang dikenal sebagai independen kondisional terhadap kelas. Asumsi ini dilakukan untuk mempermudah perhitungan, dan karena itu sering dianggap 'naif'[15]. Menggunakan kedua algoritma tersebut diharapkan penelitian ini nantinya akan mengurangi tingkat kekalahan tim dalam hal memenangkan permainan dan mampu memberikan informasi kepada anggota tim untuk memilih pahlawan yang tepat. Berdasarkan pada latar belakang tersebut, maka penelitian ini dilakukan sebagai klasifikasi kemenangan tim dalam game mobile legends dengan menggunakan algoritma Gaussian Naïve Bayes & Decision Tree.

METODOLOGI PENELITIAN

Konteks permainan *Mobile Legends*, terdapat permasalahan yaitu, sering kali terjadi pemilihan *hero* yang tidak terkoordinasi atau penerapan taktik yang tidak sesuai, yang dapat menyebabkan dampak merugikan terhadap kinerja tim dan pengalaman keseluruhan dalam pertandingan. Tujuan dari perbandingan antara algoritma *Gaussian Naïve Bayes* dan *Decision Tree* dalam konteks pemilihan *hero* pada permainan *Mobile Legends* adalah untuk mengevaluasi efektivitas kedua metode dalam merekomendasikan pilihan *hero* yang optimal berdasarkan komposisi tim lawan dan karakteristik *hero* yang ada. Dataset yang digunakan untuk melakukan penelitian penggunaan algoritma *Gaussian Naïve Bayes* dan *decision tree* untuk klasifikasi tingkat kemenangan pada *game mobile legends* adalah data yang didapat dengan melihat secara langsung hasil pertandingan pada turnamen nasional yaitu *MPL ID Season 11, M4 World Championship, & MPL PH Season 11* dengan pembaharuan *game* 1.7.68 pada 22 Maret 2023.



Gambar 1. Rancangan Model

1) Decision Tree

Dalam menentukan pohon keputusan langkah awal yang dilakukan yaitu menghitung nilai *entropy* sebagai penentu tingkat ketidakmurnian atribut dan nilai *information gain*. Menghitung nilai *entropy* dapat menggunakan rumus seperti sebagai berikut:

$$Entropy(S) = \sum_{i=1}^{n} p_i * \log_2 p_i$$

Menentukan nilai *information gain* dapat menggunakan rumus sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(s_i)$$

2) Gaussian Naïve Bayes

Gaussian Naïve Bayes adalah salah satu algoritma klasifikasi yang menggunakan data dengan target kelas atau label yang berupa nilai kategorial atau nominal. *Gaussian Naïve Bayes* merupakan salah satu *classifier* sederhana didasarkan pada *teorema bayes*[18]. *Gaussian Naïve Bayes* untuk klasifikasi adalah salah satu metode paling umum dan penting dalam menghitung probabilitas dan statistik. *Distribusi Gaussian* hasil sebagai berikut:

$$P(X_i) = \frac{1}{\delta\sqrt{2\pi}} exp^{-\frac{(X_i-\mu)^2}{2\delta^2}}$$

Dimana, μ adalah rata-rata dengan rumus sebagai berikut:

$$\mu = \frac{\sum_{i=1}^{n} xi}{n}$$

Dan δ adalah standar deviasi. Untuk mendapatkan nilai δ dapat menggunakan rumus berikut [16]:

$$\delta^2 = \frac{\sum_{i=1}^{n}(xi - \mu)^2}{n - 1}$$

Berikut metode yang digunakan untuk melakukan analisis dan evaluasi :

1) *Confusion Matrix* adalah alat ukur berbentuk *matrix* yang digunakan untuk mendapatkan jumlah ketepatan klasifikasi terhadap kelas dengan algoritma yang dipakai [26]. Secara visual, *confusion matrix* terlihat seperti Tabel 1 [26]:

Tabel 1. Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

Confusion Matrix menunjukkan nilai TP (*true positive*) dan TN (*true negative*), yang menggambarkan ketepatan klasifikasi. Semakin tinggi nilai TP dan TN, semakin baik tingkat *accuracy*, *precision*, dan *recall*. Label prediksi yang benar dan nilai sebenarnya yang salah disebut *false positive* (FP), sedangkan prediksi yang salah dan nilai sebenarnya yang *benar disebut false negative* (FN). Formulasi untuk menghitung *accuracy, precision, recall,* dan *f1-score* pada model klasifikasi pada rumus berikut:

$$accuracy(\%) = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$Precision(\%) = \frac{TP}{(TP + FN)}$$

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

$$F1 - Score(\%) = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

Confusion matrix membantu dalam mengevaluasi seberapa baik model melakukan prediksi yang tepat. Dengan informasi dari *confusion matrix*, kita dapat mengukur kinerja model menggunakan metrik seperti akurasi, presisi, recall, dan F1-score.

2) *Area Under the Receiver Operating Characteristic (AU-ROC)* adalah metrik evaluasi yang mengukur seberapa baik model klasifikasi dapat membedakan antara dua kelas yang berbeda. Ini mengukur area yang tercakup di bawah kurva ROC, yang mencerminkan kinerja keseluruhan model dalam mengidentifikasi dengan benar data positif dan negatif. Semakin besar nilai AU-ROC, semakin baik kemampuan model dalam membedakan antara kelas positif dan negatif. AU-ROC adalah nilai numerik antara 0 dan 1, di mana nilai 1 menunjukkan klasifikasi sempurna dan nilai 0.5 menunjukkan klasifikasi acak [19].

3) Data *Correlation*

Korelasi Data adalah salah satu metode statistik yang digunakan untuk mengukur sejauh mana ada hubungan atau korelasi antara dua variabel. Pada tahapan ini metode yang digunakan adalah '*Spearman'*, metode ini menggunakan peringkat data alih-alih nilai aktual. Dalam situasi di mana data tidak mematuhi distribusi normal *bivariat* atau memiliki sifat ordinal, koefisien korelasi *Spearman* sering kali dipilih sebagai metode yang lebih tepat untuk mengukur hubungan antara dua variable.

HASIL DAN PEMBAHASAN

Hasil penelitian yang telah dilakukan terdiri dari model *Gaussian Naïve Bayes, Decision Tree*, hasil pengujian *Gaussian Naïve Bayes, Decision Tree* dan *Confusion Matrix*, *Data Correlation*, serta kurva *AU-ROC*. Dataset yang digunakan terbagi menjadi 2 jenis dan beberapa konfigurasi *Hyperparameter* telah diubah untuk mendapatkan hasil deteksi objek yang optimal.

A. Data *Collection*

Pengumpulan *dataset* yang terdiri dari berbagai data terkait dengan pertandingan, pahlawan (hero), dan tim. *Dataset* ini mencakup beragam informasi yang relevan untuk analisis yang dilakukan. Data pertandingan mencakup detail tentang setiap pertandingan yang terjadi, seperti tanggal, waktu, tim, *mvp*, *pick* dan *ban*, dan hasilnya. *Dataset* yang digunakan dalam penelitian ini berjumlah 442 data mencakup informasi, penelitian ini bertujuan untuk menganalisis hubungan antara pahlawan, tim, dan hasil pertandingan.

B. Pembersihan Data (Data Cleaning)

Pada tahapan ini dataset yang yang dikumpulkan dari berbagai sumber diperiksa, diperbaiki, dan diolah agar siap digunakan dalam analisis lebih lanjut. Oleh karena itu, pembahasan proses dan hasil akan dijelaskan sebagai berikut:

1) Penanganan Nilai Hilang (*Check Missing Value*)

Pada tahapan ini, pengecekan dilakukan dengan teliti terhadap seluruh *dataset* yang telah terkumpul untuk memastikan integritasnya. Langkah awal yang diambil adalah untuk melakukan pengecekan apakah terdapat baris atau kolom yang memiliki nilai kosong atau hilang.

Tabel 2. Check Missing Value

Fitur	Nan value	Fitur	Nan value	Fitur	Nan value
date	0	pick_blue_1	0	pick_red_1	0
game	0	pick_blue_2	0	pick_red_2	0
blue	0	pick_blue_3	0	pick_red_3	0
red	0	pick_blue_4	0	pick_red_4	0
time(minutes)	0	pick_blue_5	0	pick_red_5	0
win	0	ban_blue_1	0	ban_blue_1	0

win_team	0	ban_blue_2	0	ban_blue_2	0
mvp	0	ban_blue_3	0	ban_blue_3	0
		ban_blue_4	0	ban_red_4	0
		ban_blue_5	0	ban_red_5	0

Hasil pemeriksaan menunjukkan keberadaan data kosong dalam dataset. Setiap baris dalam tabel mewakili fitur dalam dataset, dan kolom pertama menunjukkan jumlah data yang kosong (NaN) untuk setiap fitur. Dari hasil pemeriksaan ini, dapat dilihat bahwa tidak ada data yang hilang dalam setiap fitur, ditandai dengan nilai "0" dalam kolom 'Nan value'. Ini menunjukkan bahwa *dataset* yang digunakan dalam penelitian telah diisi dengan data lengkap untuk semua fitur.

2) Pengecekan Duplikat Data

Pemeriksaan ini menjadi esensial untuk memastikan integritas dan keandalan *dataset* yang digunakan dalam analisis lebih lanjut.

Tabel 3. Duplikat Data

Columns	Index	Columns	Index	Columns	Index
date	0	pick_blue_1	0	pick_red_1	0
game	0	pick_blue_2	0	pick_red_2	0
blue	0	pick_blue_3	0	pick_red_3	0
red	0	pick_blue_4	0	pick_red_4	0
time(minutes)	0	pick_blue_5	0	pick_red_5	0
win	0	ban_blue_1	0	ban_red_1	0
win_team	0	ban_blue_2	0	ban_red_2	0
mvp	0	ban_blue_3	0	ban_red_3	0
		ban_blue_4	0	ban_red_4	0
		ban_blue_5	0	ban_red_5	0

Hasil dari pemeriksaan duplikasi data menunjukkan bahwa dalam dataset yang sedang dianalisis, tidak ada baris yang memiliki duplikasi. Setiap entri dalam dataset tersebut unik dan tidak ada data yang tampil lebih dari satu kali. Kondisi ini menunjukkan integritas data yang baik, memungkinkan analisis yang lebih akurat dan andal.

3) Penghapusan Baris atau Kolom

Tahapan penyempurnaan *dataset*, langkah penting yang diambil selanjutnya adalah penghapusan kolom. Sebelum melakukan itu berikut ini akan ditampilkan kolom (fitur) sebelum dilakukan penghapusan.

Tabel 4. Sebelum Drop Kolom

Column	Column	Column	Column
date	mvp	ban_blue_2	pick_red_4
game	pick_blue_1	ban_blue_3	pick_red_5
blue	pick_blue_2	ban_blue_4	ban_red_1
red	pick_blue_3	ban_blue_5	ban_red_2
time(minutes)	pick_blue_4	pick_red_1	ban_red_3
win	pick_blue_5	pick_red_2	ban_red_4
win_team	ban_blue_1	pick_red_3	ban_red_5

Setelah mengidentifikasi kolom yang tidak relevan, langkah selanjutnya adalah menjatuhkan kolom tersebut agar dataset hanya berisi atribut yang relevan dan berkontribusi pada analisis. Tujuannya adalah menciptakan dataset yang lebih fokus dan ringkas.

Tabel 5. Setelah Drop Kolom

Column	Column
win_team	pick_red_1
pick_blue_1	pick_red_2
pick_blue_2	pick_red_3
pick_blue_3	pick_red_4
pick_blue_4	pick_red_5
pick_blue_5	

Hasil dari pemeriksaan duplikasi data menunjukkan bahwa dalam *dataset* yang sedang dianalisis, tidak ada baris yang memiliki duplikasi. Setiap entri dalam *dataset* tersebut unik dan tidak ada data yang tampil lebih dari satu kali. Kondisi ini menunjukkan integritas data yang baik, memungkinkan analisis yang lebih akurat dan handal. Penyertaan data yang tepat akan melibatkan pemilihan pahlawan oleh kedua tim (tim biru dan tim merah) dan hasil akhir pertandingan yang menunjukkan apakah tim tersebut menang atau kalah. Ini akan mempermudah analisis terkait dengan pemilihan pahlawan dan dampaknya terhadap hasil pertandingan, serta membantu dalam pengembangan model atau algoritma klasifikasi berdasarkan data tersebut. Dengan kata lain, proses pembersihan data ini telah menyederhanakan data untuk memungkinkan analisis yang lebih efisien dan relevan.

C. Transformasi Data

1) Binary-label Encoder

Pada tahapan ini mengubah nilai dalam kolom *'win_team'* berdasarkan kondisi. Dimana pada kolom *'win_team'* ini mengacu kepada kolom sebelumnya yaitu '*blue'* dan '*red'*.

Tabel 6. Daftar Tim

Team	Index	Team	Index
BLCK	0	ONIC_PH	11
BREN	1	OT	12
BURN	2	RRQ	13
ECHO	3	RRQA	14
FCON	4	RSG	15
INC	5	RSG_PH	16
MDH	6	S11	17
MVG	7	TDK	18
NXPE	8	THQ	19
OMG	9	TNC	20
ONIC	10	TV	21

Setelah itu, nama tim tersebut diubah berdasarkan kolom kemenangan yang berhubungan dengan kolom 'biru' dan 'merah'. Apabila nilai dalam kolom 'biru' sama dengan nilai dalam kolom 'kemenangan', maka nilai 'tim_pemenang' diubah menjadi 0; jika tidak, diubah menjadi 1. Berikut adalah hasil *one-hot encoding*:

Tabel 7. Binary-label Encoder

	win_team		
0	0	437	1
1	0	438	0
2	0	439	0
3	0	440	1
4	0	441	1
…	…	437	1

Transformasi ini dilakukan dengan memberikan representasi biner, yaitu 0 atau 1, yang mengindikasikan tim pemenang berdasarkan perbandingan antara kolom *'blue'* dan *'win'*. Ketika tim yang terdapat dalam kolom *'blue'* dan kolom *'win'* sama, maka nilai pada kolom *'win_team'* diubah menjadi '1'. Sebaliknya, jika tidak sama, nilai pada kolom *'win_team'* akan diubah menjadi '0'.

2) Multi-label Encoder

Kemudian, langkah selanjutnya yang dilakukan adalah penerapan teknik *Multi-label encoder* pada setiap kolom '*pick'* dalam *dataset*. Teknik ini memiliki peran penting dalam mengubah label *multi-label* ke dalam format yang dapat dengan mudah dipahami dan diolah oleh algoritma pembelajaran mesin.

Tabel 8. Daftar Sample Hero (Pahlawan)

hero	indeks	hero	indeks	hero	indeks	hero	indeks
aamon	0	alpha	4	arlott	8	badang	12
akai	1	alucard	5	atlas	9	balmond	13
aldous	2	angela	6	aulus	10	bane	14
alice	3	argus	7	aurora	11	barats	16

Tabel ini menggambarkan 5 data awal dan 5 data akhir, di mana dapat dilihat indeks untuk 9 yaitu adalah 'atlas', dan indeks ke 8 yaitu '*arlott'*. Tujuan dengan adanya perubahan *multi-label* ini adalah agar *dataset* menjadi format yang lebih mudah diakses dan diinterpretasikan oleh algoritma pembelajaran mesin yang digunakan dalam penelitian ini.

Tabel 9. Multi-label Encoder

pick_blue_1	pick_blue_2	pick_blue_3	pick_blue_4	pick_blue_5	pick_red_1	pick_red_2	pick_red_3	pick_red_4	pick_red_5
19	107	61	65	63	115	46	76	17	9
42	39	116	65	75	70	74	36	26	25
66	39	92	110	33	59	74	36	56	75
70	52	107	94	63	105	39	92	65	25
39	35	116	65	81	70	74	36	50	45
...	...	...	...	…	…	…	…	…	…
70	69	92	17	75	42	35	107	110	63
42	69	61	82	54	70	79	92	86	29
8	45	107	82	61	70	79	92	17	63
8	69	107	17	45	39	52	92	50	25
105	52	116	17	68	91	16	61	26	63

Hasil dari proses *Multi-label encoding* pada *dataset* menciptakan representasi baru untuk setiap kolom *draft* (pemilihan *hero*) dalam *dataset*. Hasil ini merupakan perubahan dari *multi-label* menjadi kode numerik yang sesuai. Setiap angka dalam tabel mewakili suatu *hero* yang dipilih dalam *draft*. Dengan representasi ini, algoritma pembelajaran mesin dapat lebih mudah mengolah dan memahami data *pick* dalam konteks permainan. Hal ini bertujuan untuk mempermudah penggunaan data oleh algoritma pembelajaran mesin dalam melakukan analisis dan klasifikasi terkait permainan *Mobile Legends*.

D. Penanganan Data Tidak Seimbang *(Balancing Data)*

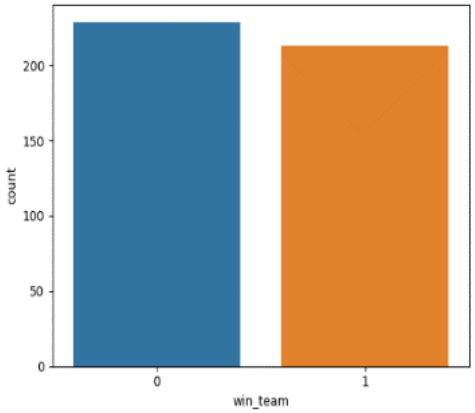
Setelah melakukan transformasi data, langkah selanjutnya adalah menyeimbangkan data. Label data dalam penelitian ini terbagi menjadi dua kategori, yaitu 'biru' dan 'merah'. Selanjutnya, jumlah data 'tim_pemenang' dihitung untuk kedua kategori. Pada tahap ini, teknik yang diterapkan untuk menangani keseimbangan data adalah "pengurangan sampel" atau *undersampling*. Dalam proses

ini, peneliti mengurangi jumlah sampel dari kelas mayoritas, yaitu 'biru' dengan total data 229 dan 'merah' dengan total data 213, untuk menyesuaikannya dengan jumlah data pada kelas minoritas. Hal ini dilakukan untuk mengurangi ketimpangan antara kelas mayoritas dan kelas minoritas.

Tabel 10. Sebelum Balancing Data

win_team	count
0	229
1	213

setelah diketahui adanya data yang tidak seimbang, yaitu '*blue'*, kemudia divisualisasikan dengan diagram batang pada gambar berikut:



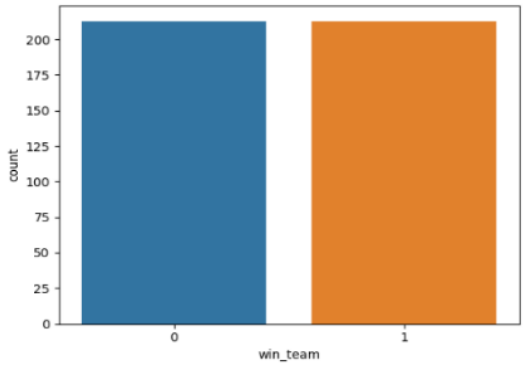
Gambar 2. Diagram Batang sebelum Balancing Data

Selanjutnya setelah melihat adanya ke tidak setimbangan data antara 'blue_tim' dan 'red_tim' pada gambar di atas terlihat bahwasanya pada 'blue_tim' terdapat 229 data dan 'red_tim' terdapat 213 data. Setelah dilakukan proses undersampling berikut didapatkan bahwasanya antara data 'blue_tim' dan 'red_tim' menjadi seimbang, terlihat seperti berikut:

Tabel 11. Setelah Balancing Data

win_team	count
0	213
1	213

Selanjutnya, sebagai langkah untuk mengkaji sejauh mana keseimbangan data yang ada, peneliti memutuskan untuk memberikan visualisasi yang lebih jelas melalui penggunaan diagram batang berikut:



Gambar 3. Diagram Batang setelah Balancing Data

Pada tahap akhir, hasil penyeimbangan dicetak dan divisualisasikan melalui diagram batang, memberikan pandangan visual tentang efek dari transformasi yang telah dilakukan. Pada Gambar 4 di atas dijelaskan *dataset* berhasil disesuaikan untuk mengatasi ketidakseimbangan kelas target dengan menerapkan teknik *undersampling*. Di mana ini menghasilkan distribusi kelas yang lebih merata untuk '*blue_tim'* dan '*red_tim'* menjadi 213 data dan siap untuk analisis lebih lanjut. Dengan demikian, *dataset* telah diolah dengan sukses untuk keperluan analisis lebih lanjut atau pengembangan model.

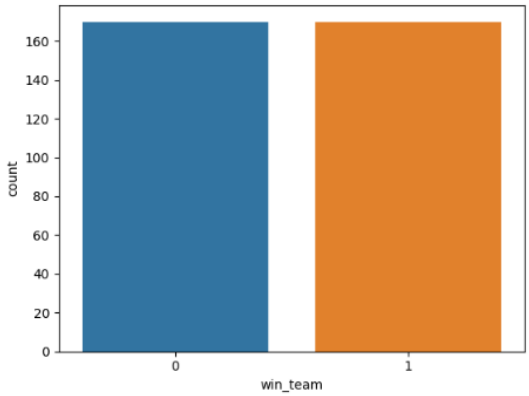
E. Pembagian Data (Splitting Data)

Tahapan berikutnya adalah mengoptimalkan kinerja model pada data baru dengan proses splitting data. Dataset awal berisi 442 data, kemudian dibagi menjadi dua subset dengan perbandingan 80% untuk data training (340 data) dan 20% untuk data testing (86 data). Langkah-langkah pembagian data training dan testing dijelaskan pada Tabel 12.

Tabel 12. Splitting Data Training

win_team	count
0	170
1	170

Pada Tabel 11 didapatkan hasil *splitting* data untuk data *training* dengan jumlah rata 170 data untuk kedua tim. Berikut akan divisualisasikan hasil *splitting* data *train* dengan menggunakan *bar chart* pada Gambar 4:



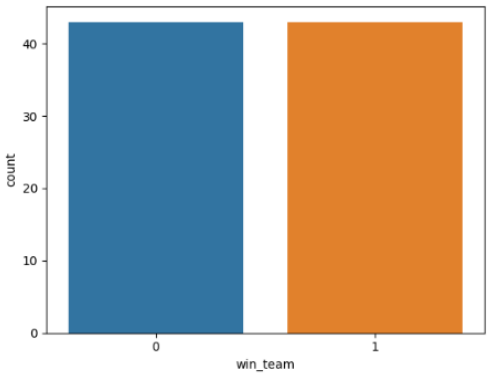
Gambar 4. Diagram Batang Splitting Data Training

Selanjutnya setelah data terbagi untuk data *training,* Langkah selanjutnya membagi data untuk data *testing* dengan perbandingan datanya 20% dari data *training,* untuk lebih jelasnya dapat dilihat pada Tabel 13:

Tabel 13. Splitting Data Testing

win_team	count
0	43
1	43

Setelah melihat *splitting* data untuk data *testing* dengan total untuk kedua tim yaitu 43 data, peneliti memvisualisasikannya pada Gambar 5:



Gambar 5. Diagram Batang Spliting Data Testing

Melihat pembagian seperti yang ditunjukkan dalam gambar di atas, terdapat 43 data untuk setiap tim dalam data *testing*, dengan total 86 data untuk data *testing*, dan 170 data untuk setiap tim dalam data training, dengan total 340 data untuk data training. Pembagian ini dilakukan karena peneliti memisahkan total 442 data menjadi data *testing* dan *training*, dengan perbandingan 80% untuk data *training* dan 20% untuk data testing. Proses ini menghasilkan empat subset, yaitu `X_train` dan `Y_train` untuk melatih model, serta `X_test` dan `Y_test` untuk menguji performa model pada data yang belum pernah terlihat sebelumnya. Dengan membagi *dataset* menjadi sub-set pelatihan dan pengujian, data memiliki landasan yang kuat untuk pengembangan dan pengujian model.

F. Pengskalaan Data (*Scaling Data*)

Langkah berikutnya adalah melakukan penyesuaian skala data, dikenal sebagai *Scaling* data. Pada tahap ini menerapkan teknik *MaxAbsScaler* untuk mempertahankan proporsi relatif antara pilihan hero dalam *draft* tanpa mengubah rentang hero yang terpilih. Hasil dari proses ini terdokumentasi dalam Tabel 14 berikut:

Tabel 14. Scalling Data

	pick_blue_1	pick_blue_2	pick_blue_3	pick_blue_4	pick_blue_5	pick_red_1	pick_red_2	pick_red_3	pick_red_4	pick_red_5
1	0.102804	0.946903	0.408602	0.526316	0.539130	1.000000	0.414414	0.652174	0.000000	0.069565
2	0.317757	0.345133	1.000000	0.526316	0.643478	0.605263	0.666667	0.304348	0.089109	0.208696
3	0.542056	0.345133	0.741935	1.000000	0.278261	0.508772	0.666667	0.304348	0.386139	0.643478
4	0.579439	0.460177	0.903226	0.831579	0.539130	0.912281	0.351351	0.791304	0.475248	0.208696
5	0.289720	0.309735	1.000000	0.526316	0.695652	0.605263	0.666667	0.304348	0.326733	0.382609
…	…	…	…	…	…	…	…	…	…	…
421	0.579439	0.610619	0.741935	0.021053	0.643478	0.359649	0.315315	0.921739	0.920792	0.539130
422	0.317757	0.610619	0.408602	0.705263	0.460870	0.605263	0.711712	0.791304	0.683168	0.243478
423	0.000000	0.398230	0.903226	0.705263	0.521739	0.605263	0.711712	0.791304	0.000000	0.539130
424	0.000000	0.610619	0.903226	0.021053	0.382609	0.333333	0.468468	0.791304	0.326733	0.208696
425	0.906542	0.460177	1.000000	0.021053	0.582609	0.789474	0.144144	0.521739	0.089109	0.539130

Penggunaan *MaxAbsScaler* pada *dataset* tersebut dilakukan dengan tujuan untuk menjaga proporsi relatif antara pilihan *hero* dalam *draft*, tanpa mengubah rentang nilai dari pilihan *hero* yang telah dipilih. *MaxAbsScaler* memungkinkan penyesuaian skala data tanpa menggeser distribusi nilai. Dengan kata lain, teknik ini mengukur relatif setiap nilai terhadap nilai tertinggi dalam *dataset*.

G. *Hyperparameter Tuning*

Pengoptimalkan parameter-parameter kunci dengan menggunakan 'Parfit' algoritma pembelajaran mesin demi mencapai hasil klasifikasi yang lebih baik dan akurat. Parameter yang didapat setelah proses *hyperparameter tuning* dapat dilihat pada Tabel 15 berikut:

Tabel 15. Hyperparameter Tuning Parfit

Algoritma	Parameter	Best Parameter	Best Accuracy
Gaussian Naïve Bayes	'var_smoothing':[1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10],	(var_smoothing=0.001)	0.4883720930232558
Decision Tree	'criterion':["gini","entropy"],'random_state':[1,2,10,20,30,42], 'max_depth': [1,5,10,15,20,25], 'splitter':['random','best'], 'max_features': ['auto', 'sqrt', 'log2'],	criterion='entropy', max_depth=15, max_features='auto', random_state=1, splitter='random'	0.6744186046511628

Melalui *hyperparameter tuning*, *Gaussian Naïve Bayes* memperbaiki akurasi dengan mengatur *'var_smoothing'* ke 0.1, mencapai akurasi 0.4651. *Decision Tree* juga meningkatkan akurasi, dengan parameter terbaik 'max_depth= 10', *'max_features'='auto'*, dan *'random_state=2'*, mencapai akurasi 0.6512.

H. Hasil Pengujian

Tahapan ini langsung mengarahkan fokus pada hasil pengujian model setelah proses tuning *hyperparameter*. Meskipun tuning *hyperparameter* menggunakan metode 'parfit' telah dilakukan, belum tentu parameter terbaik dan skor terbaik langsung diperoleh. Oleh karena itu, langkah berikutnya adalah melatih model dengan parameter yang dihasilkan dari proses tuning tersebut pada Tabel 16 berikut:

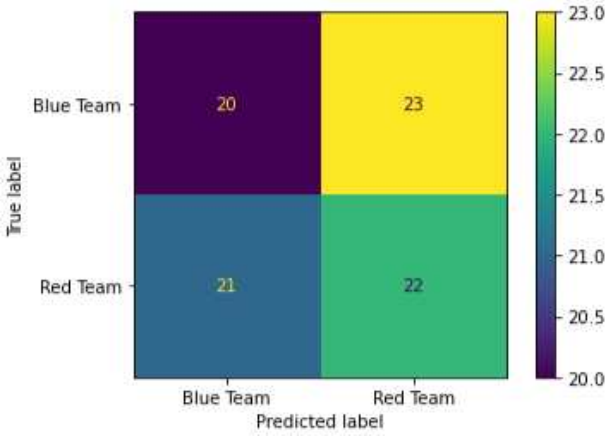
Tabel 16. Pengujian Model

Algoritma	Best Parameter	Best Accuracy	Accuracy
Gaussian Naïve Bayes	(var_smoothing=0.001)	0.4883720930232558	0.49
Decision Tree	criterion='entropy', max_depth=15, max_features='auto',random_state=1, splitter='random'	0.6744186046511628	0.67

Hasil menunjukkan bahwa akurasi dari kedua algoritma tidak mengalami penurunan. Pada algoritma *Gaussian Naïve Bayes* akurasi awal 0.4883 meningkat menjadi 0.49 setelah mengoptimalkan parameter *'var_smoothing'*='0.001'.

1) Hasil Pengujian *Gaussian Naïve Bayes*

Hasil pengujian analisis tingkat akurasi klasifikasi yang diperoleh dari algoritma *Gaussian Naïve Bayes*



Gambar 6. Grafik Confusion Matrix Gaussian Naive Bayes

Berikut perhitungan mencari nilai *accuracy* dari setiap label :

$Accuracy = \frac{(20+22)}{(20+23+22+21)} = 0.488372093023 \sim 49\%$

Selanjutnya untuk perhitungan nilai *precision* dari setiap label :

$Precision\ Blue\ Team = \frac{21}{(21+23)} = 0.488372093023 \sim 49\%$

$Precision\ Red\ Team = \frac{22}{(22+20)} = 0.4888888888889 \sim 49\%$ Perhitungan untuk nilai *recall* dari setiap label:

$Recall\ Blue\ Team = \frac{20}{(20+23)} = 0.4651 \sim 47\%\ Recall\ Red\ Team = \frac{22}{(22+21)} = 0.5116279069767 \sim 51\%$

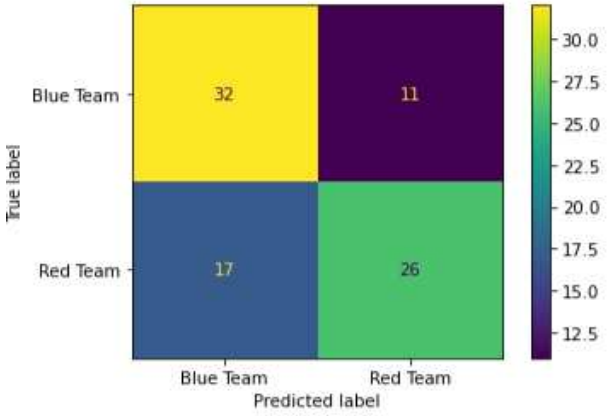
Setelah melakukan perhitungan *recall,* selanjutnya adalah perhitungan *f1-score :*

$F1 - Score\ Blue\ Team = 2 * \frac{(0.47*0.49)}{0.47+0.49} 0.4797966 \sim 48\%$

$F1 - Score\ Red\ Team = 2 * \frac{(0.51*0.49)}{0.51+0.49} = 0.4998 \sim 0.50\%$

2) Hasil Pengujian *Decision Tree*

Hasil pengujian memberikan wawasan yang mendalam tentang keberhasilan model dalam mengklasifikasikan hasil pertandingan *Mobile Legends*.



Gambar 7. Grafik Confusion Matrix Decision Tree

Berikut perhitungan nilai *accuracy* :

$Accuracy = \frac{(32+26)}{(32+17+26+11)} = 0.674418 \sim 67\%$

Selanjutnya untuk perhitungan nilai precision:

$Precision\ Blue\ Team = \frac{32}{(32+17)} = 0.65306122 \sim 65\%\ Precision\ Red\ Team = \frac{26}{(26+11)} = 0.702702702 \sim 70\%$

Setelah itu lakukan perhitungan untuk nilai *recall*:

$Recall\ Blue\ Team = \frac{32}{(32+11)} = 0.744186046 \sim 74\%\ Recall\ Red\ Team = \frac{26}{(26+17)} = 0.6046511627 \sim 60\%$

Setelah melakukan perhitungan *recall,* selanjutnya adalah perhitungan *f1-score :*

$$F1 - Score\ Blue\ Team = 2 * \frac{(0.74*0.65)}{0.74+0.65} = 0.692086 \sim 70\%$$

$$F1 - Score\ Red\ Team = 2 * \frac{(0.60*0.70)}{(0.60+0.70)} = 0.646153 \sim 65\%$$

I. Analisis dan Evaluasi

Pada tahapan ini, kinerja dari kedua model dianalisis dengan menggunakan tiga metode penting, yaitu *confusion matrix*, data *correlation* dan kurva *AU-ROC*. *Confusion matrix* digunakan untuk mengukur tingkat *accuracy, precision, recall*, dan *F1-score* dalam klasifikasi hasil pertandingan, termasuk identifikasi tim pemenang. Selanjutnya data *correlation* proses untuk memahami hubungan antara dua atau lebih variabel. Sementara itu, kurva *AU-ROC* memberikan pemahaman tentang kemampuan model dalam memisahkan hasil kemenangan dan kekalahan. Dengan menggunakan perbandingan data latih dan data uji 80%:20%.

1) *Confusion Matrix*

Pada Tabel 17 akan ditampilkan keseluruhan dari hasil *confusion matrix* :

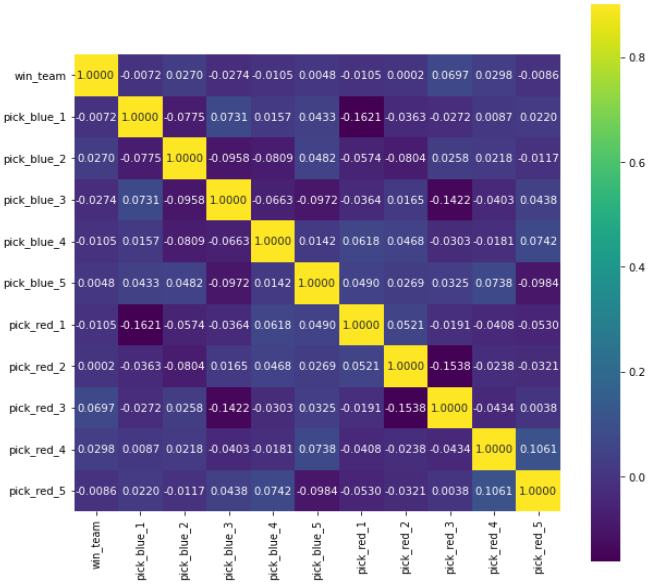
Tabel 17. Confusion Matrix

Metode	klasifikasi	presisi	recall	f1-score	akurasi
Gaussian Naïve Bayes	0	0.49	0.47	0.48	0.49
	1	0.49	0.51	0.50	
Decision Tree	0	0.65	0.74	0.70	0.67
	1	0.70	0.60	0.65	

Berdasarkan analisis *confusion matrix*, model *Decision Tree* memiliki kinerja yang baik dalam mengklasifikasikan data. Hal ini didukung oleh beberapa faktor, yaitu data yang digunakan mengikuti distribusi normal, fitur yang dipilih untuk model *Decision Tree* adalah fitur yang relevan dan memiliki korelasi yang kuat dengan label, serta *hyperparameter* untuk model *Decision Tree* telah dioptimalkan dengan baik.

2) Data Correlation

Adapun evaluasi lainnya yaitu menggunakan *correlation* dengan metode *'Spearman'*, hal ini berhasil divisualisasikan ke dalam bentuk matriks. Hasil penggunaan *correlation* dari metode *'Spearman'* dapat dilihat pada Gambar 8 :



Gambar 8. Data Correlation Spearman

Gambar 9, disajikan bahwa terdapat korelasi antara dua variabel, yakni jumlah *pick blue* dan jumlah *pick red*. Koefisien korelasi *'Spearman'* pada fitur *pick_red_1* dengan *pick_blue_1* yang bernilai -0,1621, mengindikasikan adanya hubungan negatif yang signifikan antara kedua variabel tersebut. Artinya, ketika jumlah *pick_blue* mengalami peningkatan, jumlah *pick_red* cenderung mengalami penurunan. Hal ini membuat *kernel* yang dihasilkan membuat akurasi yang rendah untuk klasifikasi jumlah *pick_blue* dan *pick_red*.

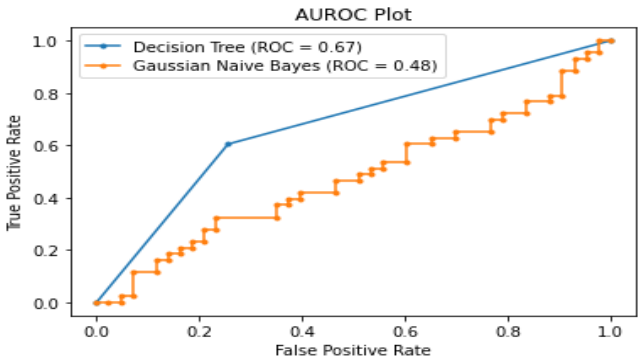
Akurasi yang rendah pada kedua algoritma, *Gaussian Naïve Bayes* dan *Decision Tree*, dapat disebabkan oleh beberapa faktor yang perlu dipertimbangkan secara terperinci, yaitu:

a. Gaussian Naïve Bayes:
   1. Independensi Asumsi: *Gaussian Naïve Bayes* mengasumsikan bahwa semua fitur dalam *dataset* adalah independen satu sama lain, yang berarti bahwa nilai satu fitur tidak tergantung pada nilai fitur lainnya. Jika asumsi ini tidak memenuhi kenyataan, seperti dalam kasus di mana fitur-fitur saling tergantung, maka model akan memberikan estimasi yang kurang akurat.
   2. Data yang Tidak Sesuai dengan Asumsi: Jika data pelatihan tidak sesuai dengan distribusi yang diharapkan oleh *Gaussian Naïve Bayes*

b. Decision Tree:
   1. *Overfitting* atau *Underfitting*: *Decision Tree* cenderung rentan terhadap *overfitting* (terlalu cocok dengan data pelatihan) atau *underfitting* (tidak cukup cocok dengan data pelatihan).
   2. Sensitif terhadap Variasi Data Pelatihan: *Decision Tree* sangat sensitif terhadap variasi dalam data pelatihan. Jika data pelatihan berubah sedikit saja, atau jika pembagian simpul daun (*leaf node*) pada pohon keputusan tidak optimal, hal ini dapat memengaruhi kinerja model secara signifikan.
   3. Parameter Tuning yang Tidak Optimal: Pemilihan parameter yang tidak tepat, seperti kedalaman maksimum pohon atau jumlah sampel minimum untuk membagi simpul, dapat mempengaruhi kinerja model. Jika parameter-parameter ini tidak disesuaikan dengan baik dengan karakteristik data, akurasi model dapat menurun.

3) Kurva *AU-ROC*

Tingkat akurasi dievaluasi menggunakan kurva Area Under The Receiver Operating Characteristic (AU-ROC) dan hasilnya divisualisasikan dalam grafik. Kurva AU-ROC memberikan gambaran visual tentang kinerja berbagai model klasifikasi dengan membandingkan area di bawah kurva (AUC), membantu pemilihan model terbaik berdasarkan kualitas klasifikasi.



Gambar 9. Kurva *Plot AU-ROC*

Pada Gambar 10, pada kurva *ROC* yang diberikan, terdapat dua garis, yaitu garis *Decision Tree* (*ROC* = 0.67) dan garis *Gaussian Naive Bayes* (*ROC* = 0.48). Garis *Decision Tree* berada di atas garis *Gaussian Naive Bayes*, yang berarti bahwa *Decision Tree* memiliki kinerja yang lebih baik

dalam memprediksi data positif dan negatif. Nilai *AUC* dari *Decision Tree* adalah 0.67, sedangkan nilai *AUC* dari *Gaussian Naive Bayes* adalah 0.48. Nilai *AUC* yang lebih tinggi menunjukkan bahwa model klasifikasi tersebut memiliki kinerja yang lebih baik dalam memprediksi data positif dan negatif.

Berdasarkan hasil penelitian yang dilakukan tersebut, terdapat dua faktor utama yang dapat mempengaruhi perbedaan nilai *AUC* antara *Decision Tree* dan *Gaussian Naive Bayes*:

a. Kompleksitas model:
   Decision Tree adalah model yang lebih kompleks dibandingkan dengan *Gaussian Naive Bayes. Decision Tree* dapat menggambarkan hubungan yang lebih kompleks antara fitur-fitur, sehingga dapat menghasilkan model yang lebih akurat.
b. Ketersediaan fitur: Terdapat 10 fitur yang digunakan untuk memprediksi kemenangan pertandingan *Mobile Legends* berdasarkan *draft pick*. *Decision Tree* dapat menggunakan semua 10 fitur tersebut untuk membuat keputusan, sedangkan *Gaussian Naive Bayes* hanya dapat menggunakan 6 fitur karena mengasumsikan distribusi normal pada fitur-fitur. *Gaussian Naive Bayes* perlu mengetahui nilai rata-rata (μ) dan standar deviasi (σ) dari setiap fitur. Jika terdapat lebih dari 6 fitur, maka akan diperlukan banyak data untuk menghitung nilai μ dan σ dari setiap fitur. Hal ini dapat menyebabkan *underfitting*, yaitu model yang tidak dapat menangkap hubungan yang kompleks antara fitur-fitur.
c. Selain itu untuk *hero* yang terbanyak digunakan adalah '*fredrinn'* dikarenakan *hero* tersebut dapat digunakan pada 2 posisi (*role)* yaitu sebagai *EXPLaner* dan *Jungler*. Dengan total *pick* 274 untuk *total_pick_blue berjumlah* 155 dan *total_pick_red* berjumlah 119.

KESIMPULAN

Analisis klasifikasi kemenangan dalam Mobile Legends Professional League menggunakan algoritma Gaussian Naïve Bayes dan Decision Tree berdasarkan draft pick menunjukkan potensi penerapan fitur draft pick untuk prediksi hasil pertandingan, meskipun perlu penelitian lebih lanjut untuk meningkatkan performa model dan mempertimbangkan faktor-faktor lain yang memengaruhi hasil pertandingan. Hasil eksperimen menunjukkan bahwa Decision Tree mengungguli Gaussian Naïve Bayes dengan akurasi 0.67 dan 0.49 secara berturut-turut. Decision Tree mampu menggambarkan hubungan kompleks antara variabel dan hasil, serta menangani variasi distribusi data yang lebih adaptif, berkontribusi dalam pemahaman lebih dalam tentang faktor-faktor yang memengaruhi hasil pertandingan Mobile Legends Professional League.

DAFTAR PUSTAKA

[1] M. Huda, "Analisis User Experience Pada Game Mobile Legend Versi 1.4.14.4454 Dengan Menggunakan Game-Design Factor Questionnaire," Jurnal Ekonomi Dan Teknik Informatika, vol. 8, p. 25, 2020.

[2] C. V. Wijaya and S. Paramita, "Komunikasi Virtual dalam Game Online (Studi Kasus dalam Game Mobile Legends)," Koneksi, vol. 3, p. 262, 2019.

[3] R. Fieter, D. Ritzky, and K. M. R. Brahmana, "Pengaruh Online Experience Terhadap Loyalty Melalui Satisfaction Pemain Mobile Legends," Agora, vol. 6, no. 2, 2018.

[4] A. Bahtiar, R. R. Muhima, D. A. Rachman, I. T. Adhi, and T. Surabaya, "Penerapan Model Spiral Pada Rancang Bangun Game Platformer," in Seminar Nasional Sains dan Teknologi Terapan VII, 2019, p. 601.

[5] Tonggi Simanjuntak, "Jumlah Pemain Game Mobile Legends di 2023 Melonjak Drastis!," revivaltv.id. Accessed: Jan. 14, 2024. [Online]. Available: https://revivaltv.id/news/mlbb/jumlah-pemain-game-mobile-legends-2023

[6] A. Haris, D. C. Anggraini, and D. Mardiana, "Pengaruh Game Online Terhadap Ketaatan Beribadah Mahasiswa Di Jurusan Pendidikan Agama Islam Universitas Muhammadiyah Malang," Jurnal Visi Ilmu Pendidikan, vol. 13, no. 2, p. 99, Sep. 2021, doi: 10.26418/jvip.v13i2.43475.

[7] S. M. Listijo, T. Purwani, S. T. Galih, and T. Hafidzin, "Prediksi Kemenangan Dan Susunan Tim Pada Game Mobile Legends Bang Bang Menggunakan Algoritma Naïve Bayes," Komputaki , vol. 6, no. 1, pp. 15–17, 2020.

[8] A. T. Susilo, H. Setiawan, R. A. Saputro, T. Purwadi, and A. Saifudin, "Penggunaan Metode Naïve Bayes untuk Memprediksi Tingkat Kemenangan pada Game Mobile Legends," Jurnal Teknologi Sistem Informasi dan Aplikasi, vol. 4, no. 1, p. 46, Jan. 2021, doi: 10.32493/jtsi.v4i1.7807.

[9] F. Saputra, F. Dwikotjo, and S. Sumantyo, "Pengaruh Sistem Informasi Manajemen: Kepuasan Konsumen dan Keputusan Pembelian Tiket MPL Mobile Legend di Aplikasi Blibli.com," Jurnal Kewirausahaan dan Manajemen Bisnis, vol. 1, no. 2, 2023.

[10] Alfa Rizki, "Berapa prize pool MPL ID S12?," oneesports. Accessed: Oct. 10, 2023. [Online]. Available: https://www.oneesports.id/mobile-legends/prize-pool-mpl-id-s12/

[11] A. C. Putro, "Sistem Prediksi Kemenangan Tim Pada Game Mobile Legends Dengan Metode Naive Bayes Mobile Legends Win Prediction Using Naive Bayes," Untag, Pp. 1–2, 2018.

[12] A. I. Marpaung, "Pengaruh Promosi Dan Persepsi Harga Terhadap Minat Beli Kembali Diamond Game Mobile Legends Di Kota Medan," Universitas HKBP Nonmesen Ekonomi Manajemen , Nov. 2022.

[13] G. A. Sandag, "Model Prediksi Kemenangan Tim dalam Game League of Legend Menggunakan Algoritma Decision Tree," Jurnal Komputer Terapan, vol. 7, no. 1, 2021, [Online]. Available: https://jurnal.pcr.ac.id/index.php/jkt/

[14] A. Dharmawan, J. Gondohanindijo, Y. Prihati, and T. Hafidzin, "Optimization Of Players And Game Win Prediction Using Naïve Bayes Algorithm," Jurnal Elektro Luceat, vol. 8, no. 1, 2022.

[15] G. Sani, G. Prawira, and H. Setiaji, "Penerapan Data Transformation Pada Database Sistem Informasi Manajemen Rumah Sakit," Sintak, 2019.

[16] S. Mutmainah, "Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke," Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi, vol. 1, no. 1, 2021, doi: 10.20885/snati.v1i1.2.

[17] P. Singh, Learn PySpark. Apress, 2019. doi: 10.1007/978-1-4842-4961-1.

[18] N. A'ayunnisa, Y. Salim, and H. Azis, "Analisis performa metode Gaussian Naïve Bayes untuk klasifikasi citra tulisan tangan karakter arab," Indonesian Journal of Data and Science (IJODAS), vol. 3, no. 3, pp. 115–121, 2022.

[19] D. Gunawan, D. Riana, D. Ardiansyah, F. Akbar, and S. Alfarizi, "Komparasi Algoritma Support Vector Machine Dan Naïve Bayes Dengan Algoritma Genetika Pada Analisis Sentimen Calon Gubernur Jabar 2018-2023", doi: 10.31294/jtk.v4i2.