

# PREDIKSI PERFORMA PESEPAKBOLA PADA KOMPETISI LIGA CHAMPIONS EROPA 2021 MENGGUNAKAN ALGORITME *DECISION TREE CLASSIFIER*

Maret Ismoyo<sup>1\*</sup>, Mohammad Syafrullah<sup>2</sup>, Purwanto<sup>3</sup>, Painem<sup>4</sup>

<sup>1,2,3,4</sup>Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur  
Jl. Ciledug Raya, RT.10/RW.2, Petungkang Utara, Pesanggrahan, Jakarta, Indonesia  
e-mail koresponden: maretismoyo@gmail.com

(received: 28/07/2023, revised: 14/08/2023, accepted: 15/08/2023)

## Abstrak

Sepakbola merupakan salah satu olahraga paling terkenal dengan jumlah penggemar terbanyak di seluruh dunia. Selain itu, dalam dunia Sepakbola terdapat berbagai kompetisi yang diikuti oleh banyak pesepakbola profesional. Oleh karena itu, terdapat minat yang cukup besar dari berbagai pihak (internal maupun eksternal) untuk mengevaluasi atau mengukur performa para pesepakbola yang bertanding. Dalam mengevaluasi pemain, beberapa faktor kompleks dapat mempengaruhi staf pelatih dalam menilai pemain. Selain itu, pihak lain juga masih jarang melakukan penelitian terkait prediksi performa pesepakbola. Masalah prediksi tersebut karena membutuhkan pengetahuan dan penggunaan indikator data sesuai dengan topik penelitian tentang prediksi performa atlet sepakbola yang akan dilakukan. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem prediksi yang dapat memprediksi performa pemain dengan menggunakan metode *Decision Tree Classifier* dimana metode ini mengklasifikasikan data menjadi tiga kelas. Data yang digunakan sebanyak 876 data yang terbagi menjadi *data training* dan *testing* dengan tiga kelas performa pemain yaitu *Bad*, *Normal* dan *Good*. Hasil penelitian ini menunjukkan akurasi terbaik diperoleh dengan perbandingan data 80% : 20%, dengan menggunakan nilai parameter *max\_depth* = 6 dan *max\_leaf\_nodes* = 4. Hasil penelitian ini menghasilkan nilai akurasi sebesar 97.73%. nilai presisi sebesar 97.76%. nilai recall sebesar 97.73%. nilai skor F1 sebesar 97.74% dan tingkat kesalahan 2.27%.

**Kata kunci:** *Data Mining*, *Decision Tree Classifier*, Prediksi, Performa Pesepakbola, Sepakbola,

## Abstract

*Football is one of the most famous sports with the most fans worldwide. In the world of football, there are various competitions in which many professional footballers participate. Therefore, there is considerable interest from various parties (internal and external) to evaluate or measure the performance of competing footballers. Several complex factors can influence the coaching staff in assessing players. In addition, other parties rarely conduct research related to predicting footballers' performance. The prediction problem requires knowledge and the use of data indicators by the research topic on predicting football athletes' future performance. Thus, this study proposed to predict player performance using the decision tree classifier method. This method classifies data into three classes. The data used is 876, divided into training and testing data with three classes of player performance, namely Bad, Normal, and Good. This study's results show the best accuracy using the split method with 80%: 20%, using the parameter values *max\_depth* = 6 and *max\_leaf\_nodes* = 4. This study resulted in an accuracy value of 97.73%, a precision value of 97.76%, a recall value of 97.73%, an F1 score value of 97.74%, and an error rate of 2.27%.*

**Keywords:** *Data Mining*, *Decision Tree Classifier*, *Footballer Performance*, *Prediction*, *Soccer*.

## 1. Pendahuluan

Olahraga merupakan salah satu unsur dalam kehidupan manusia yang jika diberdayakan melalui berbagai upaya yang sesuai, maka akan diperoleh hasil dan/atau prestasi yang mengagumkan [1]. Sepakbola adalah salah satu olahraga terkenal dengan jumlah penggemar terbanyak di dunia. Terdapat banyak kompetisi seperti Liga Champions Eropa, Piala Dunia, dan lain-lain yang disaksikan oleh

penggemar dari seluruh dunia. Banyak hasil pengamatan dan juga analisis datang dari berbagai kalangan terkait performa baik para pemain maupun tim secara keseluruhan. Pada dasarnya masing-masing tim punya mekanisme analisis sendiri yang diterapkan guna mengetahui faktor penyebab kesuksesan, maupun kemunduran, kekalahan dan hal lainnya atas tim mereka. Namun berbagai penelitian mencoba menerapkan beberapa pendekatan ilmiah guna melakukan hal tersebut, khususnya untuk mengukur performa dari seorang pesepak bola.

Adapun pihak internal seperti pemilik klub serta staf kepelatihan berkeinginan untuk mengetahui pemain mana yang menjadi aktor keberhasilan timnya dalam memperoleh kemenangan, sehingga proses perekrutan dan rotasi pemain dapat berjalan sesuai dengan kebutuhan klub. Sedangkan pada pihak eksternal, ada fans klub dan media (penyiaran, *online* maupun cetak) yang ingin mengetahui pemain mana saja yang harus diapresiasi atau dikritik performanya selama suatu kompetisi sepakbola berlangsung [2].

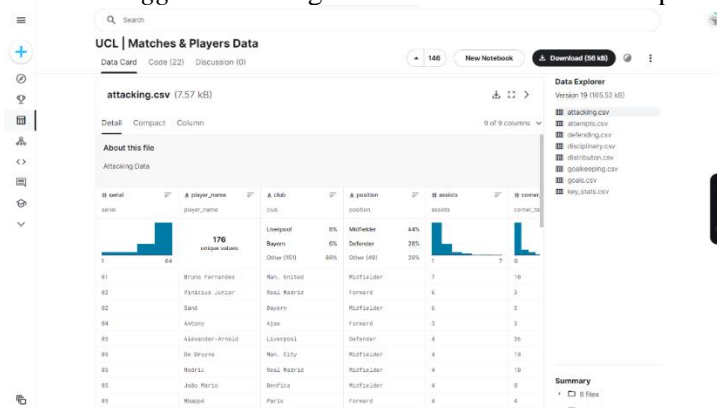
Umumnya, sepakbola adalah olahraga yang membutuhkan kerjasama tim, jadi terdapat faktor-faktor rumit didalamnya yang dapat mempengaruhi pengambilan keputusan seorang manajer/pelatih dalam menilai pemainnya, antara lain: faktor fisik, taktik, mental, teknik serta strategi permainan [3]. Maka dari itu, penting bagi staf kepelatihan untuk menggunakan prediksi yang disertai indikator performa yang mampu menggambarkan aspek-aspek spesifik dari kinerja pemain selama diatas lapangan. Mengingat salah satu tugas utama seorang manajer/pelatih yaitu mengevaluasi performa pemain, sehingga hasil feedback/umpan balik dari staf kepelatihan dapat diberikan kepada para pemainnya [4].

Terdapat beberapa metode yang telah diterapkan pada aktivitas penelitian terdahulu terkait dengan prediksi performa pesepakbola, antara lain dengan algoritme *Naïve Bayes*, *Random Forest* [5], *Support Vector Machine* (SVM), dan *K-Nearest Neighbor* (KNN) [6]. Pada penelitian tersebut, *dataset* yang digunakan adalah daftar statistik 150.000 pemain sepakbola pada gim *Football Manager 2017* yang diperoleh dari situs *Kaggle.com*. Sementara pada penelitian ini, *dataset* yang digunakan adalah 8 file berformat \*.csv yang berisi daftar statistik pemain sepakbola pada kompetisi Liga Champions Eropa 2021/2022 yang diperoleh dari situs *Kaggle.com*. Adapun pada penelitian ini menerapkan algoritme *Decision Tree Classifier* untuk dapat mengklasifikasikan performa pesepakbola ke dalam 3 kelas/*class* yaitu *Bad/Buruk*, *Normal/Biasa*, dan *Good/Bagus*. Adapun pemilihan nama 3 kelas/*class* ini mirip seperti yang digunakan dalam penelitian [7], yaitu *performer*, *moderate*, dan *failure*. Tujuan penggunaan 3 kelas pada penelitian ini yaitu agar model prediksi mampu menentukan kategori kelas/*class* untuk setiap data masukan yang belum diketahui *class*-nya [8].

## 2. Metode Penelitian

### 2.1. Data Penelitian

*Dataset* mentah yang digunakan pada penelitian ini adalah data statistik pesepakbola selama keikutsertaannya pada ajang Liga Champions Eropa musim 2021/2022 yang diperoleh dari situs *Kaggle.com* dengan kata kunci pencarian “UCL 2021-22 | Players Data” yang memuat 8 file berformat \*.csv. Adapun halaman situs *Kaggle.com* sebagai sumber *dataset* mentah dapat dilihat pada gambar 1.

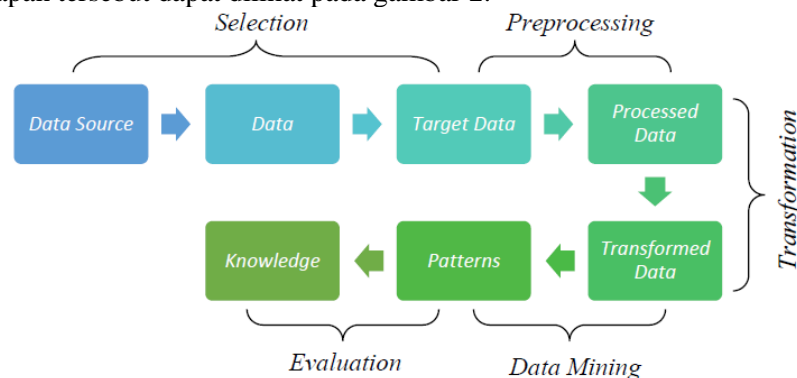


Gambar 1. Tampilan situs *Kaggle.com*

Berdasarkan pada gambar 1, dapat terlihat bahwa terdapat 8 file dengan format \*.csv di sidebar bagian kanan, antara lain *attacking.csv* yang berisi 176 baris data, *attempts.csv* yang berisi 546 baris data, *defending.csv* yang berisi 627 baris data, *disciplinary.csv* yang berisi 584 baris data, *distribution.csv* yang berisi 608 baris data, *goalkeeping.csv* yang berisi 53 baris data, *goals.csv* yang berisi 183 baris data dan *key\_stats.csv* yang berisi 747 baris data.

## 2.2. Penerapan Metode yang Digunakan

Penambangan data atau istilah lainnya *data mining* merupakan sub-bidang ilmu komputer yang memilah/menyortir sekumpulan data dalam jumlah besar untuk mengetahui pola atau tren, dalam rangka memecahkan masalah melalui analisis data [6]. *Data mining* menggunakan fungsi-fungsi statistik seperti varians, standar deviasi, dll; dimana fungsi-fungsi tersebut sangat berharga dalam mempelajari kehandalan serta relasi antar data [9]. Penambangan data sering dianggap identik dengan proses KDD (*Knowledge Discovery in Database*), dimana proses tersebut terdiri dari deretan langkah-langkah yang diterapkan untuk mengekstrak pola-pola unik yang menggambarkan pengetahuan dari data [10]. Di dalam penambangan data, terdapat 5 tahapan yang sering dilakukan dalam analisis data [11]. Adapun 5 tahapan tersebut dapat dilihat pada gambar 2.



Gambar 2. Tahapan Data Mining

### 2.2.1. Seleksi Data/Data Selection

Proses menyeleksi data dilakukan dengan tujuan untuk memilih atribut yang relevan pada data, sehingga proses analisis data untuk menentukan kategori performa pemain dapat dilakukan. Pada penelitian ini, didapatkan 10 atribut dan 1 target atribut yang dibutuhkan, antara lain: *position*, *pass\_accuracy*, *pass\_completed*, *crossing\_accuracy*, *crossing\_completed*, *freekicks\_taken*, *match\_played*, *goals*, *assists*, *distance\_covered*, dan *performance* sebagai target atribut, serta diperoleh data sebanyak 603 baris data.

### 2.2.2. Pra-pemrosesan/Preprocessing

Pra-pemrosesan/*Preprocessing* merupakan suatu tahapan membersihkan data yang telah diseleksi dari duplikasi serta dari data tidak relevan/tidak konsisten. Adapun tahapan pra-pemrosesan yang dilakukan antara lain:

- a. Menghapus *outlier data* / data bias  
Tahap ini menjelaskan bahwa jumlah baris data yang sebelumnya berjumlah 603 data berubah menjadi 522 data Hal itu disebabkan karena *dataset* ini terdapat *outlier* sebanyak 81 data bias. Oleh sebab itu, data bias tersebut harus dihilangkan supaya tidak mempengaruhi kinerja pada tahapan selanjutnya.
- b. Mengecek *missing value* / nilai kosong pada masing-masing atribut  
Sesuai *dataset* yang telah diperiksa, tidak ditemukan *missing value* karena pada keterangan angka *missing value* di semua atribut menunjukkan angka 0, yang berarti semua atribut pada *dataset* ini bersih dari data kosong/*missing value*.

### 2.2.3. Transformasi Data/Data Transformation

Proses transformasi data dilakukan untuk mengubah bentuk data kedalam format yang sesuai dengan algoritme klasifikasi yang diterapkan, dimana algoritme *Decision Tree Classifier* pada penelitian ini diperlukan transformasi data untuk meningkatkan performa model prediksinya. Adapun tahapan transformasi data yang dilakukan antara lain:

#### a. Label Encoding

Tahapan mengubah bentuk data yang sebelumnya berbentuk kategorikal, lalu diubah menjadi bentuk numerik dengan menggunakan fungsi *map()*. Hal ini bertujuan agar komputer mampu membaca data-data kategorik kedalam bentuk integer. Adapun pada penelitian ini, tahapan encoding dilakukan pada atribut *performance* dan atribut *position* yang dapat dilihat pada tabel 1 dan tabel 2.

**Tabel 1.** Encoding atribut *performance*

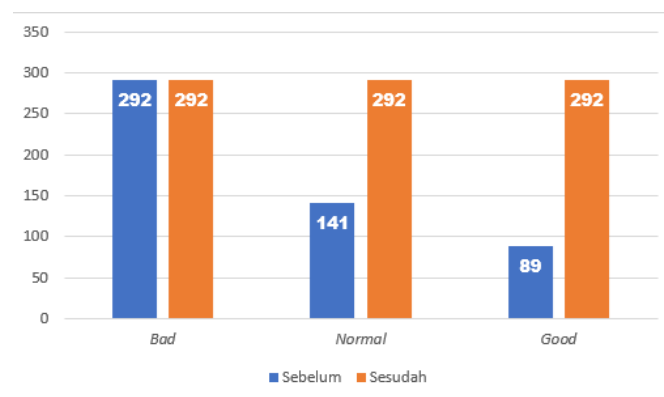
Keterangan	Encoding
<i>Bad</i>	0
<i>Normal</i>	1
<i>Good</i>	2

**Tabel 2.** Encoding atribut *position*

Keterangan	Encoding
<i>Goalkeeper</i>	0
<i>Defender</i>	1
<i>Midfielder</i>	2
<i>Forward</i>	3

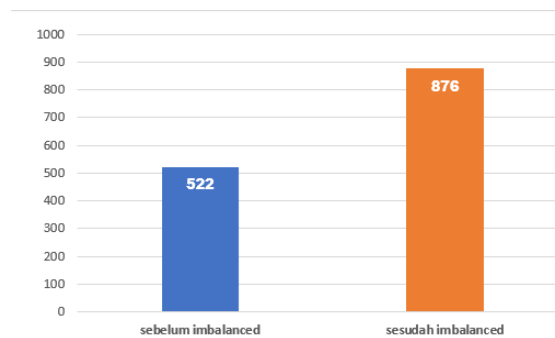
#### b. Handling Imbalanced Data

Tahapan menyeimbangkan jumlah bobot pada atribut-atribut yang memiliki jumlah bobot yang sedikit, dengan cara *balanced* / diseimbangi dengan atribut yang memiliki jumlah bobot terbanyak.



Gambar 3. Perbandingan jumlah data setiap class

Berdasarkan gambar 3, terdapat 2 warna *barchart*/grafik batang yang dimana warna biru merupakan tanda warna untuk data yang belum melalui tahap *handling imbalanced data*, sedangkan yang warna *orange* merupakan tanda warna untuk data yang telah melalui tahap *handling imbalanced data*. Pada grafik berwarna biru, jumlah data pada kelas *bad*, *normal*, dan *good* berbeda-beda, sementara pada grafik berwarna orange, jumlah data pada kelas *bad*, *normal*, dan *good* sudah sama jumlahnya. Hal itu bisa terjadi karena adanya proses *over-sampling*, dimana kelas/class yang jumlah datanya lebih sedikit akan disamakan jumlah datanya dengan kelas/class yang paling banyak jumlah datanya dengan menggunakan data *dummy* pada modul SMOTENC (*Synthetic Minority Over-sampling Technique for Nominal and Continuous*) yang ada pada *library* sklearn. Dengan ditambahkan dengan data *dummy*, akan menyebabkan jumlah data penelitian akan bertambah juga.

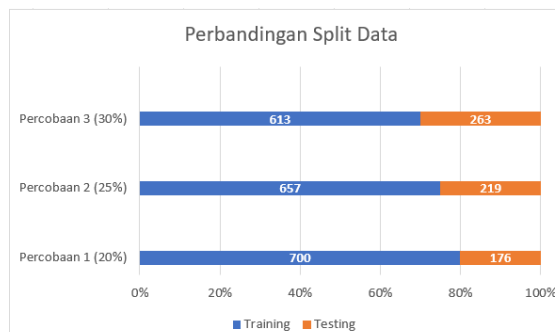


Gambar 4. Perbandingan jumlah data penelitian

Dapat dilihat pada gambar 4, *dataset* yang sebelumnya sebanyak 522 data saat sebelum ditambahkan data *dummy*, akan bertambah menjadi 876 data setelah ditambahkan data *dummy*.

#### c. *Splitting* Data

Tahapan pembagian data latih dan data uji yang akan digunakan pada proses pemodelan/modelling. Penggunaan data training dan data testing pada penelitian ini memiliki porsi yang berbeda-beda pada setiap pengujiannya. Hal ini bertujuan agar model prediksi yang telah dilatih/training dengan menggunakan data training yang lebih banyak porsinya dibandingkan dengan data testing, dapat melakukan proses klasifikasi secara tepat ketika menggunakan masukan/input data baru. Adapun porsi pembagian datanya dapat dilihat pada gambar 5.

Gambar 5. *Splitting* data penelitian

#### d. *Data Normalization*

Tahapan yang bertujuan mengubah *values* dari sebuah feature yang berbeda ukuran skala menjadi feature yang ukuran skalanya sama. Dengan dilakukannya tahapan ini, memungkinkan model memiliki performa dan kestabilan pemrosesan yang lebih baik. Pada penelitian ini, penulis menerapkan tahapan ini dengan menggunakan *library StandardScaler()*. Adapun contoh penerapannya pada atribut di penelitian ini dapat dilihat pada Tabel 3.

Tabel 3. Data Normalization

Atribut	Sebelum	Sesudah
<i>position</i>	2	=> 0.19130654273419248
<i>pass_accuracy</i>	85	=> 1.4677626144204212
<i>pass_completed</i>	165	=> -0.6148431049757687
<i>cross_accuracy</i>	0	=> -0.8821824236284991
<i>cross_completed</i>	0	=> -0.7297656515595429
<i>freekicks_taken</i>	14	=> -0.2843575099627252
<i>match_played</i>	6	=> -0.512483455587928
<i>goals</i>	0	=> 1.4350583528242042
<i>assists</i>	0	=> -0.820562626039796
<i>distance_covered</i>	66.4	=> -0.7326085221454094

### 2.2.4. Penambangan Data/Data Mining

Pemodelan merupakan tahapan utama pada *data mining* setelah data penelitian selesai dibersihkan serta di transformasikan. Tujuan dari tahapan ini yaitu untuk menentukan algoritme yang cocok digunakan pada model prediksi. Dalam tahap pemodelan, dilakukan beberapa pengaturan *tuning* parameter pada setiap algoritme klasifikasi untuk meningkatkan nilai performanya. Pada penelitian ini, penulis menggunakan fungsi *GridSearchCV()* untuk memilih nilai *tuning* parameter terbaik secara otomatis. Adapun contoh pengaturan *tuning* parameter menggunakan *GridSearchCV()* untuk algoritme DTC dapat dilihat pada Tabel 4.

**Tabel 4.** Hasil *GridSearch* algoritme DTC

<i>Hyperparameter Tuning</i>	Hasil nilai <i>Tuning</i> terbaik
<i>max_depth</i>	6
<i>max_leaf_nodes</i>	6

Keterangan:

- Nilai *max\_depth* yang digunakan penulis yaitu: 2-11
- Nilai *max\_leaf\_nodes* yang digunakan penulis yaitu: 2-11
- Semakin banyak parameter yang di *tuning*, maka semakin lama waktu komputasi yang dibutuhkan fungsi *Grid Search()* untuk menentukan nilai *tuning* terbaik.

### 2.2.5. Evaluasi/Evaluation

Setelah melakukan tahapan pemodelan, langkah selanjutnya terdapat tahapan untuk mengevaluasi kinerja dari algoritme klasifikasi yang akan digunakan sebagai model prediksi performa pesepakbola. Pada kasus klasifikasi, untuk mengukur kinerja algoritmenya, perlu melihat nilai-nilai *performance metrics* pada *confusion matrix*. Pada penelitian ini, penulis melakukan 3 kali percobaan dengan menggunakan 5 *performance metrics* pada *confusion matrix*, antara lain: *Accuracy*, *Precision*, *F1-score*, *Recall*, dan *Error rate*.

### 2.3. Rancangan Pengujian

Adapun pada penelitian ini, untuk proses pengujian yang diterapkan yaitu dengan menggunakan metode tabel *confusion matrix* disajikan dalam bentuk tabel matriks yang memvisualisasikan performa model klasifikasi pada serangkaian *data testing* yang nilai aktualnya diketahui [6]. Tabel *confusion matrix* dapat digambarkan seperti pada gambar 6.

		Aktual	
		1 (Positive)	0 (Negative)
Prediksi	1 (Positive)	TP (True Positive)	FP (False Positive)
	0 (Negative)	FN (False Negative)	TN (True Negative)

Gambar 6. Tabel *confusion matrix*

Keterangan:

- *True Positive* (TP) = jumlah nilai *true* yang diprediksi *true*
- *False Positive* (FP) = jumlah nilai *false* tetapi diprediksi sebagai *true*
- *True Negative* (TN) = jumlah nilai *false* yang diprediksi *true*
- *False Negative* (FN) = jumlah nilai *true* tetapi diprediksi sebagai *false*

Adapun beberapa *performance metrics* beserta persamaannya dibawah ini yang umum digunakan untuk menghitung kinerja model dalam kasus klasifikasi [12] yaitu:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$



$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1-score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

Berikut ini adalah penjelasan mengenai 4 *performance metrics* yang umum digunakan [13] :

- *Accuracy* = menunjukkan nilai performa model klasifikasi.
- *Precision* = menunjukkan efisiensi model klasifikasi pada setiap kategori.
- *Recall* = menunjukkan efisiensi model klasifikasi yang mampu diklasifikasikan secara penuh.
- *F1-score* = menunjukkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan.

### 3. Hasil dan Pembahasan

Langkah selanjutnya, dilakukan perhitungan performa dari model yang digunakan sebanyak 3 kali pengujian dengan porsi data latih dan data uji yang berbeda pada setiap pengujian, dengan menerapkan tabel *Confusion Matrix* dan 5 *performance metrics* yaitu: *accuracy*, *precision*, *recall*, *f1-score*, dan *error rate*. Adapun parameter yang digunakan pada ketiga pengujian ini ada 2, yaitu *max\_depth* = 6 dan *max\_leaf\_nodes* = 4.

#### 3.1. Pengujian 1 (80% Data Training:20% Data Testing)

Adapun pada pengujian pertama ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 80% untuk data latih dan 20% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 700 data dan data uji sebanyak 176 data. Adapun tabel *Confusion Matrix* pada pengujian 1 dapat dilihat pada Tabel 5.

Tabel 5. *Confusion Matrix* pengujian 1

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	57 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 58
	<i>Normal</i>	0 (FP) (cell 4)	57 (TP) (cell 5)	2 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 59
	<i>Good</i>	0 (FP) (cell 7)	1 (FN) (cell 8)	58 (TP) (cell 9)	Jml. Performa <i>Good</i> = 59

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 1.

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{172}{172 + 0 + 4} = \frac{172}{176} \times 100\% = 97.73\%$$

$$Precision = \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} = \frac{100\% + 96.61\% + 96.67\%}{3} = 97.76\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{172}{172 + 4} \times 100\% = 97.73\%$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.76\% \times 97.73\%}{97.76\% + 97.73\%} = 97.74\%$$

$$Error rate = 100\% - Accuracy = 100\% - 97.73\% = 2.27\%$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 6.

**Tabel 6.** Pengaruh *hyperparameter tuning* pada pengujian 1

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	97.73 %	97.76 %	97.73 %	97.74 %	2.27 %
Tidak	97.16 %	97.21 %	97.17 %	97.17 %	2.84 %

### 3.2. Pengujian 2 (75% Data Training: 25% Data Testing)

Adapun pada pengujian kedua ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 75% untuk data latih dan 25% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 657 data dan data uji sebanyak 219 data. Adapun tabel *Confusion Matrix* pada pengujian 2 dapat dilihat pada Tabel 7.

**Tabel 7.** *Confusion Matrix* pengujian 2

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	72 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 73
	<i>Normal</i>	0 (FP) (cell 4)	70 (TP) (cell 5)	3 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 73
	<i>Good</i>	0 (FP) (cell 7)	2 (FN) (cell 8)	71 (TP) (cell 9)	Jml. Performa <i>Good</i> = 73

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 2.

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{213}{213 + 0 + 6} = \frac{213}{219} \times 100\% = 97,26\%$$

$$Precision = \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} = \frac{100\% + 95.89\% + 95.94\%}{3} = 97.28\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{213}{213 + 6} \times 100\% = 97.26\%$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.28\% \times 97.26\%}{97.28\% + 97.26\%} = 97.27\%$$

$$Error rate = 100\% - Accuracy = 100\% - 97.26\% = 2.74\%$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 8.



**Tabel 8.** Pengaruh *hyperparameter tuning* pada pengujian 2

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	97.26%	97.28 %	97.26 %	97.27 %	2.74 %
Tidak	96.80%	96.83 %	96.80 %	96.81 %	3.20 %

### 3.3. Pengujian 3 (70% Data Training: 30% Data Testing)

Adapun pada pengujian ketiga ini, *dataset* yang berjumlah 876 data hasil *reshape* akan dibagi menjadi 70% untuk data latih dan 30% untuk data uji, dimana rasio perbandingan untuk data latih sebanyak 613 data dan data uji sebanyak 263 data. Adapun tabel *Confusion Matrix* pada pengujian 3 dapat dilihat pada Tabel 9.

**Tabel 9.** *Confusion Matrix* pengujian 3

		Aktual			
		<i>Bad</i>	<i>Normal</i>	<i>Good</i>	
Prediksi	<i>Bad</i>	86 (TP) (cell 1)	1 (FN) (cell 2)	0 (FP) (cell 3)	Jml. Performa <i>Bad</i> = 87
	<i>Normal</i>	0 (FP) (cell 4)	83 (TP) (cell 5)	5 (FN) (cell 6)	Jml. Performa <i>Normal</i> = 88
	<i>Good</i>	0 (FP) (cell 7)	2 (FN) (cell 8)	86 (TP) (cell 9)	Jml. Performa <i>Good</i> = 88

Berikut ini adalah perhitungan 5 *performance metrics* pada pengujian 3.

$$Accuracy = \frac{TP}{TP + FP + FN} = \frac{255}{255 + 0 + 8} = \frac{255}{263} \times 100\% = 96.96\%$$

$$Precision = \frac{Prec. Bad + Prec. Normal + Prec. Good}{3} = \frac{100\% + 96.51\% + 94.51\%}{3} = 97.01\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{255}{255 + 8} \times 100\% = 96.96\%$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 97.01\% \times 96.96\%}{97.01\% + 96.96\%} = 96.98\%$$

$$Error rate = 100\% - Accuracy = 100\% - 96.95\% = 3.04\%$$

Setelah menghitung 5 *performance metrics*, selanjutnya dilakukan perbandingan performa model jika tidak menggunakan parameter seperti pada Tabel 10.

**Tabel 10.** Pengaruh *hyperparameter tuning* pada pengujian 3

<i>Hyperparameter Tuning</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Error rate</i>
Ya	96.96 %	97.01 %	96.96 %	96.98 %	3.04 %
Tidak	96.58 %	96.65 %	96.59 %	96.59 %	3.42 %

#### 4. Kesimpulan

Berdasarkan penelitian terkait performa pesepakbola yang telah dijalankan dengan menggunakan *algoritme Decision Tree Classifier*, dapat dibuatkan kesimpulan jika penggunaan *hyperparameter tuning* pada algoritme ini memiliki pengaruh yang besar pada nilai performa model prediksi yang digunakan. Adapun algoritme *Decision Tree Classifier* dapat diterapkan untuk studi kasus klasifikasi performa pesepakbola dengan nilai akurasi tertinggi pada pengujian 1 sebesar 97.73% bila menerapkan *hyperparameter tuning*. Jika dibandingkan kalau tidak menerapkan *hyperparameter tuning*, maka nilai akurasinya hanya sebesar 97.16%.

#### Daftar Pustaka

- [1] Ridwan M, "Kondisi Fisik Pemain Sekolah Sepakbola (SSB) Kota Padang," *J. Performa Olahraga*, vol. 5, pp. 65–72, 2020.
- [2] T. Kharrat, I. G. McHale, and J. L. Peña, "Plus–minus player ratings for soccer," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 726–736, 2019, doi: 10.1016/j.ejor.2019.11.026.
- [3] Aditya Akbar Islami, "Keterampilan Psikologis Pemain Sepakbola Ditinjau dari Posisi Bermain dan Kelompok Usia," pp. 1–23, 2021.
- [4] R. Das, B. Jhaharia, and P. K. Das, "Prediction Model of Success and Failure in Football Competitions," *Int. J. Res. Pedagog. Technol. Educ. Mov. Sci.*, vol. 11, pp. 12–19, 2022, doi: 10.55968/uniaca.2022.11.2.3.
- [5] V. C. Pantzalis and C. Tjortjis, "Sports Analytics for Football League Table and Player Performance Prediction," 11th Int. Conf. Information, Intell. Syst. Appl. IISA 2020, 2020, doi: 10.1109/IISA50023.2020.9284352.
- [6] V. C. Pantzalis, "Sports Analytics Algorithms for Performance Prediction," *Int. Hell. Univ.*, no. December, 2019.
- [7] K. Bijukumar, "Artificial intelligence to predict sports performance," *Int. J. Phys. Educ. Sport. Heal.*, vol. 6, no. 3, pp. 238–241, 2019, [Online]. Available: [www.kheljournal.com](http://www.kheljournal.com)
- [8] E. H. Harahap, L. Muflikhah, and B. Rahayudi, "Implementasi Algoritma Support Vector Machine ( SVM ) Untuk Penentuan Seleksi Atlet Pencak Silat," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 3843–3848, 2018.
- [9] Y. L. Deep Madhvani, Ronik Moradiya, "Data Mining System for Selection of Best Basket Ball Team," *Int. Res. J. Eng. Technol.*, vol. 8, no. 6, pp. 3405–3407, 2021.
- [10] G. Theron, "The use of Data Mining for Predicting Injuries in Professional Football Players," *duo.uio.no*, 2020.
- [11] P. Koudoumas, "Sports Analytics Algorithms for Performance Prediction," *Int. Hell. Univ.*, no. January, 2021.
- [12] Y. Min, Y. Yin, and K. Khin, "Comparing the Performance of Machine Learning Algorithms for Human Activities Recognition using WISDM Dataset," *Int. J. Comput.*, vol. 38, pp. 61–72, 2020.
- [13] P. Pugsee and P. Pattawong, "Football match result prediction using the random forest classifier," *ACM Int. Conf. Proceeding Ser.*, pp. 154–158, 2019, doi: 10.1145/3358528.3358593.