

## Penerapan Metode Random Forest untuk Prediksi Win Ratio Pemain Player Unknown Battleground

1) **Reinardus Aji Haristu**

<sup>1</sup>Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma  
Paingan, Maguwoharjo, Depok, Sleman, D.I.Y, Indonesia  
Telp. (0274) 883037, 889368 Fax (0274) 886529  
E-mail: [reinardusajih@gmail.com](mailto:reinardusajih@gmail.com)

2) **Paulina H. Prima Rosa**

<sup>2</sup>Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma  
Paingan, Maguwoharjo, Depok, Sleman, D.I.Y, Indonesia  
Telp. (0274) 883037, 889368 Fax (0274) 886529  
E-mail: [rosa@usd.ac.id](mailto:rosa@usd.ac.id)

### ABSTRACTS

Online game that is on the rise as one of e-sports is the unknown battleground player. To win the game requires the right strategy. In this paper, a study of effective playing strategies in the player unknown battleground game is explored by extracting data from player statistics taken from the kaggle website. From the statistical data, a model is made to predict the win ratio of each player using the random forest method. From the results of the study, random forest was able to produce an accuracy of 88.19%.

**Keyword: data mining, Classification, Random Forest Method, Player Unknown Battleground, PUBG**

### PENDAHULUAN

Perkembangan teknologi di zaman sekarang ini sudah maju sangat pesat. Berbagai inovasi dan terobosan dibidang multimedia/hiburan diciptakan. *Game* adalah salah satu inovasi yang tercipta dari efek perkembangan teknologi pada bidang multimedia/hiburan. Saat ini berbagai game *online* mulai bermunculan, *game online* merupakan permainan yang dapat dimainkan secara daring (*online*). Saat ini *game* sudah bukan lagi hiburan untuk membuang waktu, melainkan sudah berubah menjadi industri baru yang potensial untuk menghasilkan pendapatan yang [1].

E-sport merupakan permainan video *game* yang kompetitif. saat ini *e-sport* begitu banyak dilirik oleh berbagai pihak, dari televisi sampai perusahaan-perusahaan yang bahkan tidak dalam bidang *e-sport*. Posisi *e-sport* sama halnya seperti catur yang sudah termasuk dalam kategori olahraga dan sudah ada perlombaan di berbagai olimpiade [2]. Saat ini ada beberapa aliran (*genre*) *e-sport* yang terkenal, diantaranya adalah MOBA (*Multiplayer Online Battle Arena*), FPS (*First Person Shooter*), *Fighting (Street Fighter)*, *Real Time Strategy Games*. Sudah banyak kompetisi yang diadakan untuk pertandingan *e-sport* dengan hadiah yang tidak sedikit.

Salah satu kompetisi yang terbaru adalah turnamen e-sport Asia Tenggara yang diselenggarakan oleh pemerintah Indonesia dengan total hadiah mencapai Rp1,4 M dan dilaksanakan di Jakarta pada tanggal 17 Oktober sampai 21 Oktober 2018. *Game* yang dipertandingkan antara lain adalah Mobile Legend, Arena Of Valor (AOV), Playerunknown's Battleground, Point Blank dan DOTA 2. Jika dilihat dari penyelenggara dan dari nominal hadiah yang diperebutkan, maka e-sport tidak bisa dipandang rendah. Potensi industri dalam bidang tersebut memang ada dan sudah terlihat [3].

Salah satu *game* yang termasuk dalam golongan *e-sport* yang sedang naik daun yaitu *Player Unknown Battleground* atau yang sering disebut dengan PUBG. *Game* ini termasuk dalam genre permainan FPS (*First Person Shooter*) namun juga dilengkapi dengan mode *Third Person*. Permainan ini memiliki beberapa mode bermain berdasar kategori keanggotaan yaitu *solo* (sendirian/tanpa tim), *duo* (tim beranggotakan dua orang) dan *squad* (tim beranggotakan empat orang).

Konsep permainan *Player Unknown Battleground* menggunakan konsep *survival*, dimana pemain yang bertahan hidup terakhir maka yang akan menang. Selama pertandingan pemain diminta untuk mencari

segala senjata dan perlengkapan yang dibutuhkan untuk bertahan hidup pada suatu pulau. Dalam permainan ini ada istilah *win ratio* yaitu nilai persentase yang menunjukkan rasio seberapa besar menang selama bermain dari awal mendaftarkan akun sampai terakhir I bermain [4].

Tidak mudah untuk bertahan hidup dan menjadi yang terakhir dalam pertandingan *game* tersebut. Terkadang butuh strategi yang matang untuk memenangkan pertandingan. Selama ini belum ada penelitian terkait pencarian strategi terbaik dalam bermain *game* ini. Dalam penelitian ini penulis akan menganalisis rekaman data statistik dari berbagai pemain dengan tujuan untuk mencari strategi terbaik dalam bermain *game* PUBG.

Ada beberapa penelitian yang sudah dilakukan dengan menggunakan metode *Random Forest*. Salah satunya adalah yang dilakukan oleh [5]. Dalam penelitiannya mereka melakukan klasifikasi penderita HIV/AIDS menggunakan metode *Random Forest* dan *Multivariate Adaptive Regression Spline* (MARS) *binary response*. Hasil yang didapat adalah mengetahui variabel yang paling berpengaruh untuk menentukan status HIV/AIDS yaitu usia, jenis pekerjaan, pernah ditahan karena kasus NAPZA, status nikah dan selalu pakai jarum steril. Akurasi yang didapatkan ketika menggunakan metode MARS sebesar 80,28%. Sedangkan ketika menggunakan metode *Random Forest* diperoleh akurasi terbaik yaitu 97,80%. Penulis tersebut juga mencoba gabungan antara metode MARS dan *Random Forest*, dan memperoleh hasil akurasi sebesar 91,00%.

Penelitian lain adalah yang dilakukan oleh [6]. Dalam penelitiannya mereka melakukan klasifikasi tingkat penerimaan konsumen terhadap mobil menggunakan metode *Random Forest*. Dari hasil penelitian, mereka mendapatkan bahwa variabel yang mempengaruhi tingkat penerimaan konsumen terdiri dari harga pembelian, biaya perawatan, jumlah pintu, kapasitas penumpang, ukuran bagasi dan taksiran keselamatan penumpang.

Berdasar latar belakang tersebut, dalam makalah ini penulis menggunakan metode *Random Forest* untuk memprediksi *win ratio*

terhadap data statistik pemain *game* *Player Unknown Battleground* dengan menggunakan metode *Random Forest*. Dengan demikian diharapkan dapat dihasilkan model untuk memprediksi *win ratio*. Model tersebut memiliki beberapa *tree* yang setiap *node* merupakan atribut yang dipilih karena kemampuannya yang baik untuk mengklasifikasi. Atribut yang dijadikan *Node* pada *tree* dapat dipertimbangkan sebagai strategi dalam bermain *game* PUBG.

Penelitian ini akan mencoba mengetahui bagaimana menerapkan metode *Random Forest* untuk memprediksi nilai *win ratio* dengan baik berikut atribut (fitur) apa yang paling berpengaruh dalam memprediksi nilai *win ratio* serta mengetahui berapa akurasi prediksi terbaik yang dihasilkan oleh metode *Random Forest* terhadap data statistik pemain *game* *Player Unknown Battleground*.

## METODOLOGI

Berikut ini adalah tahap-tahap penelitian yang dilakukan oleh penulis:

### 1. Pengumpulan Data

Tahap awal dalam penelitian ini adalah pengumpulan data. Pengumpulan data yang dilakukan dalam penelitian ini dengan cara mengunduh data dari *website* *Kaggle.com*.

### 2. Knowledge Discovery in Database (KDD)

Dengan mengacu proses KDD menurut [7], berikut ini adalah proses KDD yang dilakukan pada penelitian ini:

#### a. Data Selection

Dalam tahap seleksi data akan dilakukan pemilihan atribut yang relevan untuk digunakan dalam penelitian. Atribut yang tidak digunakan atau tidak relevan akan dihapus. Atribut yang akan dihapus dari data statistik Pemain *Player Unknown Battleground* adalah atribut yang tidak independen. Berikut ini adalah daftar atribut yang dihapus beserta alasannya yang ditunjukkan pada Tabel 1. Dengan demikian didapatkan atribut yang akan digunakan pada Tabel 2. Sebagai catatan, atribut *solo\_RoundsPlayed* akan disimpan untuk pengujian perangkat lunak menggunakan *dataset*.

Tabel 1 Daftar atribut yang dihapus

No	Nama Atribut	Alasan Penghapusan
1	player_name	Memiliki nilai unik (berbeda pada setiap baris data)
2	tracker_id	Memiliki nilai unik (berbeda pada setiap baris data)
3	solo_RoundsPlaye	Digunakan untuk menghitung atribut solo_WinRatio

No	Nama Atribut	Alasan Penghapusan
4	solo_Wins	Digunakan untuk menghitung atribut solo_WinRatio
5	solo_TimeSurvived	Digunakan untuk menghitung atribut solo_TimeSurvivedPg
6	solo_Top10s	Digunakan untuk menghitung atribut solo_WinTop10Ratio
7	solo_Kills	Digunakan untuk menghitung atribut solo_killDeathRatio
8	solo_Assists	Tidak memiliki nilai untuk permainan mode solo
9	solo_TeamKills	Tidak memiliki arti untuk permainan mode solo
10	solo_TeamKillsPg	Tidak memiliki arti untuk permainan mode solo
11	solo_HeadshotKills	Digunakan untuk menghitung atribut solo_HeadshotKillsPg
12	solo_RoadKills	Digunakan untuk menghitung atribut solo_RoadKillsPg
13	solo_Days	Digunakan untuk menghitung atribut solo_DailyKills
14	solo_MostSurvivalTime	Memiliki nilai yang sama dengan atribut solo_LongestTimeSurvived
15	solo_WalkDistance	Digunakan untuk menghitung atribut solo_AvgWalkDistance
16	solo_MoveDistance	Digunakan untuk menghitung atribut
17	solo_RideDistance+AP1	Digunakan untuk menghitung atribut solo_AvgRideDistance
18	solo_Revives	Tidak memiliki arti untuk permainan mode solo (menyelamatkan teman satu tim)
19	solo_WeaponAcquired	Memiliki nilai nol untuk semua baris data
20	solo_Assists	Tidak memiliki arti untuk permainan mode solo (membantu teman satu tim)
21	solo_DBNOs	Memiliki nilai nol untuk semua baris data
22	Solo_WinTop10Ratio	Menggunakan atribut solo_WinRatio (atribut label/kelas) untuk menghitung nilai

Tabel 2 Daftar atribut yang dihapus

No	Nama Atribut	Keterangan
1	solo_WinRatio	Rasio kemenangan
2	solo_killDeathRatio	Ratio jumlah korban
3	solo_Top10Ratio	Rasio kemenangan dalam 10 besar
4	solo_Losses	Jumlah kekalahan
5	solo_Rating	Rating yang dimiliki terakhir kali
6	solo_BestRating	Rating terbaik selama bermain
7	solo_DamagePg	Jumlah damage yang diciptakan per ronde
8	solo_HeadshotKillspg	Jumlah head shot yang diciptakan per ronde
9	solo_HealsPg	Jumlah poin heal (penyembuhan) per ronde
10	solo_KillsPg	Jumlah korban per ronde
11	solo_MoveDistancePg	Jarak yang ditempuh per ronde
12	solo_RoadKillsPg	Rata-rata korban yang dibunuh menggunakan kendaraan
13	solo_TimeSurvivedPg	Waktu bertahan hidup per ronde
14	solo_Top10sPg	Rata-rata kemenangan di sepuluh besar
15	solo_Suicides	Jumlah kejadian bunuh diri (mati tanpa diserang musuh)
16	solo_HeadshotKillRatio	Rasio head shot (korban tepat kepala)
17	solo_VehicleDestructions	Jumlah penghancuran kendaraan

No	Nama Atribut	Keterangan
18	solo_DailyKills	Jumlah korban harian (rata-rata korban dalam satu hari)
19	solo_WeeklyKills	Jumlah korban minggu (rata-rata korban dalam satu minggu)
20	solo_MaxKillStreaks	Jumlah korban terbanyak yang diciptakan berturut-turut
21	solo_LongestTimeSurvived	Waktu terlama bertahan hidup
22	solo_AvgSurvivalTime	Rata-rata waktu bertahan hidup
23	solo_WinPoints	Jumlah poin kemenangan terakhir
24	solo_AvgWalkDistance	Rata-rata jarak yang sudah ditempuh dengan jalan kaki
25	solo_AvgRideDistance	Rata-rata jarak yang sudah ditempuh dengan kendaraan
26	solo_LongestKill	Jarak korban terjauh
27	solo_Heals	Jumlah poin heal (penyembuhan)
28	solo_Boosts	Total energi yang digunakan
29	solo_DamageDealt	Jumlah poin damage (kerusakan) yang diterima
30	solo_RoundMostKills	Jumlah korban terbanyak dalam satu ronde

**b. Data transformation**

Ada beberapa transformasi data yang dilakukan penulis pada dataset statistik Pemain *Player Unknown Battleground*. Transformasi yang pertama adalah merubah nilai atribut solo\_WinRatio dari nilai yang kontinu menjadi kategorial dengan cara mengelompokkan nilai tersebut menjadi tiga kelompok. Dalam mengelompokkan nilai atribut solo\_WinRatio, penulis menggunakan *tools Weka* versi 3.8. Metode yang penulis pilih adalah metode simple k means dengan hasil nilai centroid akhir atribut solo\_WinRatio yang ditunjukkan pada Gambar 1.

Selain atribut solo\_WinRatio, atribut solo\_RoundsPlayed juga akan dikelompokkan (*clustering*) menjadi 3 kelompok yang menggambarkan tipe pemain yaitu pemula, normal, profesional. Cara pengelompokannya pun sama seperti mengelompokkan atribut solo\_WinRatio, yaitu menggunakan *tools Weka* versi 3.8 dengan metode simple k means. Hasil nilai akhir centroid atribut solo\_RoundsPlayed ditunjukkan pada Gambar 2. Pada Gambar 3 merupakan hasil klustering untuk atribut solo\_WinRatio setelah menghilangkan kelompok pemain dengan kluster 0 (pemula) berdasar atribut solo\_RoundsPlayed (Gambar 2).

Final cluster centroids:				
Attribute	Full Data	Cluster# 0	Cluster# 1	Cluster# 2
	(87898.0)	(38085.0)	(10199.0)	(39614.0)
solo_WinRatio	5.0175	2.663	19.1609	3.6398

Gambar 1 Hasil klustering *tools weka* untuk atribut solo\_WinRatio

Final cluster centroids:				
Attribute	Full Data	Cluster# 0	Cluster# 1	Cluster# 2
	(87898.0)	(38225.0)	(10254.0)	(39419.0)
solo_RoundsPlayed	79.2753	38.2316	10.8726	136.8694

Gambar 2 Hasil klustering *tools weka* untuk atribut solo\_RoundsPlayed

Final cluster centroids:				
Attribute	Full Data	Cluster# 0	Cluster# 1	Cluster# 2
	(58366.0)	(13804.0)	(24153.0)	(20409.0)
solo_WinRatio	3.0715	6.5579	1.8604	2.1468

Gambar 3 Hasil Klustering *Tools Weka* Untuk Atribut solo\_WinRatio Setelah Menghilangkan Kluster 0 atribut solo\_RoundsPlayed

Transformasi yang kedua adalah dengan menormalisasi seluruh nilai atribut kedalam *range*/jarak yang sama yaitu nol sampai satu. Proses normalisasi dilakukan dengan menggunakan metode *min-max normalization*, adapun rumus yang digunakan dalam metode tersebut dideskripsikan pada Persamaan 1 berikut:

$$v' = \frac{v - \min_A}{\max_A - \min_A} (new_{\max_A} - new_{\min_A}) + new_{\min_A} \quad (1)$$

Keterangan:

- $v$  : nilai sebelum ternormalisasi
- $v'$  : nilai setelah ternormalisasi
- $\min_A$  : nilai minimal dari atribut A
- $\max_A$  : nilai maksimal dari atribut A
- $new_{\min_A}$  : nilai minimal terbaru dari atribut A
- $new_{\max_A}$  : nilai maksimal terbaru dari atribut A

c. *Data mining*

Proses data mining adalah proses penggalian informasi dari data yang ada. Penggalian informasi pada penelitian ini berupa klasifikasi. Dalam penelitian ini proses klasifikasi akan dilakukan menggunakan metode *Random Forest* untuk data statistik pemain PUBG. *Random Forest* merupakan sebuah metode *ensemble*. Metode *ensemble* merupakan cara untuk meningkatkan akurasi metode klasifikasi dengan cara mengkombinasikan metode klasifikasi [7]. *Random Forest* diawali dengan teknik dasar *data mining* yaitu *decision tree*. Pada *decision tree* input dimasukkan pada bagian atas (*root*) kemudian turun kebagian bawah (*leaf*) untuk menentukan data tersebut termasuk kelas apa. *Random forest* adalah pengklasifikasi yang terdiri dari kumpulan pengklasifikasi pohon terstruktur dimana masing-masing pohon melemparkan unit suara untuk kelas paling populer di input  $x$  [8]. Dengan kata lain *Random Forest* terdiri dari sekumpulan *decision tree* (pohon keputusan), dimana kumpulan *decision tree* tersebut digunakan untuk mengklasifikasi data ke suatu kelas.

*Random Forest* merupakan sebuah metode *ensemble*. Metode *ensemble* merupakan cara untuk meningkatkan akurasi metode klasifikasi dengan cara mengkombinasikan metode klasifikasi (Han, 2012). *Random Forest* diawali dengan teknik dasar *data mining* yaitu *decision tree*. Pada *decision tree* input dimasukkan pada bagian atas (*root*) kemudian turun kebagian bawah (*leaf*) untuk menentukan data tersebut termasuk kelas apa. *Random forest* adalah pengklasifikasi yang terdiri dari kumpulan pengklasifikasi pohon terstruktur dimana masing-masing pohon melemparkan unit suara untuk kelas paling populer di input  $x$  [8]. Dengan kata lain *Random Forest* terdiri dari sekumpulan *decision tree* (pohon keputusan), dimana kumpulan *decision tree* tersebut digunakan untuk mengklasifikasi data ke suatu kelas.

*Random forest* merupakan metode klasifikasi yang *supervised*. Sesuai dengan namanya, metode ini menciptakan sebuah hutan (*forest*) dengan sejumlah pohon (*tree*). Secara umum, semakin banyak pohon (*tree*) pada sebuah hutan (*forest*) maka semakin kuat juga hutan tersebut terlihat. Pada kasus

yang sama, semakin banyak *tree*, maka semakin besar pula akurasi yang didapatkan [9].

*Decision Tree* akan menggunakan *information gain* untuk perhitungan dalam menentukan *root node* dan *rule*. Sama halnya dengan *Random Forest* yang akan menggunakan *information gain* dan *gini index* untuk perhitungan dalam membangun *tree* [7], hanya saja *Random Forest* akan membangun lebih dari satu *tree*. Masing-masing *tree* dibangun menggunakan set data dengan atribut yang diambil secara acak dari *data training*. Dengan kata lain setiap *tree* akan bergantung pada nilai dari sampel vektor yang independen dengan distribusi yang sama pada setiap *tree* (Han, 2012). Selama proses klasifikasi setiap *tree* akan memberikan *voting* kelas yang paling populer [7].

Berikut ini adalah algoritma untuk metode *random forest*. Algoritma dibagi menjadi dua bagian, bagian pertama adalah pembuatan “ $n$ ” pohon (*tree*) untuk membentuk hutan (*forest*) yang acak (*random*). Bagian kedua adalah algoritma untuk melakukan prediksi dari *Random Forest* yang sudah dibuat [7].

Input:

- $D$ , dataset yang terdiri dari  $d$  baris
- $k$ , angka dari jumlah *tree*

Algoritma untuk membentuk model *Random Forest* (Bagian 1):

- a. Buat data sampel data  $D_i$  dengan mengambil acak dari dataset  $D$  dengan pengembalian (*replacement*)
- b. Gunakan sampel data  $D_i$  untuk membangun *tree* ke  $i$  ( $i = 1, 2, \dots, k$ )
- c. Ulangi langkah a dan b sebanyak  $k$

Langkah metode *Random Forest* diawali dengan pemilihan “ $k$ ” sampel dataset  $D_i$  yang diambil secara acak dengan pengembalian (*replacement*). Langkah selanjutnya adalah menggunakan dataset  $D_i$  untuk membangun *decision tree* ke- $i$ . Dalam membangun *tree* ke- $i$ , metodologi CART dapat digunakan. Metodologi CART menggunakan *information gain* dalam menentukan setiap *node* pada *tree*. Perhitungan *information gain* dapat dihitung menggunakan Persamaan 2.

$$Gain(A) = Info(D) - Info_A(D) \quad (2)$$

Dimana nilai  $Info(D)$  dapat dicari menggunakan Persamaan 3 dan  $Info_A(D)$  dicari menggunakan Persamaan 4 dibawah ini:

$$Info(D) = -\sum_{i=1}^m p_i \log_2(p_i) \quad (3)$$

Keterangan:

- $m$  : jumlah kelas target  
 $p_i$  : probabilitas munculnya kelas ke  $i$  pada partisi  $D$

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (4)$$

Keterangan:

- $v$  : jumlah partisi  
 $D_j$  : total partisi ke  $j$   
 $D$  : jumlah *tuple*/baris pada semua partisi

Untuk menghitung *information gain* dari atribut yang memiliki nilai kontinu (numerik), maka harus ditentukan nilai pembelah (*split-point*) terbaik untuk mengelompokkan nilai dari atribut tersebut. Untuk mencari *split-point* terbaik maka data dari atribut tersebut harus diurutkan terlebih dahulu. Nilai tengah antara setiap pasangan nilai yang berdekatan dianggap sebagai kemungkinan yang bisa dijadikan *split-point* [7].

*Pseudocode* untuk proses prediksi *data test* (Bagian 2):

- Ambil *data test* dan gunakan rule dari setiap *tree* untuk memprediksi keluaran klasifikasi dari data tersebut, simpan hasil yang didapat
- Hitung suara (*vote*) untuk setiap target yang diprediksi dari setiap *tree*
- Pertimbangkan target prediksi yang terpilih dengan memilih target kelas yang paling banyak diprediksi sebagai hasil prediksi akhir dari metode *random forest*

Untuk memprediksi kelas target dari *data test* menggunakan *random forest*, masukan data test melalui aturan-aturan (*rule*) yang sudah dibuat menggunakan *tree*. Hasil prediksi setiap *tree* bisa saja ada yang berbeda dan ada yang sama, maka prediksi akhir akan dipilih berdasar prediksi kelas yang terbanyak diprediksi. Misalkan dari 100 *tree*, 80 *tree* memprediksi target adalah kelas A dan sisanya adalah kelas B, maka prediksi akhir yang dipilih adalah kelas A konsep pemilihan dengan suara terbanyak yaitu

hasil prediksi dari semua *tree* dinamakan *majority voting*.

Pengujian yang akan dilakukan, penulis menggunakan empat jenis *dataset*:

- Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data tidak ternormalisasi.
- Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi.
- Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data tidak ternormalisasi.
- Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data ternormalisasi.

Pada pengujian tersebut akan dilakukan pengujian klasifikasi dengan jumlah *fold* 3 dan kombinasi jumlah *tree* yang berbeda-beda dimulai dari 10 sampai 90 dengan interval 10. Dari pengujian ini akan dilihat hasil akurasi dari masing-masing pengujian.

- Pattern evaluation* dan *Knowledge presentation*

Untuk proses mengidentifikasi pola yang tepat yang merupakan hasil dari proses *data mining* (*Pattern evaluation*) dan proses penyajian hasil dari *data mining* kepada user (*Knowledge presentation*) dilakukan setelah sistem selesai dibangun dan proses penambangan data selesai dilakukan. Dalam proses ini penulis melakukan evaluasi dari hasil penambangan data yang didapat dari perangkat lunak yang telah dibangun dan menjelaskan hasil evaluasi tersebut agar informasi yang didapat dapat dengan mudah diterima oleh pihak-pihak yang membutuhkan. Kedua langkah tersebut akan dipaparkan pada bagian pembahasan.

### 3. Pengembangan Perangkat Lunak

Penulis melakukan pengembangan perangkat lunak sebagai alat untuk mengolah *dataset* yang dimiliki penulis untuk mendapatkan informasi (*knowledge*) yang berguna. Metode yang digunakan oleh penulis adalah metode *waterfall*. Metode tersebut adalah salah satu metode yang sangat sering digunakan oleh para pengembang perangkat lunak. Pengerjaan sistem secara *linear* diberlakukan dalam metode ini. Dimana jika tahap pertama belum selesai maka tahap kedua belum bisa dilakukan.

#### 4. Analisis Hasil

Dalam tahap ini dilakukan analisa kinerja terhadap model atau metode *Random Forest* yang sudah diimplementasikan ke dalam sebuah perangkat lunak.

#### PEMBAHASAN

##### 1. Pengujian *Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data tidak ternormalisasi

*Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data tidak ternormalisasi memiliki 87.898 baris data dan 30 atribut. Tabel pengujian *dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dapat dilihat di Tabel 3.

Tabel 3 Pengujian *dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data tidak ternormalisasi

No	Jumlah	Akurasi (%)
Tree		
1	10	86,11
2	20	87,14
3	30	87,55
4	40	87,85
5	50	87,85
6	60	87,86
7	70	88,19
8	80	87,81

##### 2. Pengujian *Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi

*Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi memiliki 87.898 baris data dan 30 atribut. Tabel pengujian *Dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi dapat dilihat di Tabel 4.

Tabel 4 Pengujian *dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi

No	Jumlah	Akurasi (%)
Tree		
1	10	86,11
2	20	87,14
3	30	87,55
4	40	87,85
5	50	87,85
6	60	87,86
7	70	88,19
8	80	87,81

##### 3. Pengujian *Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data tidak ternormalisasi

*Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data tidak ternormalisasi memiliki 58.366 baris data dan 30 atribut. Tabel pengujian *dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data tidak ternormalisasi dapat dilihat di Tabel 5.

Tabel 5 Pengujian *dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data tidak ternormalisasi

No	Jumlah	Akurasi (%)
Tree		
1	10	86,11
2	20	87,14
3	30	87,55
4	40	87,85
5	50	87,85
6	60	87,86
7	70	88,19
8	80	87,81

##### 4. Pengujian *Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data ternormalisasi

*Dataset* statistik Pemain *Player Unknown Battleground* tanpa *outlier* dan data ternormalisasi memiliki 58.366 baris data dan 30 atribut. Tabel pengujian *dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi dapat dilihat di Tabel 6.

Tabel 6 Pengujian *dataset* statistik Pemain *Player Unknown Battleground* dengan *outlier* dan data ternormalisasi

No	Jumlah	Akurasi (%)
Tree		
1	10	86,11
2	20	87,14
3	30	87,55
4	40	87,85
5	50	87,85
6	60	87,86
7	70	88,19
8	80	87,81

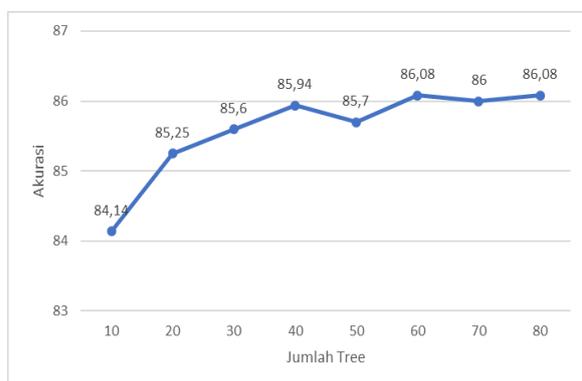
##### 5. Evaluasi Hasil Pengujian

Dari hasil pengujian ke empat *dataset* tersebut terdapat kesamaan pola pada pengujian keempat *dataset* tersebut yaitu akurasi yang cenderung bertambah ketika jumlah *tree* semakin besar. Pola tersebut dapat mudah di lihat pada grafik yang digambarkan pada Gambar 4 sampai 7. Hal tersebut menunjukkan bahwa semakin banyak

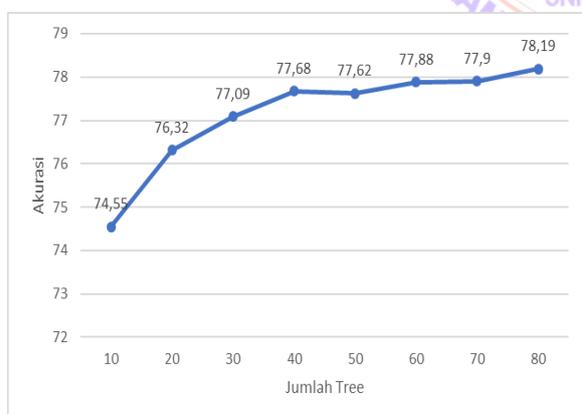
jumlah *tree* yang digunakan dalam membangun model maka akurasi cenderung meningkat.



Gambar 4 Hasil pengujian dataset statistik Pemain *Player Unknown Battleground* dengan outlier dan data tidak ternormalisasi



Gambar 5 Hasil pengujian dataset statistik Pemain *Player Unknown Battleground* dengan outlier dan data ternormalisasi



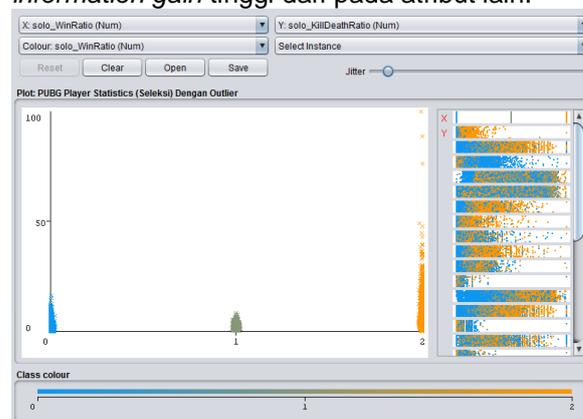
Gambar 6 Hasil pengujian dataset statistik Pemain *Player Unknown Battleground* tanpa outlier dan data tidak ternormalisasi



Gambar 7 Hasil pengujian dataset statistik Pemain *Player Unknown Battleground* tanpa outlier dan data ternormalisasi

Dari hasil pengujian keempat dataset tersebut didapatkan akurasi tertinggi yaitu 88,19%. Akurasi tersebut didapat dari hasil klasifikasi *dataset* statistik Pemain *Player Unknown Battleground* dengan outlier dan data tidak ternormalisasi dengan jumlah *tree* sebanyak 70. Jika dilihat model yang dibuat untuk mendapatkan hasil tertinggi tersebut, atribut yang sering digunakan sebagai *root node* maupun *node* lain adalah atribut *solo\_KillDeathRatio*. Artinya atribut tersebut sangat berperan besar dalam memprediksi atau mengklasifikasi *win ratio*.

Dengan memanfaatkan tools Weka, penulis mencoba untuk melihat bagaimana korelasi atribut tersebut dengan *win ratio*. Jika dilihat dari *sclater plot* pada Gambar 8, ada cukup banyak nilai *solo\_KillDeathRatio* yang tinggi yang menunjukkan *win ratio* tinggi (kluster 2) artinya jika nilai split diletakan pada tengah atribut *solo\_KillDeathRatio* maka ada cukup banyak data yang memiliki *win ratio* tinggi (kluster 2) yang terdapat pada salah satu subset. Hal tersebutlah yang menyebabkan atribut ini selalu memiliki nilai *information gain* tinggi dari pada atribut lain.



Gambar 8 *sclater plot* atribut *solo\_KillDeathRatio* dengan *win ratio*

## KESIMPULAN

Hasil penelitian penerapan metode *random forest* untuk prediksi *win ratio* pemain *player unknown battleground* ini menghasilkan kesimpulan bahwa Metode *random forest* dapat digunakan untuk melakukan prediksi/klasifikasi *win ratio* pemain *player unknown battleground* dengan baik. Semakin banyak jumlah *tree* yang digunakan maka semakin baik pula akurasi yang didapat. Atribut yang paling berpengaruh dalam menentukan klasifikasi *win ratio* pemain *player unknown battleground* adalah *solo\_KillDeathRatio*. Untuk Akurasi prediksi terbaik yang dihasilkan oleh metode *Random Forest* terhadap *win ratio* pada data statistik pemain game *Player Unknown Battleground* adalah sebesar 88,19%.

## SARAN

Untuk pengembangan penelitian yang akan datang dapat ditambahkan fungsi untuk menyimpan atau mengeksport model yang berhasil dibangun oleh perangkat lunak. Selain itu dapat juga ditambahkan fungsi untuk menampilkan semua rule yang memiliki leaf suatu label. Untuk mempermudah pemrosesan awal (*preprocessing*) juga dapat ditambahkan fungsi untuk melakukan transformasi/normalisasi data dengan secara otomatis dari sistem. Pada penelitian selanjutnya juga bisa dicobakan dataset yang berbeda atau metode yang berbeda.

## DAFTAR PUSTAKA

- [1] S. M. Jannah, "Bukan Cuma Main Game, Esport Mulai Jadi Industri Masa Depan," 2018. [Online]. Available: <https://finance.detik.com/berita-ekonomi-bisnis/d-4316768/bukan-cuma-main-game-esport-mulai-jadi-industri-masa-depan>. [Accessed: 24-Oct-2019].
- [2] R. Restika, "Apa Itu Esports? - Esportsnesia," 2018. [Online]. Available: <https://esportsnesia.com/penting/apa-itu-esports/>. [Accessed: 24-Oct-2019].
- [3] A. Suryo, "UniPin Bikin Kompetisi eSports Berhadiah Rp 1,4 Miliar," 2018. [Online]. Available: <https://inet.detik.com/games-news/d-4162385/unipin-bikin-kompetisi-esports-berhadiah-rp-14-miliar>. [Accessed: 24-Oct-2019].
- [4] O. Laoly and T. Limbong, "Visualisasi Pengumuman dan SOP Fakultas Ilmu Komputer Universitas Katolik Santo Thomas Medan berbasis Multimedia," *MEANS (Media Inf. Anal. dan Sist.*, vol.

3, no. 2, pp. 126–139, Dec. 2018.

- [5] W. O. Nidhomuddin; Bambang, "RANDOM FOREST DAN MULTIVARIATE ADAPTIVE REGRESSION SPLINE (MARS) BINARY RESPONSE UNTUK KLASIFIKASI PENDERITA HIV/AIDS DI SURABAYA | Nidhomuddin | Jurnal Statistika Universitas Muhammadiyah Semarang," *Jurnal Statistika Universitas Muhammadiyah Semarang*, 3, 1, 2015, 2015. [Online]. Available: <https://jurnal.unimus.ac.id/index.php/statistik/article/view/1439>. [Accessed: 24-Oct-2019].
- [6] Yusuf Sulisty Nugroho; Nova Emiliyawati, "Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest," 2017. [Online]. Available: [https://www.researchgate.net/publication/320413581\\_Sistem\\_Klasifikasi\\_Variabel\\_Tingkat\\_Penerimaan\\_Konsumen\\_Terdapap\\_Mobil\\_Menggunakan\\_Metode\\_Random\\_Forest](https://www.researchgate.net/publication/320413581_Sistem_Klasifikasi_Variabel_Tingkat_Penerimaan_Konsumen_Terdapap_Mobil_Menggunakan_Metode_Random_Forest). [Accessed: 24-Oct-2019].
- [7] J. Han, *Data mining: Data mining concepts and techniques*. 2014.
- [8] L. Breiman, "Random Forests. transparencias," *Statistics (Ber)*, vol. 45, no. 1, pp. 1–33, 2001.
- [9] S. Polamuri, "How the random forest algorithm works in machine learning," 2017. [Online]. Available: <https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learning/>. [Accessed: 24-Oct-2019].