

OPTIMALISASI SUSUNAN PEMAIN DAN PREDIKSI KEMENANGAN GAME MENGGUNAKAN ALGORITMA NAÏVE BAYES

OPTIMIZATION OF PLAYERS AND GAME WIN PREDICTION USING NAÏVE BAYES ALGORITHM

Alexander Dharmawan¹, Jutono Gondohanindijo², Yani Prihati³, Taufiq Hafidzin⁴

^{1,2,3,4}Universitas AKI

¹alexander.dharmawan@unaki.ac.id, ²jutono.gondohanindijo@unaki.ac.id, ³yani.prihati@unaki.ac.id,
⁴hatsuke98@gmail.com

Abstrak

Mobile Legends Bang Bang telah berkembang menjadi salah satu *mobile game* yang populer. Bahkan telah dipertandingkan di SEA Games 2021 di Vietnam melalui cabang olah raga *esports*. Banyak pemain yang mencoba mengikuti perlombaan tetapi belum mencapai hasil yang maksimal. Faktor-faktor yang mempengaruhi diantaranya adalah susunan team dan hero yang digunakan. Penelitian ini bertujuan untuk menghitung prediksi kemenangan dan susunan suatu tim melalui *specialty*, *role*, dan *counter hero* menggunakan algoritma *naïve bayes* dan menerapkannya pada pertandingan sehingga harapan kemenangan dapat tercapai. Penggunaan parameter untuk perhitungan probabilitas kemenangan dihitung menggunakan rumus *naïve bayes* $P(X|C_i) = P(x_1|C_i) \cdot P(x_2|C_i) \cdot P(x_3|C_i) \cdot P(x_4|C_i)$ dimana masing-masing variabel mewakili total kemenangan atau kekalahan, total *winrate specialty hero* dalam satu tim, jenis *role hero* yang digunakan, dan ada atau tidaknya *counter hero* tim lawan dalam tim sekutu. Hasil uji yang didapatkan, *winrate specialty* berpengaruh sebesar 91%, jenis *role* memiliki pengaruh 86%, dan *counter hero* berpengaruh sebesar 60%, dan untuk tingkat kemenangan, didapatkan hasil menang sebanyak 40 kali dari total 50 pertandingan yang artinya aplikasi ini memiliki akurasi sebesar 80%.

Kata kunci : *hero, speciality, role, naïve bayes*

Abstract

Mobile Legends Bang Bang has developed into one of the most popular mobile games. It has even been competed in the SEA Games 2021 Vietnam through esports. Many players are trying to take part in the race but have not achieved maximum results. The influencing factors include the composition of the team and the heroes used. This study aims to calculate the prediction of victory and the composition of a team through *specialty*, *role*, and *counter heroes* using the *nave Bayes* algorithm and apply it to matches so that the hope of victory can be achieved.

The use of parameters for calculating the probability of winning is calculated using the *nave Bayes* formula $P(X|C_i) = P(x_1|C_i) \cdot P(x_2|C_i) \cdot P(x_3|C_i) \cdot P(x_4|C_i)$ where each variable represents the total victory or defeat, the total *winrate of the specialty hero* in one team, the type of *role hero* used, and the presence or absence of the opposing team's *counter hero* in the allied team.

The test results obtained, the *specialty winrate* has an effect of 91%, the type of *role* has an influence of 86%, and the *counter hero* has an effect of 60%, and for the win rate, the results obtained are 40 times from a total of 50 matches, which means this application has an accuracy of 80 %.

Keywords: *hero, speciality, role, naïve bayes*

1. PENDAHULUAN

Teknologi telah berkembang dengan pesatnya, termasuk perkembangan *hardware* dan *software*. Perkembangan ini juga telah merambah ke teknologi permainan (*game*). *Smartphone* dengan spesifikasi yang memadai dan harga yang semakin terjangkau, sudah bisa mengakses banyak *game*. Dengan kemajuan *hardware* ini, tentu saja juga berpengaruh terhadap perkembangan *game software*. Para pengembang berlomba untuk menciptakan *game* yang berkualitas baik dari sisi cerita, kreatifitas, tampilan dan daya tarik lainnya. *Game* ini pun sudah bukan *game* yang bersifat individu tapi sudah ke permainan yang bersifat kelompok. Bahkan bisa dimainkan tanpa harus berada pada tempat yang sama karena sudah bersifat *online*.

Dari perkembangan itulah sekarang muncul cabang olahraga baru yaitu *e-sport* yang merupakan kependekan dari *elektronik sport*. Cabang olah raga ini tergolong masih muda, kekinian [1] dan memiliki potensi untuk berkembang. Sudah banyak digelar pertandingan-pertandingan *e-sport* yang mana tujuannya adalah untuk mencari bibit-bibit atlet *e-sport* yang berbakat.

Salah satu *game* yang masuk cabang *e-sport* adalah Mobile Legends Bang Bang. Bahkan sudah dipertandingkan di SEA Games 2021 di Vietnam. *Game* ini dimainkan oleh 2 tim yang masing-masing tim berjumlah 5 orang. Permainan dimulai dengan setiap pemain akan memilih 1 *hero*. *Hero* merupakan karakter yang dapat dipilih dan dimainkan oleh satu *player* di setiap ronde permainan. Jumlah *hero* ada sekitar 100 yang dapat dimainkan sehingga membuat *game* dengan sistem 5 pemain lawan 5 pemain ini menjadi sangat bervariasi. Dimana setiap *hero* mempunyai karakteristik sendiri-sendiri.

Dari 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 memprediksi kemenangan dengan pemilihan *hero* dan susunan yang tepat.

2. DASAR TEORI /MATERIAL DAN METODOLOGI/PERANCANGAN

2.1 Tinjauan Pustaka

Dalam penelitian yang berjudul “Sistem Prediksi Kemenangan Tim Pada Game Mobile Legends Dengan Metode *Naïve Bayes*”, Andrianto Cahyono Putro menggunakan *ability effect*, *durability*, *offense* dan *difficulty* sebagai parameternya. Dan didapat 16 pertandingan dengan prediksi benar dari 20 pertandingan atau dalam bentuk persentase sebesar 75% dan tingkat error sebesar 25% [2].

Berikutnya adalah Eduardus Hardika Sandy Atmaja dengan penelitiannya yang berjudul “Prediksi Kemenangan *e-sport* DOTA 2 Berdasarkan Data Pertandingan”. Metode yang digunakan dalam penelitian ini adalah algoritma klasifikasi *Naive Bayes*. *Tool* yang digunakan untuk implementasi sistem adalah *Weka*. Berdasarkan serangkaian kegiatan yang telah dilakukan dalam penelitian tersebut maka disimpulkan bahwa algoritma *Naive Bayes* dapat digunakan untuk melakukan prediksi dengan baik. Akurasi yang didapatkan sebesar 98,804 % dengan *3-fold cross validation* dengan jumlah data 50000 [3].

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Christian Adikusuma Tanjung dengan judul “Sistem Penentuan Hero Counter Pada Game Mobile Legends Dengan Logika *Fuzzy* Metode Mamdani”. *Tool* yang digunakan untuk implementasi sistem adalah *MatLab*. Pembentukan himpunan *fuzzy*, fungsi implikasi dengan fungsi MIN, komposisi aturan dengan fungsi MAX, dan defuzzyfikasi dengan metode centroid menghasilkan angka akurat 84,375% [4].

Dari penelitian-penelitian sebelumnya, dapat disimpulkan bahwa Algoritma *Naïve Bayes* dapat digunakan untuk memprediksi kemenangan dengan parameter-parameter tertentu. Dalam penelitian ini parameter yang digunakan adalah *specialty*, *role*, dan *counter hero*.

Algoritma *Naïve Bayes*

Naïve Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Menurut Olson dan Delen [5] menjelaskan *naïve bayes* untuk setiap kelas keputusan, menghitung probabilitas bersyarat bahwa kelas keputusan adalah kelas yang benar, mengingat vektor informasi objek. Algoritma mengasumsikan bahwa atribut objek adalah independen. Probabilitas yang terlibat dalam menghasilkan perkiraan akhir dihitung sebagai jumlah frekuensi dari tabel keputusan "master". Pengklasifikasi *naïve bayes* sering bekerja dengan sangat baik dalam praktiknya, dan hasil klasifikasi yang sangat baik dapat diperoleh bahkan ketika perkiraan probabilitas mengandung kesalahan besar.

Sedangkan menurut Han dan Kamber [6] proses dari *The Naïve Bayesian classifier*, atau *Simple Bayesian Classifier*, sebagai berikut :

1. *Variable D* menjadi pelatihan set *tuple* dan *label* yang terkait dengan kelas. Setiap *tuple* diwakili oleh vektor atribut *n*- dimensi, $X = (x_1, x_2, \dots, x_n)$, ini menggambarkan pengukuran *n* dibuat pada *tuple* dari atribut *n*, masing-masing, A_1, A_2, \dots, A_n .
2. Misalkan ada kelas *m*, C_1, C_2, \dots, C_m . Diberi sebuah *tuple*, X , *classifier* akan memprediksi X yang masuk kelompok memiliki probabilitas *posterior* tertinggi, kondisi-disebutkan pada X . Artinya, *classifier naive bayesian* memprediksi bahwa X *tuple* milik kelas C_i jika dan hanya jika :

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

Jadi memaksimalkan $P(C_i | X)$. C_i kelas yang $P(C_i | X)$ dimaksimalkan disebut hipotesis posteriori maksimal. Dengan teorema Bayes:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Keterangan :

- $P(C_i|X)$ = Probabilitas hipotesis C_i jika diberikan fakta atau record X (Posterior probability)
- $P(X|C_i)$ = mencari nilai parameter yang memberi kemungkinan yang paling besar (likelihood)
- $P(C_i)$ = Prior probability dari X (Prior probability)
- $P(X)$ = Jumlah probability *tuple* yg muncul.

3. Ketika $P(X)$ adalah konstan untuk semua kelas, hanya $P(X | C_i) P(C_i)$ butuh dimaksimalkan. Jika probabilitas kelas sebelumnya tidak diketahui, maka umumnya diasumsikan ke dalam kelas yang sama, yaitu, $P(C_1) = P(C_2) = \dots = P(C_m)$, maka akan memaksimalkan $P(X | C_i)$. Jika tidak, maka akan memaksimalkan $P(X | C_i) P(C_i)$. Perhatikan bahwa probabilitas sebelum kelas dapat diperkirakan oleh $P(C_i) = |C_i, D| / |D|$, dimana $|C_i, D|$ adalah jumlah *tuple* pelatihan kelas C_i di D .
4. Mengingat *dataset* mempunyai banyak atribut, maka akan sangat sulit dalam mengkomputasi untuk menghitung $P(X|C_i)$. Agar dapat mengurangi perhitungan dalam mengevaluasi $P(X|C_i)$ bersyarat dibuat. Dianggap bahwa $P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$ satu sama lain, diberikan kelas $= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i)$. tungan diantara atribut) dengan demikian :

Maka dapat dengan mudah memperkirakan probabilitas $P(x_1 | C_i)$, $P(x_2 | C_i)$, . . . , $P(x_n | C_i)$ dari pelatihan *tuple*. Ingat bahwa di sini x_k mengacu pada nilai atribut A_k untuk *tuple* X . Untuk setiap atribut, dilihat dari apakah atribut tersebut kategorikal atau *continuous-valued*. Misalnya, untuk menghitung $P(X | C_i)$ mempertimbangkan hal-hal berikut:

- Jika A_k adalah kategorikal, maka $P(X_k | C_i)$ adalah jumlah *tuple* kelas C_i di D memiliki nilai X_k untuk atribut A_k , dibagi dengan $|C_i, D|$, jumlah *tuple* kelas C_i di D .
- Jika A_k *continuous-valued*, maka perlu melakukan sedikit lebih banyak pekerjaan, tapi perhitungannya cukup sederhana. Sebuah atribut *continuous-valued* biasanya diasumsikan memiliki distribusi *Gaussian* dengan rata-rata μ dan standar deviasi σ , didefinisikan

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

sehingga :

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

Setelah itu hitung μ_{C_i} dan σ_{C_i} , yang merupakan deviasi *mean* (rata-rata) dan standar masing-masing nilai atribut k untuk *tuple* pelatihan kelas C_i . Setelah itu gunakan kedua kuantitas dalam Persamaan, bersama-sama dengan x_k , untuk memperkirakan $P(x_k | C_i)$.

- Untuk memprediksi *label* kelas x , $P(X|C_i)P(C_i)$ dievaluasi untuk setiap kelas C_i . *Classifier* memprediksi kelas *label* dari *tuple* x adalah kelas C_i , jika :

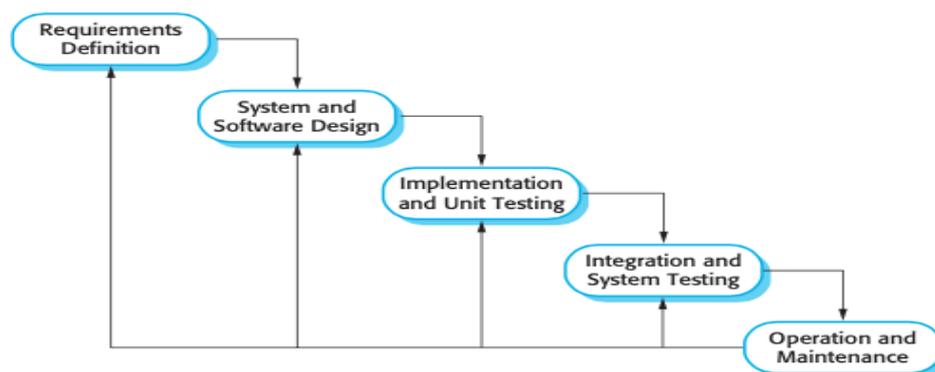
$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

Dengan kata lain, label kelas diprediksi adalah C_i yang mana $P(X | C_i) P(C_i)$ adalah maksimal.

Pengklasifikasi *Bayesian* memiliki tingkat kesalahan minimal dibandingkan dengan klasifikasi lainnya. Namun, dalam prakteknya hal ini tidak selalu terjadi, karena ketidakakuratan asumsi yang dibuat untuk penggunaannya, seperti kondisi kelas independen, dan kurangnya data probabilitas yang tersedia. Pengklasifikasi *Bayesian* juga berguna dalam memberikan pembenaran teoritis untuk pengklasifikasi lain yang tidak secara eksplisit menggunakan teorema *Bayes*.

2.2 Metode Penelitian

Metode pengembangan sistem yang digunakan adalah metode *waterfall* menurut referensi Sommerville [7], yaitu metode yang menggambarkan proses *software development* dalam aliran sekuensial. Model *waterfall* yaitu suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, design, kode, pengujian dan pemeliharaan. Jika telah memasuki tahap selanjutnya dalam *project* ini, maka anda tidak dapat kembali. Hal yang sama menurut Pressman dalam Lianawati, yang menyatakan bahwa metode *waterfall* menggambarkan pendekatan yang sistematis juga berurutan pada pengembangan perangkat lunak [8].



Gambar 1. Model Waterfall

Langkah-langkah dalam metode waterfall pada penelitian ini :

1. *Requirement analysis and Definition*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap. Tahapan ini merupakan proses dimana penulis menentukan klasifikasi data yang akan membantu dan mendukung dalam perancangan basis data untuk mempermudah dan memperjelas dalam pengaksesan program yang akan dibuat. Mengumpulkan data mulai dari *hero*, *specialty*, *meta hero*, dan *counter hero* kemudian menganalisa data tersebut agar sesuai dengan yang dibutuhkan.

2. *System and Software Design*

Desain sistem merupakan tahap penyusunan proses, data, aliran proses, dan hubungan antar data yang paling optimal untuk menjalankan proses bisnis dan memenuhi kebutuhan sesuai dengan hasil analisis kebutuhan. Di tahap ini, menentukan dan membuat desain sistem dan aliran proses dari sistem yang akan dirancang.

3. *Implementation and Unit Testing*

Pada tahap ini merupakan tahap untuk mengubah desain yang telah dibuat menjadi sebuah sistem yang dapat berjalan sesuai dengan kebutuhan. Tahap ini merupakan pengkodean dari desain ke dalam suatu bahasa pemrograman. Dalam sistem ini desain yang telah dibuat dikodekan dengan menggunakan salah satu bahasa pemrograman. Data yang diperoleh dari suatu desain sistem yang telah dirancang akan diubah ke dalam bahasa komputer atau diubah menjadi kode. Untuk tahap ini, mulai melakukan pengkodean menggunakan bahasa pemrograman yang telah ditentukan untuk menciptakan desain sistem dan aliran proses yang telah dirancang sebelumnya.

4. *Integration and System Testing*

Agar sistem aplikasi yang telah dibuat dapat berjalan dengan baik dan memberikan hasil yang optimal, maka perlu proses pengujian. Pendekatan yang penulis gunakan adalah *blackbox*, dimana program dianggap sebagai suatu “blackbox”, pengujian berbasiskan spesifikasi, kebenaran perangkat lunak yang diuji hanya dilihat berdasarkan keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk

mendapatkan keluaran tersebut. Melakukan testing pada aplikasi yang telah dibuat untuk menguji apakah sistem telah berjalan sesuai dengan yang diinginkan.

5. *Operation and Maintenance*

Ini merupakan tahap perawatan sistem yang telah dikembangkan seperti perawatan perangkat lunak, perawatan perangkat keras dan media lain yang berhubungan dengan komputer. Pada tahap ini pula harus dijaga *performance* perangkat lunak agar berjalan dengan baik. Pada tahap akhir ini, melakukan perawatan mulai dari software dan hardware agar performa dari sistem yang telah dibuat tetap stabil dan mengupdate hero-hero baru yang akan hadir dalam game.

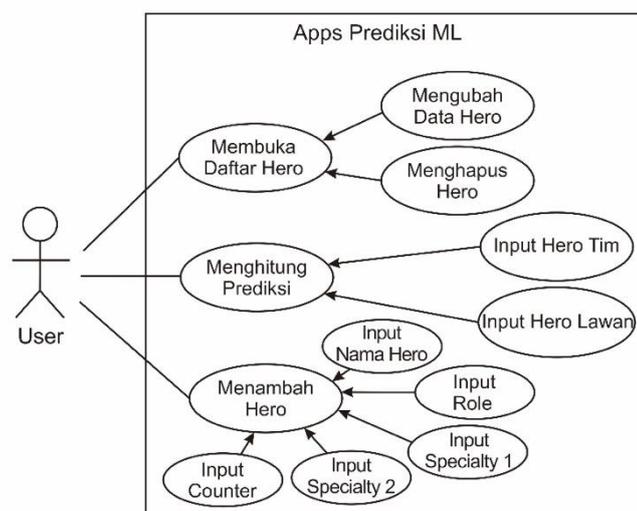
3. PEMBAHASAN

Aplikasi yang dirancang berdasarkan kebutuhan para pemain Mobile Legends yang ingin menuju ke tingkat professional untuk memudahkan dalam pemilihan hero saat mode *Draft Pick* dan memastikan pada sesi pemilihan hero, hero yang dipilih memiliki nilai persentase kemenangan pada *early game* yang cukup untuk mengalahkan tim lawan dan mencapai kemenangan.

Sebagai karakter yang dikendalikan langsung oleh pemain Hero memiliki spesialisasi dan keunikan tersendiri berupa kemampuan atau *skill* yang menjadi penentu bagaimana hero tersebut berperan dalam *teamfight*. Karena kemampuan tersebut Hero juga memiliki musuh alami berupa Hero yang memiliki kemampuan menetralkan seluruh *skill* yang dimilikinya atau biasa disebut sebagai *counter hero*. Oleh karena itu sangatlah penting dalam memilih hero pada mode *Draft Pick*.

Use Case Diagram

Use case menurut Rosa merupakan permodelan untuk kelakuan sistem informasi yang dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [9]. Sedangkan menurut Munawar *use case* adalah deskripsi fungsi dari sebuah sistem dari perpektif pengguna. *Use case* bekerja dengan cara mendiskripsikan interaksi antar pengguna [10]. *Use case diagram* aplikasi prediksi kemenangan dan susunan tim pada game Mobile Legends Bang Bang adalah sebagai berikut :



Gambar 2. *Use case* aplikasi optimalisasi pemain dan prediksi kemenangan

Penerapan *Naïve Bayes Classifier*

Untuk memperoleh hasil yang akurat selain dari data *record* yang berkualitas penulis juga menggunakan algoritma *naïve bayes* untuk melakukan perhitungan data dimana algoritma *naïve bayes* cocok untuk perhitungan data yang memiliki record relatif banyak dan memiliki lebih dari satu parameter.

Aplikasi ini menggunakan tiga parameter utama sebagai dasar perhitungan algoritma, yaitu rata-rata persen *winrate specialty* dari seluruh *hero* dalam satu tim, total jenis *role* yang ada dalam satu tim, dan ada atau tidaknya *counter hero* lawan dalam tim tersebut. Untuk lebih jelasnya dapat dilihat pada tabel berikut.

Tabel 1. Penetapan Parameter

No.	Parameter	Skala	Keterangan
1.	Total Specialty Winrate	$\geq 40\%$ s/d $\leq 100\%$	Total specialty winrate diperoleh dari rata-rata kelima specialty winrate dari hero dalam satu tim.
2.	Jenis Role	1 s/d 5	Jenis role yang dimaksud adalah banyaknya role yang berbeda pada satu tim, sehingga apabila terdapat jenis role yang sama lebih dari satu hero maka akan tetap dihitung satu role.
3.	Counter	Ya atau Tidak	Keberadaan satu atau lebih hero yang dapat meng- <i>counter</i> hero milik lawan

Dengan parameter tersebut dan data record yang telah dikumpulkan langkah selanjutnya adalah menyusun parameter tersebut dan memasukkannya ke dalam rumus *naïve bayes*. Dikarenakan perhitungan ini memiliki tiga parameter maka digunakan rumus :

$$\begin{aligned}
 P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\
 &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i).
 \end{aligned}$$

Dimana $P(X|C_i)$ adalah probabilitas menang atau kalah dan $P(x_n|C_i)$ adalah probabilitas dari parameter yang ada. Kemudian dibandingkan untuk memperoleh nilai yang terbesar antara menang atau kalah dengan dasar :

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i.$$

Sehingga diperoleh nilai maksimal untuk masing-masing perhitungan menang atau kalah.

Berdasarkan data *record* sebanyak 50 hasil pertandingan diperoleh data :

1. Total pertandingan sebanyak 50 *match* dengan hasil menang sebanyak 45 *match* dan hasil kalah 5 *match*.
2. Rata-rata *winrate specialty* (WRS) yang berada pada angka terendah sebesar 45% dan tertinggi sebesar 90% dengan detail bervariasi tiap *match*.
3. Terdapat 16 *match* dengan 5 jenis *role hero*, 27 *match* dengan 4 jenis *role hero*, 5 *match* dengan 3 jenis *role hero*, dan 2 *match* dengan 2 jenis *role hero*.
4. *Counter hero* lawan terdapat pada 30 *match* dengan hasil menang dan 2 *match* dengan hasil kalah sehingga terdapat total 32 *match* dengan *counter hero* lawan pada tim sendiri.

Untuk perhitungan kemungkinan menang berdasarkan *record* yang ada adalah sebagai berikut :

$$P(X|C_i) = P(x_1|C_i) \cdot P(x_2|C_i) \cdot P(x_3|C_i) \cdot P(x_4|C_i)$$

Menjadi :

$$P(\text{Menang}) = \left(\frac{\text{total menang}}{\text{total pertandingan}}\right) \cdot \left(\frac{\text{rata-rata WRS}}{\text{total menang}}\right) \cdot \left(\frac{\text{jumlah role}}{\text{total menang}}\right) \cdot \left(\frac{\text{counter}}{\text{total menang}}\right)$$

Dan

$$P(\text{Kalah}) = \left(\frac{\text{total kalah}}{\text{total pertandingan}}\right) \cdot \left(\frac{\text{rata-rata WRS}}{\text{total kalah}}\right) \cdot \left(\frac{\text{jumlah role}}{\text{total kalah}}\right) \cdot \left(\frac{\text{counter}}{\text{total kalah}}\right)$$

Dimana :

P (Menang) = Probabilitas menang

P (Kalah) = Probabilitas kalah

Total menang = Jumlah pertandingan yang dimenangkan

Total kalah = Jumlah pertandingan kalah

Rata-rata WRS = Jumlah rata-rata winrate specialty dengan kondisi menang/kalah

Jumlah role = Jumlah role yang berbeda dengan kondisi menang/kalah

Counter = Jumlah pertandingan dengan adanya counter hero lawan dalam tim dengan kondisi menang/kalah

Setelah itu hasil dari P(Menang) dan P(Kalah) dibandingkan, apabila P(Menang) memiliki nilai yang lebih tinggi dari P(Kalah) maka bisa dipastikan hasilnya akan menang, sebaliknya apabila P(Menang) lebih rendah dari P(Kalah) maka dipastikan hasilnya akan kalah.

Sebagai contoh, berdasarkan data tersebut apabila terdapat record susunan tim baru dengan nilai rata-rata winrate specialty sebesar 80% (nilai rata-rata winrate specialty 80% terdapat 5 record, 4 record menang 1 record kalah), jumlah jenis role yang ada pada satu tim sebanyak 4 role, dan terdapat counter hero lawan maka perhitungannya adalah sebagai berikut :

$$\begin{aligned} P(\text{Menang}) &= (45/50) \cdot (4/45) \cdot (26/45) \cdot (23/45) \\ &= \mathbf{0,0236} \end{aligned}$$

$$\begin{aligned} P(\text{Kalah}) &= (5/50) \cdot (1/5) \cdot (1/5) \cdot (9/5) \\ &= \mathbf{0,0072} \end{aligned}$$

Jadi karena hasil P(Menang) dengan nilai 0,0236 lebih besar daripada P(Kalah) yang nilainya 0,0072 maka bisa dipastikan tim dengan susunan tersebut menang.

Pengaruh Specialty Terhadap Kemenangan Tim

Berdasarkan data training yang diperoleh dengan memainkan seratus kali pertandingan menggunakan hero dengan specialty tertentu, specialty yang memiliki kemungkinan tertinggi memenangkan pertandingan adalah specialty Push yang memiliki total kemenangan sebesar 91 kali dari 100 pertandingan atau bisa disebut *winrate specialty* = 91% dan yang terendah adalah specialty Guard dengan total kemenangan sebesar 40 kali dari 100 pertandingan atau bisa disebut *winrate specialty* = 40%.

Dengan data tersebut dapat disimpulkan bahwa hero dengan *specialty Push* memiliki peluang menang paling tinggi dibandingkan dengan *specialty* yang lain. Kemampuan hero Push

yang mampu menghancurkan turret tim lawan dengan cepat menjadikan hero dengan spesialisasi ini memiliki *winrate* yang tinggi karena pada dasarnya kondisi menang dalam game Mobile Legends adalah apabila *base* milik lawan dapat dihancurkan.

Namun dikarenakan hanya terdapat sedikit hero dengan *specialty Push* dan kehadiran *counter hero* dengan spesialisasi ini di dalam tim lawan membuatnya harus didukung oleh hero dengan spesialisasi lain untuk memancing tim lawan melakukan *war* sehingga tugasnya untuk menghancurkan turret tim lawan dapat berjalan dengan lancar tanpa adanya gangguan dari *hero* lawan.

Pengaruh Jenis *Role Hero* Terhadap Kemenangan Tim

Berdasarkan data training yang diperoleh dan perhitungan dengan rumus *naïve bayes* semakin banyak jenis *role hero* yang ada pada sebuah tim maka kemungkinan menang tim tersebut juga akan meningkat. Dibuktikan dengan data training dimana lebih dari 50% kemenangan dari total data dengan hasil menang terdapat 4-5 jenis *role hero* dalam satu tim.

Selain itu, kebiasaan yang dilakukan oleh para *player* Mobile Legends dari yang pemula hingga ke tingkat profesional bahwa dalam satu tim setidaknya harus terdapat satu *role Tank* untuk menerima *damage* yang masuk, satu *role Marksman* untuk memberi *damage* pada lawan, dan satu *role Mage* untuk memberi *damage* tambahan, kemudian sisanya melakukan *push turret*, semakin menguatkan pernyataan bahwa kondisi menang dipengaruhi banyaknya jenis *role hero* yang digunakan dalam sebuah tim.

Perhitungan yang telah dilakukan membuktikan bahwa tim yang memiliki susunan 3 sampai 5 jenis *role* yang berbeda memiliki probabilitas menang lebih tinggi dibandingkan dengan tim yang memiliki susunan kurang dari 3 jenis *role*. Semakin banyak *role* yang ada pada sebuah tim maka variasi permainan akan lebih banyak dan strategi untuk *mid-game* hingga *late-game* dapat lebih fleksibel.

Pengaruh *Counter Hero* Terhadap Kemenangan Tim

Kehadiran *counter hero* lawan dalam tim sekutu memiliki perbandingan 1 banding 2 berdasarkan data training yang telah dikumpulkan dimana dalam 45 kemenangan terdapat 30 kondisi menang dengan adanya *counter hero* lawan dalam tim sekutu dan 15 kondisi menang tanpa adanya *counter hero* lawan. Jika dibandingkan dengan kedua parameter sebelumnya yakni *winrate specialty* dan jenis *role*, kehadiran *counter hero* memiliki probabilitas yang tidak lebih banyak dari kedua parameter tersebut.

Namun dalam banyak kasus, salah satunya seperti yang terjadi pada pertandingan *El Classico* atau pertandingan antara dua tim besar yang telah diketahui dan diakui kemampuannya di tingkat Internasional, yaitu pertandingan antara Evos Legends dengan RRQ Hoshi pada MPL season 5 tahun 2019, Lemon, seorang *player* dari tim RRQ Hoshi yang menggunakan hero Claude terus didesak oleh tim lawan yang menggunakan hero Gatotkaca yang notabene sebagai *counter hero* dari Claude membuatnya tidak mampu melakukan *farming* secara optimal.

Berdasarkan hal tersebut kehadiran *counter hero* lawan pada sebuah tim mampu menambah probabilitas kemenangan tim sekutu terhadap tim lawan meskipun tidak sebesar *winrate specialty* dan jenis *role* yang digunakan. Dalam prakteknya *counter* mampu menghambat pertumbuhan *hero* lawan yang di-*counter* sehingga mampu memberikan kesempatan tim sekutu untuk menjadi lebih kuat dari *early-game* hingga *late-game*.

Perhitungan Akurasi

Berikut adalah hasil awal dari pertandingan yang telah menggunakan aplikasi untuk penentuan pemilihan hero.

Tabel 2. Pertandingan yang dilakukan berdasarkan perhitungan aplikasi

Pertandinganke-	Tim					Hasil
	Hero 1	Hero2	Hero 3	Hero 4	Hero 5	Tim
1	Nana	Minotaur	Dyrroth	Moskov	Ling	Kalah
2	X.Borg	Aldous	Nana	Grock	Hanabi	Menang
3	X.Borg	Cyclops	Uranus	Wanwan	Fanny	Menang
4	Valir	Belerrick	X.Borg	Granger	Karina	Menang
5	X.Borg	Belerrick	Granger	Odette	Lolita	Menang
6	Belerrick	Irithel	Hylos	Lunox	Lapu-lapu	Kalah
7	Sun	Dyrroth	Belerrick	Harith	Lesley	Menang
8	Hilda	Dyrroth	Nana	Tigreal	Claude	Menang
9	Clint	Bane	Minotaur	Jawhead	Cyclops	Menang
10	Clint	Khufra	Karina	Nana	Alucard	Menang
11	Clint	Aldous	Guinevere	Minotaur	Change	Menang
12	Alpha	Dyrroth	Jhonson	Odette	Kimmy	Menang
13	Dyrroth	Nana	Kagura	Lapu-lapu	Layla	Menang
14	X.Borg	Clint	Grock	Nana	Argus	Kalah
15	Jhonson	Clint	Leomord	Gord	Gatotkaca	Menang

Dari data tersebut akurasi yang didapatkan adalah 12 kali menang dari 15 kali pertandingan yang berarti aplikasi memiliki akurasi sebesar 80%.

Tampilan Aplikasi



Gambar 3. Tampilan Menu Utama

No	Hero	Role	Specialty	Specialty	Counte	Total WRS (%)
1	Akai	Tank	Guard	Control	Nana	45
2	Aldous	Fighter	Burst	Push	Kaja	85
3	Alice	Mage	Charge	Regen	Eudora	55
4	Alpha	Fighter	Charge	Damage	Layla	75
5	Alucard	Fighter	Chase	Damage	Karina	75
6	Angela	Support	Guard	Poke	Karrie	55
7	Argus	Fighter	Charge	Burst	Akai	70
8	Atlas	Tank	Control	Initiator	Harith	60
9	Aurora	Mage	Control	Poke	Lancelot	65
10	Badang	Fighter	Charge	Burst	Khufra	70
11	Balmond	Fighter	Damage	Regen	Sun	65
12	Bane	Fighter	Push	Burst	Zlong	85
13	Barats	Tank	Damage	Control	Diggie	70
14	Baxia	Tank	Initiator	Damage	Aldous	80
15	Belerick	Tank	Control	Regen	Gusion	50
16	Benedetta	Assassin	Chase	Burst	Lesley	70
17	Brody	Marksmar	Burst	Reap	Alice	80
18	Bruno	Marksmar	Reap	Burst	Masha	80
19	Carmilla	Support	Control	Damage	Masha	70
20	Cecilion	Mage	Poke	Burst	Gusion	75
21	Change'e	Mage	Poke	Burst	Saber	75
22	Chou	Fighter	Chase	Control	Nana	60

Specialty
 Initiator
 Control
 Poke
 Charge
 Push
 Burst
 Damage
 Regen
 Guard
 Chase
 Reap

Role
 Tank
 Marksmar
 Mage
 Assassin
 Fighter
 Support

UBAH

HAPUS

MENU

Gambar 4. Tampilan Menu Daftar Hero



INPUT HERO

ALLY

Karrie
Sun
Lylia
Masha
Saber

ENEMY

Aldous
Johnson
Brody
Change'e
Gusion

SARAN HERO

Kaja
Karrie
Alice
Saber

WIN VALUE (%) : 84

RESET

MENU

KETERANGAN

BERDASARKAN COUNTER HERO

BERDASARKAN ROLE TIM

Gambar 5. Tampilan Menu Hitung Prediksi



TAMBAH HERO BARU

Nama Hero	: PACQUITO
Role	: Fighter
Specialty 1	: Burst
Specialty 2	: Reap
Counter	: Belerick

SIMPAN

RESET

MENU

Gambar 6. Tampilan Menu TambahHero

4. KESIMPULAN

Berdasarkan perhitungan menggunakan algoritma *naïve bayes* maka dapat ditarik kesimpulan sebagai berikut :

1. Dengan parameter *specialty*, *role*, dan *counter* didapatkan lebih stabil dan akurat karena tidak akan berubah sepanjang pertandingan dibandingkan dengan parameter yang ditentukan oleh penelitian sebelumnya yaitu tingkat kesulitan, daya tahan hero, efek skill, dan *output damage*.
2. Hasil perhitungan aplikasi mencapai akurasi 80% dari data yang diperoleh yaitu 12 kemenangan dari total 15 pertandingan.
3. Hasil dari perhitungan akan lebih efektif jika disertai *Early Game Strategy* dalam pertandingan guna menghindari adanya *hero late game* yang akan semakin kuat apabila durasi pertandingan semakin lama yang tentunya akan mengurangi probabilitas kemenangan tim.

DAFTAR PUSTAKA

- [1] F. Kurniawan, "E-Sport Dalam Fenomena Olahraga Kekinian," *Jorpres (Jurnal Olahraga Prestasi)*, vol. 15, no. 2, pp. 61–66, 2020.
- [2] A. C. Putro, "Universitas 17 Agustus," *Surabaya Jl. Semolowaru No*, vol. 45, p. 60118, 1945.
- [3] E. H. S. Atmaja, "Prediksi Kemenangan eSport DOTA 2 Berdasarkan Data Pertandingan," *Avitec*, vol. 2, no. 1, pp. 31–38, 2020, doi: 10.28989/avitec.v2i1.612.
- [4] C. A. Tanjung, F. A. Hermawati, and E. Indasyah, "Aplikasi Metode Fuzzy Mamdani Untuk Penentuan Hero Counter Pada Permainan Mobile Legends," *Konvergensi*, vol. 15, no. 1, 2019, doi: 10.30996/konv.v15i1.2834.
- [5] D. L. Olson and D. Delen, *Advanced data mining techniques*, no. January. 2008.
- [6] M. K. Han Jiawei, *Data Mining Concepts and Techniques*, 3rd editio. Waltham: Morgan Kaufmann, 2012.
- [7] I. Sommerville, *Software Engineering*, 9th Editio. Boston: Pearson Education, Inc, 2011.
- [8] Y. Lianawati, "SISTEM INFORMASI SIMPAN PINJAM BERBASIS WEBSITE PADA KOPERASI SIMPAN PINJAM 'BETA KARINA JAYA' di CILACAP," *JELC*, vol. 7, no. 2, 2021.
- [9] M. S. A.S Rosa, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Revisi. Bandung: Informatika, 2018.
- [10] Munawar, *Analisis Perancangan Sistem Berorientasi Objek Dengan UML*. Bandung: Informatika, 2018.