Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 4, No. 10, Oktober 2024, Hal. 515-536
DOI: https://doi.org/10.52436/1.jpti.672
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Prediksi Nilai Pasar Pemain Sepak Bola Menggunakan Algoritma Random Forest Berdasarkan Atribut Permainan Dari Game Football Manager 2023 Pada Lima Liga Top Eropa (Berdasarkan Koefisien UEFA)

Syihabuddin Affandi*[1], Eddy Maryanto[2], Yogiek Indra Kurniawan[3]

[1,2,3]Jurusan Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Indonesia
Email: [1]syihabuddin.affandi@mhs.unsoed.ac.id, [2]eddy.maryanto@unsoed.ac.id, [3]yogiek@unsoed.ac.id

Abstrak

Sepak bola bukan hanya sekadar olahraga, tetapi juga industri bernilai miliaran dolar, khususnya di Eropa. Salah satu aspek krusial dalam industri ini adalah penentuan nilai pasar pemain, yang menjadi dasar bagi transaksi transfer pemain. Nilai pasar pemain dipengaruhi oleh berbagai faktor, seperti performa, usia, posisi, serta aspek fisik dan mental. Namun, terdapat kesenjangan dalam penilaian nilai pasar, di mana pemain dengan statistik performa tinggi terkadang memiliki nilai pasar yang lebih rendah dibandingkan pemain dengan performa yang kurang optimal. Oleh karena itu, prediksi nilai pasar pemain secara objektif menjadi tantangan penting bagi klub sepak bola dalam pengambilan keputusan strategis. Penelitian ini mengusulkan model prediksi berbasis *Random Forest* untuk mengestimasi nilai pasar pemain secara objektif dengan memanfaatkan data atribut permainan dari *Football Manager 2023*. Dataset mencakup 1.405 pemain dari lima liga top Eropa (berdasarkan koefisien UEFA 2023) dengan 66 variabel. Metodologi penelitian meliputi tahap *preprocessing* data (*handling missing values,label encoding*), *Exploratory Data Analysis* (*EDA*), pembangunan model *Random Forest*, dan implementasi sistem berbasis web. Pembagian data menggunakan rasio 80:20 (*training-testing*), sementara evaluasi kinerja model dilakukan melalui metrik *RMSE (Root Mean Squared Error)*, *MAE (Mean Absolute Error)*, dan *R^2 (Koefisien Determinasi)*. Hasil eksperimen menunjukkan bahwa model *baseline* dengan parameter default memperoleh nilai *Root Mean Squared Error* (RMSE) sebesar 0.63, *Mean Absolute Error* (MAE) sebesar 0.517, dan koefisien determinasi (R^2) sebesar 0.75. Setelah dilakukan optimasi *hyperparameter* menggunakan *Grid Search*, kinerja model mengalami peningkatan yang signifikan dengan RMSE sebesar 0.62, MAE sebesar 0.513, dan R^2 sebesar 0.76. Model optimal diimplementasikan ke dalam sebuah situs web untuk mempermudah melakukan prediksi nilai pasar pemain. Hasil penelitian menunjukkan bahwa model *Random Forest Regression* mampu memberikan prediksi nilai pasar dengan tingkat akurasi yang lebih baik dibandingkan metode lain yang telah diuji dalam penelitian terdahulu.

Kata kunci: *Football Manager 2023, nilai pasar pemain, prediksi, Random Forest, Tuning Hyperparameter*

Prediction Of Football Player Market Value Using Random Forest Algorithm Based On Game Attributes From Football Manager 2023 Game In Top Five European Leagues (Based On UEFA Coefficient)

Abstract

Football is not just a sport, but also a multi-billion dollar industry, especially in Europe. One of the crucial aspects in this industry is determining the market value of players, which is the basis for player transfer transactions. The market value of players is influenced by various factors, such as performance, age, position, and physical and mental aspects. However, there is a gap in the assessment of market value, where players with high performance statistics sometimes have a lower market value than players with less than optimal performance. Therefore, objectively predicting the market value of players is an important challenge for football clubs in making strategic decisions. This study proposes a Random Forest-based prediction model to objectively estimate the market value of players by utilizing game attribute data from Football Manager 2023. The dataset includes 1,405 players from five top European leagues (based on UEFA coefficients 2023) with 66 variables. The research methodology includes data preprocessing stages (handling missing values, label encoding), Exploratory Data Analysis (EDA), Random Forest model development, and web-based system implementation. Data distribution uses a ratio of 80:20 (training-testing), while model performance evaluation is carried out using the RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (Coefficient of Determination) metrics. The experimental results show that the baseline model with default parameters obtains a Root Mean Squared Error (RMSE) value of 0.63, a Mean Absolute Error (MAE) of 0.517, and a coefficient of

p-ISSN: 2775-4227
e-ISSN: 2775-4219

determination (R^2) of 0.75. After hyperparameter optimization using Grid Search, model performance has increased significantly with an RMSE of 0.62, MAE of 0.513, and R^2 of 0.76. The optimal model is implemented into a website to make it easier to predict the market value of players. The results of the study show that the Random Forest Regression model is able to provide market value predictions with a better level of accuracy than other methods that have been tested in previous studies.

Keywords: *Football Manager 2023, Hyperparameter Tuning, Player Market Value, Prediction, Random Forest*

1. PENDAHULUAN

Sepak bola adalah olahraga permainan yang dimainkan 2 tim yang terdiri dari 11 orang masing-masing tim. Tujuan permainan sepak bola adalah untuk meraih kemenangan dengan cara mencetak skor sebanyak-banyaknya atau mencegah lawan untuk mencetak skor, tim yang mencetak lebih banyak skor dari tim lawan akan mendapatkan kemenangan [1]. Saat ini, sepak bola menjadi salah satu cabang populer yang memiliki pengaruh dalam berbagai aspek dalam kehidupan. Bahkan di eropa, sepak bola menarik jutaan pengikut dan menjadi komponen penting dari industri berskala besar. Beberapa penulis berpendapat bahwa sepak bola bukan sekadar olahraga, melainkan bisnis yang dikelola dengan baik, di mana klub sepak bola dapat disamakan dengan Perusahaan. Hal ini dibuktikan dengan langkah INEOS, Perusahaan Kimia yang dipimpin oleh Sir Jim Ratcliffe, untuk mengakuisisi kepemilikian klub Manchester United hingga 25% saham senilai $1,65 miliar [2]. Dalam industri seperti ini, klub sepak bola menghasilkan pendapatan melalui berbagai aktivitas, salah satunya adalah pembelian & penjualan pemain sepak bola selama musim transfer.

Salah satu elemen terpenting di dalam sepak bola adalah pemain sepak bola yang memiliki peran utama dalam aspek olahraga dan ekonomi dari klub pemain tersebut. Pemain diharapkan dapat membawa manfaat bagi klub di masa depan. Klub sepak bola dapat membangun tim dengan dua cara: mereka dapat merekrut pemain sepak bola di usia yang sangat muda lalu mengikuti pendidikan akademis sampai mereka dapat bergabung dengan tim profesional; atau mereka dapat mengontrak atau meminjamkan pemain profesional melalui transfer window [3]. Hal ini menjadikan pemain sepak bola sebagai aset berharga dalam industri sepak bola di kawasan Eropa, di mana transaksi pembelian atau penjualan pemain seringkali melibatkan jumlah uang yang signifikan.

Istilah yang digunakan dalam sepak bola untuk mengukur seberapa besar aset pemain sepak bola untuk transaksi pembelian atau penjualan pemain adalah market value. Market value atau nilai pasar merupakan nilai intrinsik yang melekat pada pemain dengan berbagai faktor pertimbangan. Banyaknya faktor pertimbangan yang dapat membuat market value pemain sepak bola menjadi lebih tinggi ataupun menjadi lebih rendah [4]. Faktor pertimbangan seperti baik tidaknya performa pemain pada saat pertandingan, posisi pemain, dan usia pemain,

Menurut Bahtra [1], kemampuan fisik pemain, teknik atau taktik, dan kondisi mental atau psikologi pemain adalah komponen yang mempengaruhi performa pemain selama pertandingan. Mengingat sepak bola merupakan olah raga yang sangat kompetitif dan menuntut intensitas tinggi selama pertandingan, kondisi fisik yang prima menjadi sangat krusial. Pemain sepak bola seringkali terlibat dalam berbagai aksi, baik dengan maupun tanpa bola. Pemain sepak bola yang mempunyai statistik performa yang baik maka market value pemain tersebut akan tinggi. Demikian pula, pada posisi pemain, seperti penyerang atau gelandang yang seringkali berkontribusi dalam hal mencetak gol ataupun memberikan assist yang dapat meningkatkan peluang kemenangan tim akan membuat market value pemain tersebut semakin tinggi [4].

Lebih lanjut, penelitian Adam Metelski [5], yang berfokus pada liga sepak bola utama Polandia, Ekstraklasa, menemukan bahwa sebagian besar transfer pemain dari liga tersebut melibatkan pemain berusia antara 21 hingga 24 tahun. Selain itu, biaya transfer tertinggi seringkali tercatat untuk pemain yang berusia 21 tahun ke bawah. Temuan ini mengindikasikan preferensi klub asing untuk merekrut pemain muda dari Ekstraklasa. Hal ini selaras dengan kondisi dalam sepak bola, dimana masa produktif seorang pemain umumnya berkisar antara 23 hingga 28 tahun.

Diskusi mengenai nilai pasar pemain sepak bola sering kali menarik perhatian para penggemar sepak bola diseluruh dunia. Fenomena kesenjangan dalam penilaian nilai pasar seringkali terjadi, dimana terdapat pemain dengan statistik performa yang tinggi namun memiliki nilai pasar yang rendah, dan sebaliknya, pemain dengan performa yang kurang mengesankan terkadang dihargai lebih tinggi oleh klub mereka. Lebih lanjut, bukan hanya performa pemain saja, tetapi juga posisi pemain berpengaruh karena pemain yang bermain di posisi yang lebih mudah untuk mencetak gol atau memberikan assist memiliki nilai pasar yang lebih tinggi; dibandingkan dengan posisi seperti pemain bertahan yang memiliki peran krusial dalam menghalau serangan lawan, juga perlu mendapatkan apresiasi yang setara.

Beberapa penelitian telah dilakukan untuk memprediksi nilai pasar pemain menggunakan pendekatan berbasis data. Misalnya, Al-Asadi dan Tasdemir [6] memprediksi nilai pasar pemain dengan data dari *FIFA*

p-ISSN: 2775-4227
e-ISSN: 2775-4219

20 menggunakan empat model regresi: *Linear Regression*, *Multiple Linear Regression*, *Decision Trees*, dan *Random Forest Regression*. Hasilnya, *Random Forest Regression* menunjukkan kinerja terbaik dengan koefisien determinasi (R^2) sebesar 0,95, *Root Mean Squared Error* (RMSE) sebesar 1.649,92, dan *Mean Absolute Error* (MAE) sebesar 574,874. Studi lain oleh Laros [7] membandingkan *Multiple Linear Regression*, *Support Vector Regression*, dan *Random Forest Regression*, dengan hasil bahwa *Random Forest* kembali unggul (R^2 0,67, RMSE 6,034). Selain itu, penelitian oleh Tama dkk. [8] mengusulkan pendekatan *Bayesian Ensemble* dan *LightGBM*, menghasilkan RMSE 1.069,91 dan MAE 387,44.

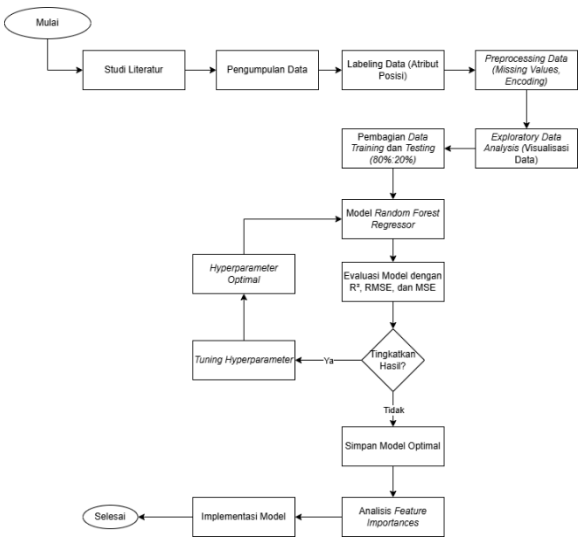
Keunggulan *Random Forest* tidak hanya terbatas pada prediksi nilai pasar pemain, tetapi juga terbukti efektif dalam domain lain. Sebagai contoh, dalam penelitian tentang prediksi harga rumah yang dilakukan oleh Evita Fitri [9], *Random Forest Regression* menghasilkan koefisien determinasi (R^2) sebesar 0,81, *Root Mean Squared Error* (RMSE) sebesar 0.440, dan *Mean Absolute Error* (MAE) sebesar 0.220, mengungguli metode *Linear Regression* dan *Gradient Boosted Trees Regression.*

Metode seperti *Linear Regression* dan *Decision Trees* seringkali kurang mampu menangani kompleksitas hubungan nonlinier antar variabel. *Linear Regression*, misalnya, mengasumsikan hubungan linier antara variabel independen dan dependen, yang jarang terjadi dalam data dunia nyata. Sementara itu, *Decision Trees* rentan terhadap *overfitting*, terutama ketika dataset memiliki banyak fitur. Di sinilah *Random Forest* menawarkan solusi yang lebih robust. Dengan menggabungkan banyak *Decision Trees* dan menerapkan teknik *bagging*, *Random Forest* mampu mengurangi risiko *overfitting* dan meningkatkan generalisasi model. Selain itu, algoritma ini dapat menangani data nonlinier, interaksi antar fitur, serta ketidakseimbangan data dengan lebih baik dibandingkan metode konvensional.

Berdasarkan penjabaran di atas, penelitian ini bertujuan untuk mengembangkan model prediksi nilai pasar pemain sepak bola dengan menggunakan algoritma *Random Forest*. Pemilihan algoritma ini didasarkan pada hasil penelitian sebelumnya yang menunjukkan bahwa *Random Forest* memiliki performa unggul dalam permasalahan prediksi dan regresi. Selain itu, model prediksi yang dihasilkan diimplementasikan ke dalam sistem berbasis web guna memudahkan penggunaannya oleh pencari bakat (scout) dan penggguna game *Football Manager 2023*.

2. METODE PENELITIAN

Penelitian berjudul "Prediksi Nilai Pasar Pemain Sepak Bola Berdasarkan Atribut Permainan di Game *Football Manager 2023* dengan *Random Forest* pada Lima Liga Eropa Terbaik" ini dilakukan dengan mengikuti langkah-langkah yang digambarkan dalam *flowchart* berikut.



Gambar 1. Mekanisme Penelitian

2.1. Studi Literatur

Proses Studi Literatur yang dilakukan pada tahap awal penelitian ini merupakan langkah penting dalam penelitian, karena hasil dari studi literatur akan menjadi dasar dalam menentukan metode, tahapan penelitian, serta batasan yang diterapkan pada setiap langkah penelitian. Dengan demikian, penelitian ini dapat dilakukan secara terarah dan menghasilkan kesimpulan yang memuaskan..

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.2. Pengumpulan Data

Pengumpulan data dilakukan secara manual melalui aplikasi *Football Manager 2023* dengan memilih lima liga terbaik di Eropa berdasarkan koefisien UEFA musim 2022/2023, yaitu *English Premier League*, *Spanish La Liga*, *German 1. Bundesliga*, *Italian Serie A*, dan *French Ligue 1*. Pemilihan liga ini didasarkan pada peringkat resmi UEFA untuk memastikan representasi kompetitif dan kualitas pemain yang lebih homogen. Data yang dikumpulkan mencakup 66 variabel yang dikelompokkan ke dalam lima kategori utama, yaitu *General Stats*, *Personality*, *Physical*, *Mentality*, dan *Technical*. Kategori *General Stats* terdiri dari 17 atribut yang mencerminkan informasi umum pemain, seperti *Player Name*, *Division*, *Club*, dan *Preferred Foot*. Kategori *Personality* memiliki 13 atribut yang menggambarkan karakteristik psikologis dan perilaku pemain, seperti *Adaptability*, *Ambition*, *Loyalty*, dan *Professionalism*. Kategori *Physical* terdiri dari 8 atribut yang merepresentasikan kemampuan fisik pemain, termasuk *Acceleration*, *Agility*, *Balance*, dan *Pace*. Kategori *Mentality* mencakup 14 atribut yang menilai aspek psikologis yang memengaruhi performa pemain di lapangan, seperti *Aggression*, *Anticipation*, *Bravery*, dan *Composure*. Sementara itu, kategori *Technical* terdiri dari 14 atribut yang menilai keterampilan teknis pemain, seperti *Passing*, *Crossing*, *Dribbling*, dan *Finishing*.

Football Manager 2023 dipilih sebagai sumber data utama karena menyediakan cakupan atribut pemain yang lebih komprehensif dibandingkan sumber lain seperti *FIFA 23* atau *Transfermarkt*. Keunggulan ini memungkinkan analisis holistik terhadap faktor-faktor yang memengaruhi nilai pasar pemain, termasuk aspek teknis, fisik, dan psikologis yang sering kali kurang diperhatikan dalam dataset sejenis.

Tabel 1. Perbandingan Dataset

Sumber Data	Jumlah Atribut	Cakupan Atribut	Keterbatasan
Football Manager 2023	66	Teknis, fisik, mental, taktis	Data simulasi (bukan riil)
FIFA 23	30	Teknis, fisik	Tidak mencakup atribut mental
Transfermarkt	15-20	Statistik performa, usia, kontrak	Terbatas pada data historis

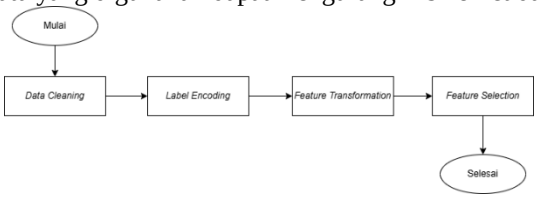
Dataset yang dikumpulkan terdiri dari 1.405 sampel pemain aktif. Sebelum analisis lebih lanjut, data menjalani tahap *preprocessing* untuk memastikan integritasnya, termasuk eliminasi redundansi, penanganan nilai hilang (*missing values*), serta transformasi format. Dataset yang telah diproses kemudian disimpan dalam format *.csv* untuk memfasilitasi proses analisis dan pemodelan lebih lanjut.

2.3. *Labeling Data*

Pada tahap *labeling data*, proses pelabelan dilakukan terhadap atribut posisi pemain. Data atribut posisi ini diperoleh dari situs sofifa.com berdasarkan posisi yang paling sering dimainkan oleh setiap pemain. Klasifikasi posisi pemain dibagi menjadi tujuh kategori utama, yaitu: Centreback (CB), Fullback (FB), Defensive Midfielder (DMF), Central Midfielder (CMF), Attacking Midfielder (AMF), Winger (W), dan Striker (ST). Proses *Labeling Data* dilakukan secara manual menggunakan Microsoft Excel dengan mengevaluasi data pemain dan menentukan posisi terbaik (*best position*) yang akan digunakan dalam penelitian. Pelabelan posisi pemain bertujuan untuk meningkatkan efektivitas dan efisiensi model prediksi nilai pasar pemain sepak bola. Dengan menentukan posisi terbaik pemain, model dapat menghasilkan prediksi yang lebih akurat, karena posisi memiliki pengaruh signifikan terhadap nilai pasar pemain.

2.4. *Preprocessing Data*

Untuk memastikan model Random Forest menghasilkan prediksi yang akurat, dataset akan melalui tahap preprocessing yang bertujuan menjaga keutuhan dan kebersihan data. Proses ini mencakup identifikasi dan penanganan data yang hilang atau tidak lengkap, eliminasi data yang tidak relevan, transformasi data kategorikal menjadi numerik atau sebaliknya menggunakan teknik feature encoding, transformasi atribut menggunakan teknik feature transformation, serta pemilihan atribut terbaik menggunakan korelasi Pearson. Dengan preprocessing ini, diharapkan data yang digunakan dapat mengurangi risiko ketidakakuratan pada model.



Gambar 2. Tahapan Preprocessing

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.4.1. *Data Cleaning*

Data cleaning adalah proses yang bertujuan memastikan keakuratan, konsistensi, dan kegunaan data dalam suatu kumpulan data. Proses ini mencakup deteksi dan penanganan kesalahan data atau data yang korup, serta perbaikan atau penghapusan data sesuai kebutuhan. Pembersihan data merupakan tahap awal dalam proses Knowledge Discovery in Databases (KDD) untuk mendukung penggalian informasi yang bernilai dan bermanfaat [10]. Langkah-langkah yang dilakukan meliputi penanganan nilai yang hilang (missing values), penghapusan data duplikat, eliminasi data yang tidak relevan, serta koreksi data yang mengandung kesalahan. Penanganan data hilang pada penelitian ini dilakukan dengan menggunakan teknik median imputation, yaitu menggantikan data hilang dengan nilai median dari kolom yang bersangkutan. Selain itu, data hilang yang merupakan hasil kesalahan pengolahan file .csv tanpa header dihapus. Pembersihan data juga dilakukan pada data yang memiliki spasi yang tidak diperlukan karena dapat menimbulkan masalah seperti kesulitan dalam pencocokan string.

2.4.2. *Label Encoding*

Transformasi data kategorikal menjadi format numerik dilakukan dengan mengonversi setiap kategori dalam sebuah variabel menjadi nilai integer yang unik [11]. Metode yang digunakan dalam penelitian ini adalah *Ordinal Encoding* dan *Categories Mapping*. *Ordinal Encoding* adalah metode label encoding yang digunakan untuk mengonversi data kategorikal menjadi data numerik, di mana urutan kategori memiliki makna. Teknik ini menggantikan setiap kategori dengan bilangan bulat yang mewakili urutan atau peringkat kategori tersebut. *Mapping Categories* adalah metode label encoding yang menggunakan peta atau kamus (*dictionary*) untuk mengonversi data kategorikal menjadi data numerik. Teknik ini menggantikan setiap kategori dengan nilai numerik yang sesuai berdasarkan peta yang telah ditentukan.

2.4.3. *Feature Transformation*

Proses ini bertujuan untuk memilih, mengembangkan, dan mentransformasikan fitur yang telah ada menjadi atribut baru dengan pendekatan berbasis pengetahuan serta pemetaan matematika. Dengan adanya fitur yang lebih relevan, diharapkan model prediksi dapat mengalami peningkatan akurasi serta menyelesaikan permasalahan secara lebih efektif. Dalam penelitian ini, dilakukan proses *Feature Transformation* dengan menggabungkan atribut yang telah ada sebelumnya guna meningkatkan relevansi dan efisiensi model.

Pendekatan ini memungkinkan sebanyak 49 atribut dari kategori *Personality*, *Physical*, *Mentality*, dan *Technical* untuk diorganisir ulang agar lebih representatif terhadap performa pemain di lapangan serta mendukung kinerja model secara keseluruhan. Kombinasi atribut baru yang digunakan dalam penelitian ini diperoleh dari sumber *Passion4FM.com* dan *GuidetoFM.com*, yang menjelaskan bahwa atribut pemain dapat berinteraksi secara sinergis dalam membentuk keterampilan tertentu. Selama pertandingan, seorang pemain memanfaatkan kombinasi atribut untuk menjalankan berbagai peran, baik dalam aspek ofensif yang bertujuan mencetak gol maupun dalam aspek defensif yang bertujuan mencegah lawan mencetak gol. Oleh karena itu, kombinasi atribut tertentu dapat merepresentasikan kemampuan bermain spesifik yang berkontribusi terhadap efektivitas performa pemain.

Selain rekayasa fitur berbasis atribut gabungan, dilakukan pula proses normalisasi terhadap atribut *Height*, *Weight*, dan *Salary*. Normalisasi atribut fisik bertujuan untuk mengurangi bias yang dapat timbul akibat perbedaan karakteristik fisik antar pemain di posisi yang berbeda. Sebagai contoh, pemain dengan tinggi 170 cm yang berposisi sebagai *Winger* dibandingkan dengan pemain dengan tinggi yang sama yang berposisi sebagai *Centre-back* dapat mengalami perlakuan berbeda dalam penilaian atribut fisiknya. Dengan menerapkan normalisasi, atribut fisik kedua pemain tersebut dapat dibandingkan secara lebih adil. Selain itu, normalisasi atribut *Salary* bertujuan untuk mengurangi bias akibat perbedaan struktur gaji di berbagai liga. Perbedaan ini dapat disebabkan oleh variasi ekonomi antar liga yang memengaruhi standar gaji pemain. Dengan melakukan normalisasi, nilai gaji pemain dari liga yang berbeda dapat dibandingkan dengan lebih objektif, sehingga analisis terhadap nilai pasar pemain menjadi lebih akurat.

2.4.4. *Feature Selection*

Tahap seleksi fitur dilakukan untuk mengevaluasi dan menentukan atribut yang paling relevan dalam pembentukan model prediksi, sehingga hanya fitur dengan kontribusi signifikan yang digunakan dalam proses pemodelan. Seleksi fitur bertujuan untuk mengurangi kompleksitas model, meningkatkan akurasi prediksi, serta mengurangi risiko *overfitting* dengan menghilangkan atribut yang memiliki informasi redundan atau kurang relevan.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Dalam penelitian ini, metode *Feature Selection* diterapkan untuk mengevaluasi kualitas atribut yang telah dipilih sebelumnya. Salah satu teknik yang digunakan adalah analisis korelasi menggunakan *Pearson Correlation Coefficient*, yang mengukur hubungan linier antara masing-masing fitur dengan variabel target. Proses ini didasarkan pada hipotesis heuristik bahwa atribut yang optimal harus memiliki korelasi yang tinggi terhadap variabel target (*market value*), tetapi tidak boleh memiliki korelasi yang terlalu tinggi dengan fitur lain agar tidak menyebabkan multikolinearitas dalam model.

2.5. *Exploratory Data Analysis*

Exploratory Data Analysis (EDA) adalah proses analisis data yang dilakukan untuk memahami karakteristik utama suatu data secara lebih mendalam, terutama melalui metode visualisasi. EDA memungkinkan untuk mendeteksi pola, mengidentifikasi anomali, menguji hipotesis, serta memvalidasi asumsi. Proses ini juga membantu menentukan apakah teknik statistik yang akan digunakan dalam analisis data sesuai. Metode EDA, yang awalnya dikembangkan oleh John Tukey pada tahun 1970-an, tetap menjadi metode penting dalam proses penemuan data hingga saat ini.

2.6. Model *Random Forest*

Pemodelan prediksi nilai pasar pemain sepak bola menggunakan algoritma Random Forest Regression dengan cara membagi data secara acak: 80% sebagai data latih dan 20% sebagai data uji setelah preprocessing. Model ini dibangun menggunakan bahasa pemrograman Python. Selanjutnya, dilakukan penyesuaian Hyperparameter seperti jumlah pohon (n_estimators) dan kedalaman maksimum pohon (max_depth) untuk meningkatkan akurasi prediksi model. Penyesuaian ini bertujuan untuk mengontrol kompleksitas model dan mengurangi risiko overfitting.

2.7. Hasil Evaluasi

Akurasi model Machine Learning diukur berdasarkan hasil prediksi yang diperoleh dari data pengujian. Sebuah model Machine Learning dianggap berkualitas apabila menghasilkan output yang akurat, yang mendukung pengambilan keputusan berdasarkan hasil yang diperoleh [12]. Evaluasi kinerja model dilakukan dengan menggunakan metrik Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), dan koefisien determinasi (R-squared atau R^2). RMSE merupakan akar kuadrat dari Mean Squared Error (MSE) dan berfungsi untuk mengukur rata-rata kuadrat kesalahan prediksi. MAE mengukur rata-rata dari nilai absolut kesalahan prediksi, memberikan gambaran yang jelas tentang seberapa jauh nilai prediksi dari nilai aktual tanpa mempertimbangkan arah kesalahan. R-squared (R^2) adalah metrik yang menunjukkan proporsi variabilitas dalam variabel dependen yang dapat dijelaskan oleh variabel independen dalam model. Nilai R^2 berkisar antara 0 dan 1, di mana nilai yang lebih tinggi menunjukkan bahwa model memiliki kemampuan yang lebih baik dalam menjelaskan variasi data.

2.8. Implementasi Model

Tahap akhir dari penelitian ini yaitu implementasi model yang telah dikembangkan ke dalam bentuk website. Proses implementasi model ke dalam website dilakukan dengan mengekspor model yang telah dibuat ke dalam format file pickle, yang kemudian dimuat kembali melalui kode yang mengintegrasikan model tersebut dengan halaman website yang dirancang. Pengembangan website ini dilakukan dengan memanfaatkan library Flask, yang bertujuan untuk menjaga konsistensi penggunaan bahasa pemrograman Python sebagai backend situs web.

3. HASIL DAN PEMBAHASAN

Hasil dari setiap tahapan metode penelitian yang direncanakan sebelum pelaksanaan penelitian ini adalah sebagai berikut.

3.1. Studi Literatur

Tahap awal dalam penelitian ini adalah melakukan studi literatur untuk mendapatkan pemahaman tentang masalah yang dihadapi serta metodologi yang sesuai. Studi literatur bertujuan untuk mengetahui konsep, metode, dan penelitian sejenis yang dilakukan sebelumnya terkait topik yang diangkat. Fokus studi literatur ini adalah dengan mencari jurnal yang berhubungan dengan kondisi sepak bola di eropa, faktor-faktor yang mempengaruhi

p-ISSN: 2775-4227
e-ISSN: 2775-4219

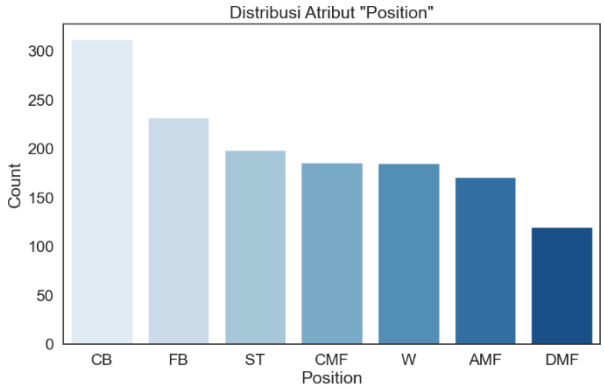
nilai pasar pemain sepak bola, penggunaan machine learning dalam sepak bola, model regresi, serta algoritma *Random Forest*. Pencarian artikel jurnal dibatasi hanya pada artikel yang terbit dalam lima tahun terakhir.

3.2. Pengumpulan Data

Pengumpulan data dilakukan secara manual melalui aplikasi *Football Manager 2023*, yang menyediakan data atribut pemain sepak bola dari lima liga teratas Eropa berdasarkan koefisien pemeringkatan liga oleh UEFA pada musim 2022/2023. Proses ini mengacu pada panduan dari artikel yang diterbitkan di Medium oleh Furkan Ulutaş [13]. Berdasarkan batasan masalah yang telah ditetapkan, pengumpulan data hanya mencakup pemain dengan posisi *outfield player* (bukan penjaga gawang) yang telah berpartisipasi dalam minimal satu pertandingan profesional. Dataset yang dihasilkan mencakup 66 atribut dengan total 1.405 pemain.

3.3. *Labeling Data*

Dataset yang diperoleh dari Football Manager 2023 menunjukkan bahwa pemain dapat bermain di berbagai posisi, termasuk posisi non-natural. Kondisi ini mencerminkan fleksibilitas pemain dalam bermain di beberapa posisi di lapangan. Namun, untuk penelitian ini, diperlukan klasifikasi posisi natural pemain guna meningkatkan akurasi analisis. Klasifikasi posisi pemain dibagi menjadi tujuh kategori, yaitu Centre Back (CB), Full Back (FB), Defensive Midfielder (DMF), Central Midfielder (CMF), Attacking Midfielder (AMF), Winger (W), dan Striker (ST). Data pelabelan posisi natural pemain diperoleh dari situs *sofifa.com* sebagai referensi utama. Persebaran data pada setiap label setelah dilakukan *labeling* dapat dilihat pada Gambar 3.



Gambar 3. Persebaran Hasil Labeling Atribut "Position"

3.4. *Preprocessing Data*

Dataset yang diperoleh dalam format *.csv* melalui tahap *preprocessing* untuk memastikan data dalam kondisi bersih, konsisten, dan bebas dari redundansi. Proses *preprocessing* dilakukan secara sistematis melalui empat langkah utama: *data cleaning*, *label encoding*, *feature engineering*, dan *feature selection*. Seluruh tahapan ini diimplementasikan menggunakan *Jupyter Notebook* dengan bahasa pemrograman Python serta didukung oleh library *Pandas*, *NumPy*, *Seaborn*, *Matplotlib*, dan *Scikit-learn*.

3.4.1. *Data Cleaning*

Proses pembersihan data bertujuan untuk menghilangkan kesalahan atau inkonsistensi dalam dataset. Langkah-langkah yang dilakukan dalam penelitian ini mencakup penghapusan data yang tidak relevan, perubahan nilai, serta perubahan format nilai.

1. Langkah awal dalam *data cleaning* adalah memeriksa nilai yang hilang (*missing values*) menggunakan fungsi isnull(). Fungsi ini memeriksa setiap elemen dalam *dataframe* dan menghasilkan nilai Boolean, di mana *True* menandakan elemen tersebut adalah *NaN* (*Not a Number*), dan *False* menunjukkan elemen tersebut memiliki nilai valid. Selanjutnya, metode sum() digunakan untuk menghitung jumlah nilai *True* pada setiap kolom. Berdasarkan hasil analisis, ditemukan nilai yang hilang pada kolom *Team* serta kolom *Unnamed: 68* yang tidak memiliki nama yang jelas akibat file *.csv* yang tidak dilengkapi dengan *header* yang sesuai. Kolom-kolom tersebut dihapus dari *dataframe* untuk memastikan konsistensi data.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

```
# Memeriksa missing values
missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values[missing_values > 0])
```

```
Missing values in each column:
 Team          719
Unnamed: 68    1400
dtype: int64
```

Gambar 4. Kode Proses memeriksa missing values pada dataframe

2. Pembersihan data selanjutnya dilakukan dengan menghapus spasi yang tidak perlu pada nilai-nilai string dalam *dataframe*. Spasi yang tidak diinginkan dapat menyebabkan masalah dalam analisis data, seperti kesulitan dalam pencocokan string. Langkah ini dilakukan menggunakan fungsi strip() untuk menghilangkan spasi di awal dan akhir nilai string. Sebagai contoh, nama pemain " Philippe Coutinho " diubah menjadi "Philippe Coutinho". Proses ini memastikan konsistensi data dan meminimalkan kesalahan analisis.

```
# Memeriksa Spasi pada Data
def strip_spaces(dataframe):
    for column in dataframe.select_dtypes(include='object').columns:
        dataframe[column] = dataframe[column].str.strip()
    return dataframe

df = strip_spaces(df)
```

Gambar 5. Kode proses memeriksa spasi pada dataframe

3. Pengubahan nama header kolom pada *dataframe* dilakukan menggunakan metode .columns. Sebagai contoh, atribut "Prof" diubah menjadi "Professionalism" untuk meningkatkan kejelasan dan pemahaman terhadap data. Langkah ini bertujuan untuk memastikan interpretasi data yang konsisten serta menghindari potensi kesalahan atau kebingungan dalam analisis.

```
df.columns = ['UID','Name','DOB','Nationality','Division','Club','Preferred Foot','Position','Height','Weight','Age','Transfer Value','Salary',
              'Agreed Playing Time', 'All Time League Appearance','Team','Caps','Adaptability','Ambition','Loyalty','Pressure','Professionalism',
              'Sportsmanship','Temperament','Controversy','Consistency','Dirtiness','Important Match','Injury Prone','Versatility','Acceleration',
              'Agility','Balance','Jumping Reach','Natural Fitness','Pace','Stamina','Strength','Aggression','Anticipation','Bravery','Composure',
              'Concentration','Decisions','Determination','Flair','Leadership','Off The Ball','Positioning','Teamwork','Vision','Work Rate','Corners',
              'Crossing','Dribbling','Finishing','First Touch','Free Kick Taking','Heading','Long Shots','Long Throws','Marking','Passing',
              'Penalty Taking','Tackling','Technique']
df = df.set_index('UID')
df.head()
```

Gambar 6. Kode proses perubahan nama atribut

4. Atribut yang tidak relevan dihapus dari *dataset* untuk meningkatkan kualitas analisis. Dari 66 atribut awal, beberapa di antaranya, seperti "All Time League Appearance," "Caps," dan "Team," diidentifikasi tidak memiliki pengaruh signifikan terhadap nilai pasar pemain. Atribut-atribut ini lebih menggambarkan frekuensi bermain, yang meskipun mencerminkan pengalaman, tidak secara langsung mencerminkan kualitas teknik atau fisik pemain yang menentukan nilai pasar. Sebagai contoh, seorang pemain yang sering tampil tidak selalu memiliki kemampuan teknis atau performa fisik yang luar biasa. Sebaliknya, atribut seperti *dribbling*, *passing*, dan *speed* dianggap lebih relevan dalam menilai nilai pasar secara akurat. Penghapusan atribut yang tidak relevan ini bertujuan untuk mengurangi *noise* dalam data, sehingga analisis lebih fokus pada faktor-faktor yang memiliki korelasi langsung terhadap nilai pasar pemain.
5. Koreksi pada *dataset* dilakukan untuk memastikan konsistensi dan kesesuaian format data. Langkah pertama adalah membersihkan atribut "DOB" (*Date of Birth*) dengan menghapus informasi tambahan yang tidak relevan, seperti usia. Sebagai contoh, data pemain "Philippe Coutinho" yang awalnya tercatat sebagai "6/12/1992 (30 years old)" diubah menjadi "6/12/1992". Selanjutnya, atribut "Weight" (*berat badan*) dan "Height" (*tinggi badan*) dikoreksi dengan menghapus satuan ukur, sehingga nilai hanya menampilkan angka. Sebagai contoh, tinggi badan "172 cm" dan berat badan "71 kg" diubah menjadi "172" dan "71". Langkah ini dilakukan untuk memastikan format data seragam dan menghilangkan karakter yang tidak relevan, sehingga memudahkan proses analisis lebih lanjut.
6. Tahap berikutnya adalah pembersihan dan konversi data pada kolom "Salary" (gaji pemain) dalam *dataframe*. Proses ini bertujuan untuk menyajikan data gaji dalam format yang seragam dan konsisten. Langkah pertama melibatkan penghapusan simbol mata uang, seperti euro (€), teks "p/w" (per minggu), tanda koma (,), dan spasi yang tidak relevan. Selanjutnya, jika ditemukan simbol dolar AS ($), nilai tersebut dikonversi ke euro (EUR) berdasarkan nilai tukar resmi tahun 2022/2023, yaitu 1 USD = 0.969 EUR. Proses ini diimplementasikan menggunakan fungsi convert_salary() untuk memastikan data gaji sesuai dengan standar yang ditetapkan, sehingga dapat digunakan secara efektif dalam analisis lebih lanjut.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

```
# Fungsi untuk membersihkan dan mengonversi salary ke integer
def convert_salary(salary):
    clean_salary = salary.replace('€', '').replace('p/w', '').replace(',', '').strip()

    # Konversi dari USD ke Euro jika ada simbol $
    if '$' in salary:
        # Mengonversi ke integer
        salary_value = int(clean_salary.replace('$', ''))
        euro_value = salary_value * 0.969
        return int(euro_value)
    else:
        return int(clean_salary)

# Terapkan fungsi ke kolom 'Salary'
df['Salary'] = df['Salary'].apply(convert_salary)

# Mengubah tipe data kolom Salary dari float64 menjadi int64
df['Salary'] = df['Salary'].astype('int64')
```

Gambar 7. Kode proses pembersihan dan konversi pada kolom "Salary"

7. Tahap akhir dalam proses pembersihan data melibatkan penanganan format nilai pada atribut "Transfer Value", di mana nilai tersebut diakhiri dengan "M" (juta) atau "K" (ribu). Proses ini memanfaatkan ekspresi reguler untuk mengekstraksi nilai numerik dan mengonversinya menjadi tipe data integer. Langkah ini bertujuan untuk menyatukan data ke dalam format yang konsisten, tanpa memandang apakah nilai awal direpresentasikan dalam jutaan atau ribuan.

```
# Fungsi untuk mengonversi nilai transfer ke bentuk numerik
def convert_transfer_value(value):
    value = value[1:].upper()  # Hilangkan simbol mata uang dan ubah menjadi huruf besar
    if 'M' in value:
        return float(value.replace('M', '')) * 1_000_000
    elif 'K' in value:
        return float(value.replace('K', '')) * 1_000
    return float(value)

# Terapkan fungsi ke kolom 'Transfer Value'
df['new_transfer_value_series'] = df['Transfer Value'].apply(convert_transfer_value)

# Mengubah tipe data kolom new_transfer_value_series dari float64 menjadi int64
df['new_transfer_value_series'] = df['new_transfer_value_series'].astype('int64')
```

Gambar 8. Kode Proses format atribut “Transfer Value”

3.4.2. *Label Encoding*

Dalam Machine Learning, penting dicatat bahwa sebagian besar algoritma beroperasi lebih optimal dengan data bertipe numerik untuk mencapai hasil yang akurat. Oleh karena itu, dalam penelitian ini dilakukan konversi data kategorikal menjadi data numerik guna membangun model yang efektif. Proses Label Encoding diterapkan pada kolom-kolom kategorikal, seperti "Club", "Agreed Playing Time", dan "Preferred Foot".

1. Transformasi kolom "Club" menjadi "Club Rating" dilakukan melalui tiga tahapan utama: standardisasi nama klub, pemberian rating klub, dan penerapan *ordinal encoding*. Proses ini memanfaatkan pustaka *pandas* dan *scikit-learn*. Pertama, standardisasi nama klub dilakukan dengan menyusun kamus untuk mengganti nama klub yang tidak konsisten dengan nama resmi. Sebagai contoh, "Man UFC" diubah menjadi "Manchester United". Pemberian rating klub berdasarkan peringkat yang diterbitkan oleh UEFA untuk musim 2022/2023, di mana klub dengan peringkat 1-20 diberikan rating "5", peringkat 21-40 diberi rating "4", peringkat 41-60 diberi rating "3", peringkat 61-80 diberi rating "2", dan peringkat 81-100 diberi rating "1". Sebagai ilustrasi, Manchester United yang berada di peringkat 7 mendapatkan rating "5". Tahap terakhir adalah penerapan *OrdinalEncoder* untuk mengubah kolom "Rating" menjadi nilai ordinal, sehingga mempermudah analisis data lebih lanjut.

```
# Mendefinisikan club_ratings
club_ratings = {
    5: ['Manchester City','Bayern Munich','Chelsea','Liverpool','Real Madrid','Paris Saint-Germain',
        'Manchester United','Juventus','Barcelona','AS Roma','Inter','Sevilla','Borussia Dortmund',
        'Atlético Madrid','RB Leipzig','Villarreal','Napoli'
       ],

    4: ['Tottenham','Eintracht Frankfurt','Arsenal','Bayer Leverkusen','Olympique Lyon','Atalanta',
        'Milan','Stade Rennais','Lazio','Valencia','Real Betis','Real Sociedad',
        'Olympique de Marseille','LOSC Lille','AS Monaco','VfL Wolfsburg','Newcastle','Brighton',
        'Fiorentina','Athletic Bilbao','Aston Villa','Borussia Monchengladbach',
       ],

    3: ['Brentford', 'Leicester', 'Wolves','Union Berlin','West Ham', 'Torino', 'SC Freiburg',
        'Hoffenheim', 'Sassuolo', 'Espanyol', 'Getafe','RC Lens', 'Bologna',
        'Celta Vigo',  'OGC Nice', 'Vallecano', 'Udinese', 'Verona', '1. FC Köln', 'Everton'
       ],

    2: ['Sampdoria','Cádiz','Hertha Berlin','Crystal Palace', 'Fulham', 'Leeds', 'Brest',
        'Strasbourg', 'Montpellier', 'Nantes', 'Angers SCO', 'Troyes','Clermont', 'Lorient',
        'Mainz 05', 'VfB Stuttgart', 'Almería', 'Osasuna', 'Empoli','Lecce', 'Girona', 'Mallorca',
        'Nottingham Forest','Toulouse FC','FC Lorient','Reims',
       ],

    1: ['Southampton','Schalke 04','VfL Bochum','Cremonese','Salernitana',
        'Brianza', 'Elche',  'Valladolid', 'Spezia', 'AC Ajaccio','FC Nantes',
        'AJ Auxerre', 'SV Werder', 'FC Augsburg', 'Bournemouth']
}
```

Gambar 9. Kode proses pemberian rating klub

2. Transformasi kolom "Agreed Playing Time" melibatkan dua tahap utama: penanganan nilai hilang dan penerapan ordinal encoding. Nilai hilang diatasi dengan menggantinya menggunakan nilai modus, yang diidentifikasi melalui metode .mode(). Nilai "-" yang menandakan status tidak diketahui diganti dengan

p-ISSN: 2775-4227
e-ISSN: 2775-4219

modus menggunakan fungsi .replace(). Selanjutnya, ordinal encoding dilakukan dengan memetakan setiap nilai dalam kolom ke angka ordinal menggunakan dictionary. Proses ini menggunakan objek OrdinalEncoder dan metode fit_transform(), dengan penyesuaian agar nilai encoding dimulai dari 1. Hasil transformasi disimpan dalam kolom baru "APT_Encoded".

```
agreed_playing_time_mapping = {
    'Surplus to Requirements' : 0,
    'Youngster' : 1,
    'Future Prospect' : 2,
    'Breakthrough Prospect' : 3,
    'Fringe Player' : 4,
    'Squad Player' : 5,
    'Impact Sub' : 6,
    'Regular Starter' : 7,
    'Important Player' : 8,
    'Star Player' : 9
}

# Membuat daftar pemetaan dan kolom yang sesuai
mappings = [
    {'col': 'Agreed Playing Time','mapping':agreed_playing_time_mapping,'new_col':'APT_Encoded'}
]

# Proses encoding
for mapping in mappings:
    encoder = OrdinalEncoder(categories=[list(mapping['mapping'].keys())])
    df[mapping['new_col']] = encoder.fit_transform(df[[mapping['col']]]) + 1
```

Gambar 10. Kode proses *Ordinal Encoding* pada kolom "Agreed Playing Time"

3. Proses transformasi pada kolom "Preferred Foot" dilakukan berdasarkan referensi dari repositori GitHub dengan username Pierre Smague, yang memberikan penjelasan terkait kondisi kaki dominan pemain sepak bola, baik kaki kanan maupun kaki kiri. Informasi rinci mengenai kondisi kaki untuk setiap kategori dapat dilihat pada Tabel 1 berikut.

Tabel 2. Nilai pada Kaki Dominan pada pemain sepak bola

No	Kaki Kanan/Kaki Kiri	Value
1	Very Weak	0
2	Weak	4
3	Acceptable	8
4	Fairly Strong	12
5	Strong	16
6	Very Strong	20

Identifikasi nilai pada kolom "Preferred Foot" menggunakan metode *.unique()* mengungkapkan lima nilai unik: "Either", "Left", "Left Only", "Right", dan "Right Only". Berdasarkan Tabel 1, nilai-nilai ini dipetakan sesuai dominasi kaki pemain. Misalnya, "Either" diberi nilai "Very Strong" setara 20. Nilai-nilai ini dijumlahkan dan dirata-rata. Langkah akhir encoding melibatkan pembuatan *dictionary* value_mapping, dengan hasil transformasi disimpan dalam kolom "Preferred_Foot_Encoded".

```
# Definisikan nilai untuk setiap kategori
value_mapping = {
    'Very Weak': 0,
    'Weak': 4,
    'Acceptable': 8,
    'Fairly Strong': 12,
    'Strong': 16,
    'Very Strong': 20
}

# Hitung nilai untuk setiap kategori 'Preferred Foot'
preferred_foot_mapping = {
    'Left': (value_mapping['Acceptable'] + value_mapping['Very Strong']) / 2,
    'Left Only': (value_mapping['Very Weak'] + value_mapping['Strong']) / 2,
    'Right': (value_mapping['Very Strong'] + value_mapping['Acceptable']) / 2,
    'Right Only': (value_mapping['Strong'] + value_mapping['Very Weak']) / 2,
    'Either': (value_mapping['Very Strong'] + value_mapping['Very Strong']) / 2
}

# Ganti nilai dengan bobot manual menggunakan map
df['Preferred_Foot_Encoded'] = df['Preferred Foot'].map(preferred_foot_mapping)
```

Gambar 11. Kode proses Kode proses mapping kolom "Preferred Foot"

3.4.3. *Feature Transformation*

Tujuan dari proses feature transformation adalah menghasilkan fitur baru yang lebih representatif terhadap konteks model yang dikembangkan. Dengan fitur yang lebih relevan, model diharapkan dapat meningkatkan akurasi prediksi serta membantu dalam menyelesaikan permasalahan secara lebih efektif. Fitur baru melalui proses ini sebanyak 17 atribut sehingga jumlah atribut sekarang berjumlah 83 atribut. Atribut tambahan pada proses ini dapat dilihat pada Tabel 2.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

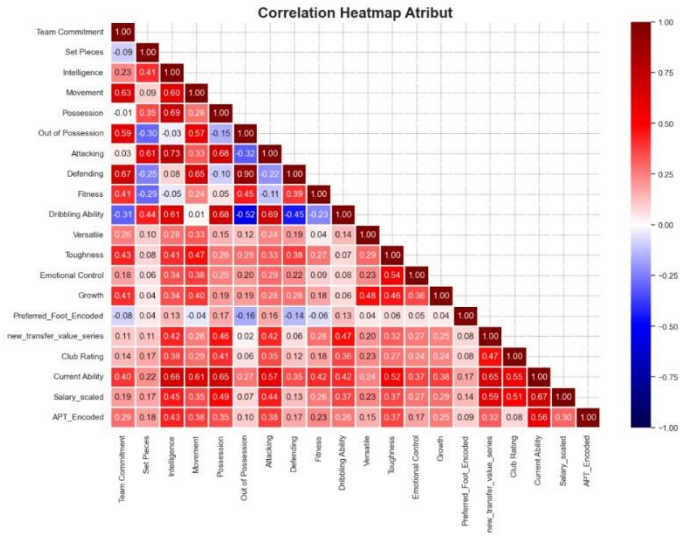
Tabel 3. Atribut hasil *Feature Transformation*

No	Atribut
1	Team Commitment
2	Set Pieces
3	Intelligence
4	Movement
5	Possession
6	Out of Possession
7	Attacking
8	Defending
9	Fitness
10	Dribbling Ability
11	Versatile
12	Toughness
13	Emotional Control
14	Growth
15	Current Ability
16	Height_scaled
17	Weight_scaled

3.4.4. *Feature Selection*

Dalam penelitian ini, *feature selection* dilakukan untuk mengevaluasi kualitas fitur-fitur yang telah terpilih sebelumnya menggunakan *Pearson Correlation Coefficient*. Analisis didasarkan pada hipotesis bahwa atribut optimal harus memiliki korelasi tinggi dengan kelas target. Sebelum melakukan *Feature Selection*, atribut yang termasuk dalam kategori *general features* seperti *Salary_scaled*, *Club Rating*, dan *Age* digabungkan dengan kategori baru seperti "Team Commitment" yang telah dihasilkan dari proses *Feature Transformation.*

Atribut "Age", "Height_scaled", dan "Weight_scaled" dikecualikan karena memiliki karakteristik khusus yang memengaruhi hasil analisis. Sebagai contoh, atribut "Age" tidak menunjukkan hubungan linier konsisten dengan variabel target "Transfer Value", sehingga dianggap tidak relevan. Usia pemain tidak selalu berbanding lurus dengan nilai transfer, mengingat kontribusi pemain muda dan veteran dipengaruhi oleh faktor berbeda, seperti potensi pertumbuhan dan pengalaman. Nilai antar korelasi setiap fitur dapat dilihat pada Gambar 4.



Gambar 12. *Correlation Heatmap*

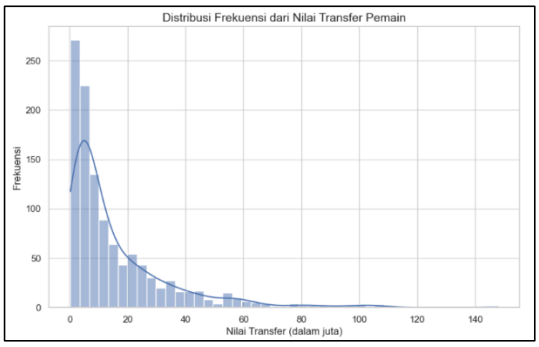
Analisis korelasi dilakukan untuk mengevaluasi hubungan linier antara setiap atribut dengan variabel target "new_transfer_value_series". Atribut dengan korelasi signifikan terhadap variabel target diprioritaskan untuk meningkatkan akurasi prediksi, sementara atribut dengan korelasi rendah dikeluarkan untuk menyederhanakan model dan mengurangi kompleksitas. Fitur dengan nilai korelasi ≥0,3 terhadap variabel target dipertahankan, sedangkan fitur di bawah ambang tersebut dikeluarkan. Hasil feature selection menghasilkan 9 atribut dengan

p-ISSN: 2775-4227
e-ISSN: 2775-4219

korelasi di atas 0,3 yang digunakan dalam proses pemodelan. Daftar atribut yang akan digunakan dalam pembentukan model regresi disajikan pada Tabel 3.

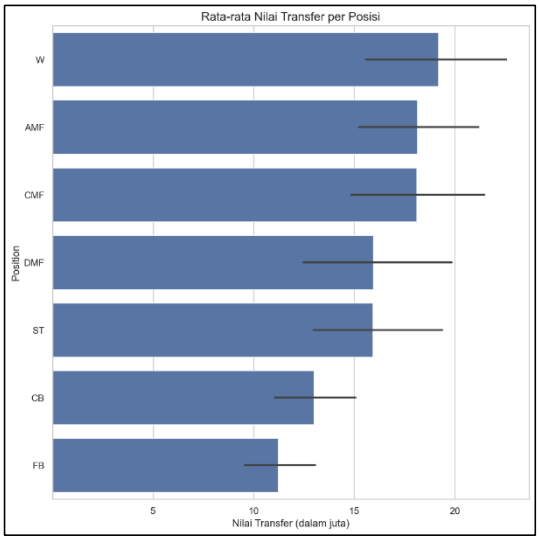
3.5. *Exploratory Data Analysis*

Tujuan utama dari EDA (*Exploratory Data Analysis*) adalah untuk menganalisis data sebelum menarik kesimpulan. EDA berguna untuk mengidentifikasi kesalahan, memahami pola yang ada dalam data, mendeteksi anomali data, serta menemukan hubungan menarik antara berbagai variabel.



Gambar 13. Plot Histogram Distribusi Nilai Pasar Pemain

Distribusi data nilai pasar pemain menunjukkan kemiringan ke kanan, dengan frekuensi tertinggi pada kisaran nilai pasar sekitar 10 juta. Frekuensi menurun secara bertahap seiring peningkatan nilai pasar, mencerminkan jumlah pemain yang lebih sedikit dengan nilai pasar tinggi. Dari distribusi ini, terlihat bahwa sebagian besar pemain memiliki nilai pasar di bawah 20 juta, dengan jumlah pemain yang menurun seiring meningkatnya nilai pasar. Fenomena ini wajar dalam industri sepak bola, karena hanya sedikit pemain bintang yang memiliki nilai transfer sangat tinggi, sementara mayoritas pemain berada dalam kisaran harga yang lebih rendah.



Gambar 14. Rata-rata nilai pasar per Posisi

Grafik batang (bar chart) menampilkan rata-rata nilai pasar pemain berdasarkan posisi, dengan garis horizontal yang merepresentasikan rentang kesalahan atau deviasi standar. Hasil analisis menunjukkan bahwa posisi Striker (ST) dan Attacking Midfielder (AMF) memiliki rata-rata nilai pasar tertinggi, berkisar antara 15 hingga 20 juta. Sementara itu, posisi Fullback (FB) dan Center Back (CB) memiliki rata-rata nilai pasar yang lebih rendah, yaitu sekitar 5 hingga 10 juta. Posisi Winger (W) dan Attacking Midfielder (AMF) menunjukkan nilai pasar rata-rata yang lebih tinggi dibandingkan posisi lainnya. Hal ini mengindikasikan bahwa klub sepak bola cenderung mengalokasikan anggaran lebih besar untuk merekrut pemain yang memiliki kemampuan menyerang yang dominan, khususnya di sektor sayap dan gelandang serang. Pemain di posisi ini umumnya memiliki peran penting dalam mencetak gol dan menciptakan peluang, sehingga permintaan pasar terhadap

p-ISSN: 2775-4227
e-ISSN: 2775-4219

mereka lebih tinggi. Di sisi lain, posisi Center Back (CB) memiliki nilai pasar rata-rata yang lebih rendah dibandingkan pemain dengan peran menyerang. Fenomena ini dapat dikaitkan dengan kecenderungan pasar yang lebih memprioritaskan pemain ofensif dibandingkan pemain bertahan. Meskipun pemain bertahan memiliki peran krusial dalam menjaga keseimbangan tim, nilai pasarnya cenderung lebih stabil dan tidak mengalami lonjakan harga sebesar pemain di lini serang. Hasil ini sejalan dengan tren transfer dalam industri sepak bola, di mana pemain dengan kontribusi ofensif yang signifikan sering kali memiliki nilai pasar yang lebih tinggi dibandingkan pemain bertahan atau pemain dengan peran yang lebih defensif.

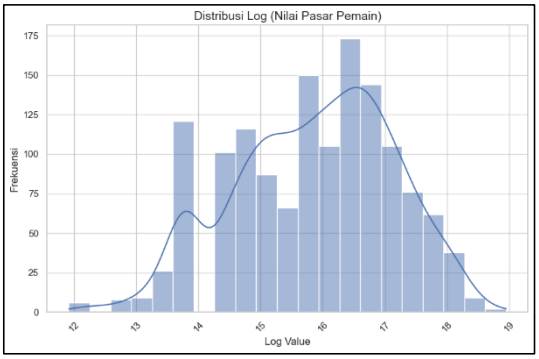
min	150000
25%	3000000
50%	9000000
mean	15308220
75%	20000000
max	170000000

Gambar 15. Deskriptif Variabel "Transfer Value"

Hasil analisis menunjukkan bahwa nilai pasar pemain sepak bola memiliki rentang yang luas, dengan nilai minimum sebesar 150.000 euro dan nilai maksimum mencapai 170.000.000 euro. Nilai median tercatat sebesar 9.000.000 euro, sedangkan rata-rata nilai pasar pemain adalah 15.308.220 euro. Distribusi nilai pasar menunjukkan pola yang tidak merata, di mana mayoritas pemain memiliki nilai pasar yang relatif rendah, sementara hanya sebagian kecil pemain yang memiliki nilai pasar yang sangat tinggi. Hal ini mencerminkan adanya disparitas ekonomi yang signifikan dalam industri sepak bola, di mana pemain bintang dengan performa unggul cenderung memiliki nilai pasar yang jauh lebih tinggi dibandingkan pemain dengan kontribusi yang lebih moderat.

3.6. *Model Random Forest*

Sebelum menerapkan algoritma Random Forest untuk pemodelan, nilai pasar pemain ditetapkan sebagai variabel target. Nilai pasar menunjukkan variasi signifikan, terutama untuk pemain top dengan nilai yang dapat meningkat secara eksponensial. Berdasarkan analisis distribusi, nilai pasar memiliki kemiringan ke kanan (right-skewed distribution), yang dapat menyulitkan proses prediksi, terutama pada pemain dengan nilai transfer tinggi dan bervariasi. Untuk mengatasi masalah ini, dilakukan transformasi logaritmik, yaitu metode yang menerapkan fungsi logaritma pada setiap nilai data. Transformasi ini bertujuan untuk mengurangi kemiringan distribusi (skewness), meminimalkan dampak nilai ekstrem, dan meningkatkan akurasi model.



Gambar 16. Plot Histogram distribusi Nilai pasar setelah Transformasi Logaritmik

Tahapan selanjutnya dalam penelitian ini adalah penentuan variabel independen dan dependen. Sebanyak dua belas variabel independen (x) digunakan, yaitu *Age*, *Height_scaled*, *Weight_scaled*, *Current Ability*, *Possession*, *Intelligence*, *Club Rating*, *Attacking*, *Dribbling Ability*, *Toughness*, *Salary_scaled*, dan *APT_Encoded*. Variabel dependen (y) adalah nilai pasar pemain yang telah ditransformasikan menggunakan logaritma. Data kemudian dibagi menjadi dua komponen utama, yaitu variabel atribut (x) yang mencakup seluruh variabel independen yang telah diproses sebelumnya, serta variabel target (y), yang hanya terdiri dari nilai logaritma dari nilai pasar pemain. Proses ini bertujuan untuk memastikan pemisahan yang jelas antara fitur dan label guna mendukung pemodelan yang efektif.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Pada tahap pemodelan *Random Forest*, data dibagi menjadi dua bagian utama: data pelatihan (training) dan data pengujian (testing). Data pelatihan digunakan untuk membangun model yang dapat mengenali pola dan memprediksi variabel dependen, sedangkan data pengujian berfungsi untuk mengevaluasi kinerja model yang dihasilkan, termasuk dalam mengukur akurasi prediksi. Dalam penelitian ini, pembagian data dilakukan dengan proporsi 80% untuk data pelatihan dan 20% untuk data pengujian, guna memastikan distribusi yang representatif dalam proses pelatihan dan pengujian model.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import numpy as np

# Membagi data menjadi training (80%) dan testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
```

Gambar 17. Kode proses pembagian dataset

Pada tahap awal pengujian, pemodelan Random Forest dilakukan menggunakan parameter default yang disediakan oleh Random Forest Regressor. Secara default, jumlah pohon *(n_estimators)* yang digunakan adalah 100, sementara jumlah fitur yang dipilih secara acak untuk proses split *(max_features)* diatur pada nilai auto, yang berarti akar kuadrat dari jumlah total fitur. Penggunaan parameter *random_state* untuk memastikan bahwa hasil yang diperoleh dapat direproduksi setiap kali kode dijalankan. Konfigurasi default parameter ini ditunjukkan pada Gambar 18.

```
# Membuat model Random Forest Regression
model = RandomForestRegressor(random_state=42)
# 1. Training model
model.fit(X_train, y_train)
```

Gambar 18. Kode proses *Model Random Forest* dengan *Parameter Default*

Fungsi *fit()* digunakan untuk melatih model dengan data pelatihan. Selama proses pelatihan, model *Random Forest* akan membangun beberapa pohon keputusan (decision tree) berdasarkan subset acak dari data pelatihan dan menghitung prediksi untuk setiap subset. Tahap berikutnya adalah evaluasi model Random Forest dengan parameter default menggunakan dua set data: training dan testing. Pada data training, skor R-squared (R^2) dihitung untuk menilai kemampuan model dalam mempelajari pola dari data. Berdasarkan Gambar 19, skor R-squared (R^2) pada data pelatihan mencapai 0.96, menunjukkan bahwa model Random Forest dengan parameter default dapat menangkap sebagian besar variabilitas data pelatihan.

```
# Evaluasi pada data training
train_score = model.score(X_train, y_train)
print("Training R^2 Score:", train_score)

Training R^2 Score: 0.9653158727370943
```

Gambar 19. Kode proses skor R-squared pada data training

Untuk mengukur kemampuan generalisasi model terhadap data baru, dilakukan evaluasi komprehensif pada data testing menggunakan metrik seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan R-squared (R^2).

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Membuat prediksi menggunakan model
y_pred = model.predict(X_test)

# Menghitung metrik evaluasi
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)
accuracy = r_squared * 100

# Menampilkan metrik evaluasi
print('Mean Absolute Error:', mae)
print('Mean Squared Error:', mse)
print('Root Mean Squared Error:', rmse)
print('R-squared:', r_squared)
print('Accuracy (R-squared %):', accuracy)

Mean Absolute Error: 0.4917180948465008
Mean Squared Error: 0.37030219799609326
Root Mean Squared Error: 0.6085246075518173
R-squared: 0.7740237813619801
Accuracy (R-squared %): 77.40237813619801
```

Gambar 20. Kode proses Kode proses hasil Metriks Evaluasi pada *data testing*

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Hasil Evaluasi model *Random Forest* dengan parameter default menunjukkan hasil MAE sebesar 0.4917 yang berarti rata-rata prediksi model meleset sekitar 0.4917 unit logaritma dari nilai aktual. Nilai MSE menunjukkan bahwa rata-rata kuadrat kesalahan prediksi adalah 0.3703. Nilai RMSE berarti, rata-rata, prediksi model meleset sekitar 0.6085 unit dari nilai aktual. Semakin kecil RMSE, semakin baik model. Lalu, R-squared mengukur proporsi variabilitas dalam variabel target yang dapat dijelaskan oleh model. Nilainya berkisar antara 0 dan 1, di mana 1 menunjukkan model yang sempurna. Nilai R^2 sebesar 0.7740 berarti model menjelaskan sekitar 77.40% variasi dalam nilai pasar pemain.

Hasil evaluasi model Random Forest dengan parameter default menunjukkan akurasi 77.40% mengindikasikan potensi peningkatan performa. Oleh karena itu, dilakukan optimalisasi lebih lanjut melalui hyperparameter tuning untuk menentukan kombinasi parameter optimal guna meningkatkan kinerja model.

3.6.1. *Tuning Hyperparameter*

Tuning Hyperparameter adalah proses yang bertujuan untuk menemukan nilai parameter optimal bagi model. Pengoptimalan parameter dalam *machine learning* sangat penting untuk meningkatkan kualitas prediksi dan akurasi model, di mana *hyperparameter* mendefinisikan karakteristik model yang dapat mempengaruhi akurasi. Dalam *Random Forest*, beberapa parameter dapat disesuaikan untuk meningkatkan performa model. Dua parameter utama yang akan dituning dalam penelitian ini adalah jumlah pohon yang diwakili oleh *n_estimators* dan jumlah fitur acak yang digunakan untuk setiap *split*, yang diwakili oleh *max_features*. Selain itu, parameter *max_depth* juga akan dipertimbangkan untuk mengatur kompleksitas *decision tree* dan mencegah *overfitting*.

```
# Hyperparameter n_estimators
n_estimators = [int(x) for x in np.linspace(start=200, stop=1000, num=5)]

# Hyperparameter max_features
max_features = [1.0, 'sqrt', 'log2']

# Hyperparameter max_depth
max_depth = [10, 20, 30, None]

# Hyperparameter min_samples_split dan min_samples_leaf
min_samples_split = [2, 5, 10]
min_samples_leaf = [1,2,3]

# Membuat dictionary param_grid
param_grid = {
    'n_estimators': n_estimators,
    'max_features': max_features,
    'max_depth': max_depth,
    'min_samples_split': min_samples_split,
    'min_samples_leaf': min_samples_leaf,
    'bootstrap': [True, False]
}

pprint(param_grid)
```

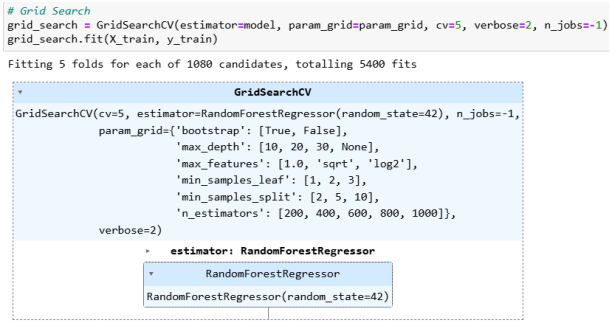
```
{'bootstrap': [True, False],
 'max_depth': [10, 20, 30, None],
 'max_features': [1.0, 'sqrt', 'log2'],
 'min_samples_leaf': [1, 2, 3],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [200, 400, 600, 800, 1000]}
```

Gambar 21. Kode proses Pemilihan Sampel Parameter

Penjelasan mengenai masing-masing parameter adalah sebagai berikut: *n_estimators* menentukan jumlah pohon keputusan yang dibangun dalam model Random Forest, di mana jumlah pohon yang lebih banyak dapat meningkatkan akurasi. *max_features* mengontrol jumlah fitur yang dipertimbangkan pada setiap *split* untuk menemukan pemisahan terbaik. *max_depth* menetapkan kedalaman maksimum setiap pohon dalam model Random Forest. *min_samples_split* menentukan jumlah minimum sampel yang diperlukan untuk membagi suatu simpul dalam pohon keputusan, sedangkan *min_samples_leaf* menetapkan jumlah minimum sampel yang harus ada di setiap simpul daun. Terakhir, *bootstrap* digunakan untuk menentukan apakah bootstrap sampling diterapkan.

Terdapat dua pendekatan umum dalam *tuning hyperparameter*, yaitu *grid search* dan *random search*. Penelitian ini menerapkan metode *grid search*, yang mengevaluasi semua kombinasi parameter yang telah ditentukan sebelumnya. Meskipun metode ini cenderung memakan waktu lebih lama karena menguji setiap kombinasi, *grid search* memiliki potensi tinggi untuk menemukan parameter optimal. Oleh karena itu, metode ini dianggap lebih unggul dibandingkan *random search*.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



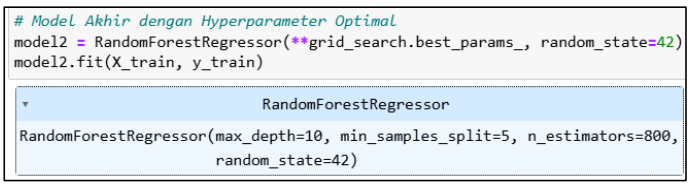
Gambar 22. Kode proses GridSearchCV

Untuk mengevaluasi total kombinasi hyperparameter yang mungkin, digunakan fungsi GridSearchCV dari Scikit-Learn. Salah satu argumen dalam GridSearchCV adalah *cv*, yang menentukan jumlah lipatan (folds) dalam cross-validation. Data akan dibagi menjadi 5 subset (80% training, 20% validation dalam setiap iterasi). *n_jobs=-1* memiliki arti yaitu menggunakan semua core CPU yang tersedia untuk mempercepat proses pencarian. Dalam penelitian ini, parameter yang diuji disusun dalam sebuah dictionary bernama "param_grid" yang mencakup konfigurasi yang telah dijelaskan sebelumnya.

```
pprint(grid_search.best_params_)
{'bootstrap': True,
 'max_depth': 10,
 'max_features': 1.0,
 'min_samples_leaf': 1,
 'min_samples_split': 5,
 'n_estimators': 800}
```
Gambar 23. Parameter Optimal Hasil *Tuning Hyperparameter*

Tahap selanjutnya adalah menggunakan perintah *grid_search.best_params* untuk mengidentifikasi parameter optimal hasil tuning hyperparameter. Berdasarkan hasil GridSearchCV, parameter optimal untuk model ini adalah: *n_estimators* sebanyak 800 pohon, *max_features* diatur ke nilai 1.0, *max_depth* sebesar 10, *min_samples_leaf* sebesar 1, *min_samples_split* sebesar 5, dan *bootstrap* diatur ke nilai True. *bootstrap*:*True* artinya model menggunakan *bootstrap sampling*, yaitu mengambil sampel secara acak dengan penggantian (*replacement*) saat membangun setiap pohon dalam hutan yang dapat meningkatkan prediksi karena mengurangi *variance*. *max_depth:10* artinya kedalam maksimum setiap pohon dalam hutan adalah 10 tingkat dengan anggapan jika terlalu dalam model akan mengalami *overfitting* sedangkan jika terlalu dangkal model tidak cukup fleksibel (*underfitting*). *max_features*:*1.0* artinya model menggunakan 100% fitur (semua fitur dalam dataset) untuk membangun setiap pohon karena jumlah fitur dalam dataset tidak terlalu besar dan setiap fitur memiliki korelasi tinggi sehingga dapat memberikan performa yang optimal. *min_samples_leaf:1* artinya setiap daun (leaf node) dalam pohon minimal berisi 1 sampel, memungkinkan pohon untuk menangkap pola detail dalam data. *min_samples_split:5* artinya sebuah node akan dibagi (split) hanya jika mengandung setidaknya 5 sampel untuk mencegah pemisahan (splitting) yang terlalu cepat sehingga model tidak terlalu kompleks. *n_estimators:800* model membangun 800 pohon dalam *Random Forest* karena membangun banyak pohon dapat menstabilkan model tetapi dalam batas wajar.

Parameter optimal yang telah ditentukan diterapkan kembali dalam pelatihan model Random Forest untuk memprediksi nilai pasar pemain sepak bola. Pada percobaan kedua, model yang telah dioptimalkan diaplikasikan pada data testing.



Gambar 24. Kode Proses Model *Random Forest* dengan Parameter Optimal

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Dengan menggunakan metode .predict(), model menghasilkan prediksi untuk variabel target berdasarkan input data testing. Hasil prediksi disimpan dalam variabel *y_pred_model2*, yang digunakan untuk evaluasi kinerja model terhadap data yang belum pernah dilihat sebelumnya.

```
# Prediksi pada Testing Set
y_pred_model2 = model2.predict(X_test)
```

Gambar 25. Kode Proses prediksi pada Testing Set

Evaluasi model Random Forest Regressor dengan parameter hasil tuning hyperparameter, seperti ditampilkan pada Gambar 27, menghasilkan nilai MAE sebesar 0.4916, nilai MSE sebesar 0.3635, nilai RMSE sebesar 0.6029, nilai Nilai R^2 sebesar 0.7782 dan Akurasi (R-squared %) sebesar 77.82%

```
# Prediksi pada Testing Set
y_pred_model2 = model2.predict(X_test)

# Evaluasi Model dengan Berbagai Metrik
mae = mean_absolute_error(y_test, y_pred_model2)
mse = mean_squared_error(y_test, y_pred_model2)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred_model2)
accuracy_r = r_squared * 100

print('Mean Absolute Error (MAE):', mae)
print('Mean Squared Error (MSE):', mse)
print('Root Mean Squared Error (RMSE):', rmse)
print('R-squared:', r_squared)
print('Accuracy (R-squared %):', accuracy_r)
```

```
Mean Absolute Error (MAE): 0.4916612340375558
Mean Squared Error (MSE): 0.36349767042192804
Root Mean Squared Error (RMSE): 0.602907679849849
R-squared: 0.778176231493654
Accuracy (R-squared %): 77.8176231493654
```

Gambar 26. Hasil Evaluasi Model Random *Forest* dengan Parameter Optimal

Tabel di bawah ini menyajikan perbandingan hasil evaluasi antara model dengan parameter *default* dan model setelah dilakukan *tuning hyperparameter*.

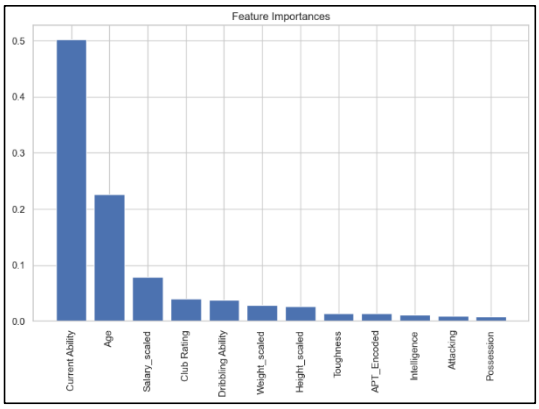
Tabel 4. Perbandingan hasil evaluasi antara model dengan parameter default dan model yang telah di-tuning hyperparameter.

Metrik Evaluasi	Model Default (model1)	Model Tuned (model2)	Perbedaan
MAE	0.4917	0.4916	Lebih kecil (lebih baik) pada model2
MSE	0.3703	0.3635	Lebih kecil pada model2
RMSE	0.6085	0.6029	Lebih kecil pada model2
R^2	0.7740	0.7782	Lebih tinggi pada model2 (lebih baik)
Akurasi (R^2 %)	77.40%	77.82%	Lebih baik pada model2

3.6.2. *Feature Importances*

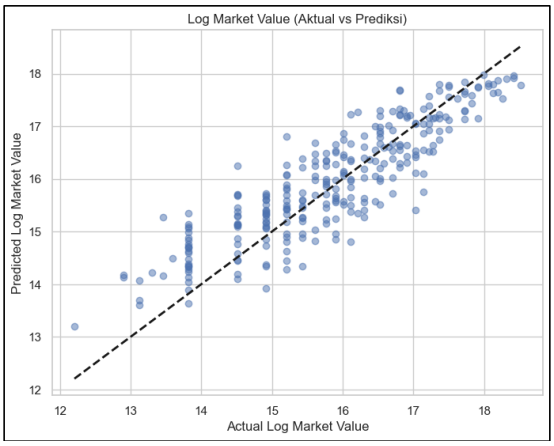
Feature Importances adalah teknik yang digunakan untuk mengukur kontribusi relatif dari setiap fitur dalam model terhadap prediksi yang dihasilkan. Dalam konteks Random Forest, *feature importance* dihitung berdasarkan seberapa banyak setiap fitur meningkatkan akurasi model ketika digunakan dalam pembentukan pohon keputusan.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



Gambar 27. *Feature Importances*

Berdasarkan analisis *feature importances*, variabel *Current Ability* menunjukkan kontribusi terbesar terhadap variabel target dengan pengaruh sekitar 0,5. Selanjutnya, variabel *Age* menempati posisi kedua dengan pengaruh sekitar 0,25, diikuti oleh variabel *Salary_scaled* yang berada di posisi ketiga dengan pengaruh sebesar 0,1. Variabel *Club Rating* dan *Dribbling Ability* masing-masing memberikan kontribusi sekitar 0,05. Sementara itu, variabel *Weight_scaled*, *Height_scaled*, *Toughness*, *APT_Encoded*, *Intelligence*, *Attacking*, dan *Possession* memiliki pengaruh terkecil, yaitu di bawah 0,05. Berdasarkan hasil feature importance, *Current Ability* merupakan fitur utama dalam model, di mana jika fitur ini dihapus, akurasi model akan menurun secara drastis karena model sangat bergantung pada informasi tersebut. *Age* juga memiliki kontribusi yang signifikan sebagai faktor pendukung, namun model masih dapat bekerja tanpanya dengan sedikit penurunan akurasi. Sementara itu, fitur lainnya memiliki kontribusi yang lebih kecil tetapi tetap membantu model dalam membuat prediksi. Jika beberapa fitur dengan importance rendah dihapus, akurasi model kemungkinan tidak akan berubah secara signifikan, tetapi jika banyak fitur kecil dihapus sekaligus, akurasi dapat mengalami sedikit penurunan. Selain itu, fitur dengan kontribusi minimal atau hampir nol dapat dipertimbangkan untuk dihapus guna menyederhanakan model tanpa mengorbankan performa.



Gambar 28. *Scatter Plot* Nilai Log Market Value Aktual vs Prediksi

Selanjutnya, scatter plot digunakan untuk menggambarkan hubungan antara nilai "Log Market Value" aktual dan nilai "Log Market Value" yang diprediksi oleh model, seperti yang ditampilkan pada Gambar 19. Berdasarkan analisis scatter plot tersebut, model prediksi menunjukkan performa yang cukup baik, dengan mayoritas titik prediksi berada dekat dengan nilai aktual. Meskipun terdapat beberapa titik yang menyimpang dari garis referensi, sebagian besar prediksi menunjukkan kesesuaian yang baik. Secara keseluruhan, terdapat tren positif yang mengindikasikan bahwa peningkatan nilai aktual diikuti oleh peningkatan nilai prediksi.

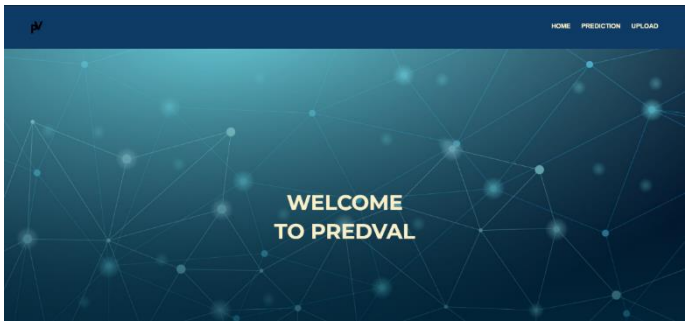
3.7. Implementasi Model

Tahap selanjutnya adalah membangun situs web menggunakan framework Flask dan bahasa pemrograman Python. Flask merupakan web framework berbasis Python yang menyediakan pustaka serta sekumpulan kode

p-ISSN: 2775-4227
e-ISSN: 2775-4219

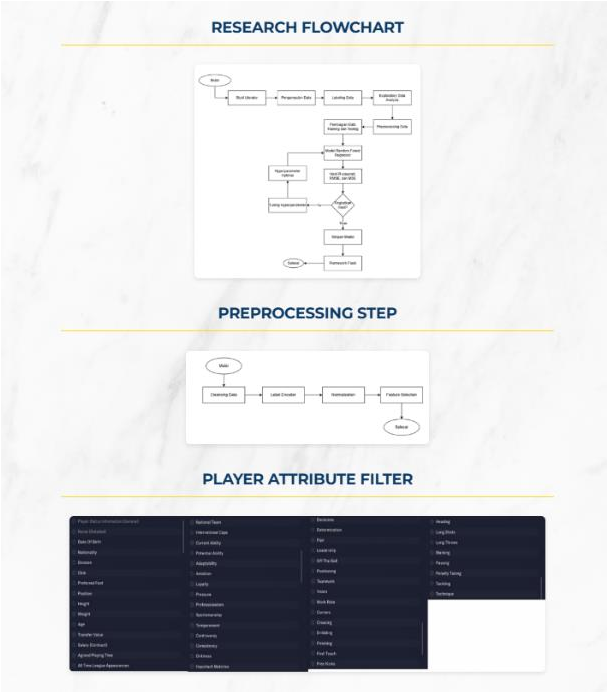
siap pakai, sehingga memudahkan pengembangan situs web. Dalam pengembangan situs web dengan Python, framework seperti Flask dan Django sering menjadi pilihan utama karena keduanya populer dan relatif mudah dipelajari. Dalam penelitian ini, framework Flask dipilih untuk mengimplementasikan model machine learning yang telah dibangun pada tahap sebelumnya menjadi sebuah aplikasi berbasis web.

Situs web dibuat dengan nama PredVal memiliki 3 halaman menu yaitu Home, Prediction, dan Upload. Halaman Home menyajikan informasi terkait pembuatan model yang diimplementasikan dalam aplikasi web. Halaman Home dapat dilihat pada Gambar 20.



Gambar 29. Halaman antarmuka Home (1)

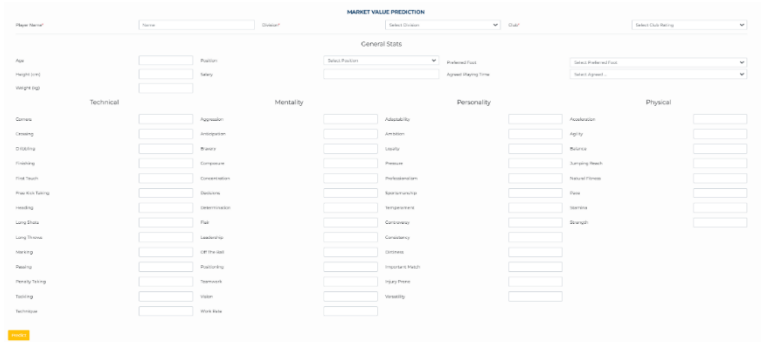
Bagian selanjutnya dari halaman Home menyajikan informasi mengenai *Research Flowchart*, *Preprocessing Step*, dan Atribut pemain yang difilter. Tampilan bagian ketiga pada halaman Home dapat dilihat pada Gambar 60.



Gambar 30. Halaman antarmuka Home (II)

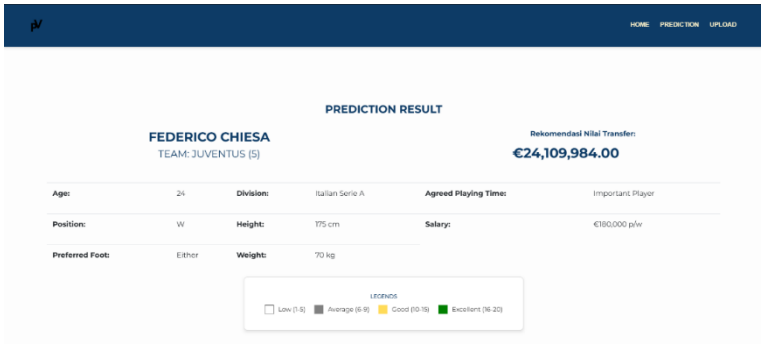
Selanjutnya, halaman yang terdapat pada situs web PredVal adalah halaman Prediction. Di halaman ini, pengguna dapat melakukan input data secara manual. Tampilan antarmuka halaman Prediction dapat dilihat pada Gambar 22.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



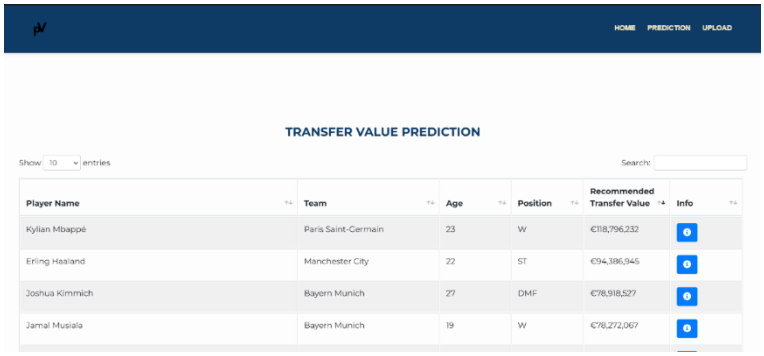
Gambar 31. Halaman antarmuka Prediction

Setelah input dimasukkan melalui halaman Prediction, akan muncul halaman hasil prediksi yang menampilkan kolom 'Rekomendasi Nilai Transfer', berisi rekomendasi nilai transfer berdasarkan model yang telah dikembangkan. Halaman antarmuka hasil prediksi dapat dilihat pada Gambar 23.



Gambar 32. Halaman antarmuka Output Prediksi

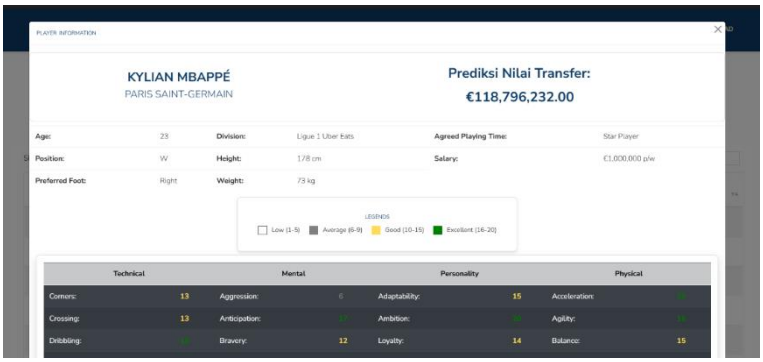
Halaman terakhir pada situs web PredVal adalah halaman di mana pengguna dapat menggunggah sebuah file untuk dilakukan prediksi. Pengguna dapat menggunggah file yang diperoleh dari hasil *scrapping* yang dilakukan sendiri, dengan format file .csv atau .xlsx. Pada setiap data terdapat tombol informasi mengenai data Pemain. Halaman antarmuka hasil prediksi oleh pengguna dapat dilihat pada Gambar 24.



Gambar 33. Halaman antarmuka Hasil Prediksi

Tombol informasi yang terdapat pada setiap data, ketika diklik, akan menampilkan *modals* yang berisi detail informasi mengenai nama pemain, tim pemain, atribut pemain, dan rekomendasi nilai transfer berdasarkan model yang dibuat. Tampilan antarmuka *modals* informasi pemain dapat dilihat pada Gambar 25.

p-ISSN: 2775-4227
e-ISSN: 2775-4219



Gambar 34. Tampilan modals Informasi Pemain.

4. KESIMPULAN

Berdasarkan hasil analisis menggunakan model *Random Forest* untuk memprediksi nilai pasar pemain sepak bola, penelitian ini menunjukkan bahwa fitur *Current Ability* dan *Age* merupakan faktor utama dalam menentukan nilai pasar pemain. Model pertama yang dibangun menggunakan parameter default menghasilkan akurasi R-squared sebesar 77.40%. Untuk meningkatkan performa model, dilakukan *hyperparameter tuning* menggunakan *GridSearchCV*, yang menghasilkan parameter optimal dengan *n_estimators:800, max_depth:10, max_features:1.0, min_samples_split:5, min_samples_leaf:1*, serta *bootstrap:True*. Model yang telah dituning menghasilkan akurasi sebesar 77.81%, menunjukkan peningkatan meskipun tidak terlalu signifikan. Dari analisis *Feature Importance*, ditemukan bahwa *Current Ability* memiliki pengaruh terbesar, diikuti oleh *Age, Salary_scaled, Dribbling Ability, dan Club Rating*, sementara fitur lain seperti *Intelligence, Attacking, dan Possession* memiliki pengaruh yang lebih kecil.

Penelitian ini berkontribusi dalam mengembangkan model prediksi nilai pasar pemain sepak bola berbasis *Random Forest*, yang diintegrasikan dengan halaman web menggunakan *Flask*, sehingga dapat diakses dengan mudah oleh pengguna. Dibandingkan dengan pendekatan tradisional, model ini menawarkan pendekatan *data-driven* yang lebih objektif dalam menilai pemain. Namun, keterbatasan penelitian ini terletak pada kurangnya data terkait performa pemain di dunia nyata, seperti jumlah gol, assist, dan durasi kontrak, yang dapat memengaruhi nilai pasar pemain secara lebih kompleks.

Sebagai rekomendasi untuk penelitian selanjutnya, disarankan untuk (1) mengintegrasikan data performa aktual pemain, seperti gol, assist, dan statistik permainan lainnya, guna meningkatkan relevansi model dengan kondisi nyata. (2) Menggunakan posisi alternatif pemain, bukan hanya posisi terbaiknya, tetapi juga posisi lain yang bisa dimainkan, untuk memahami fleksibilitas pemain dalam pasar transfer. (3) Mengeksplorasi metode *RandomizedSearchCV* dalam *hyperparameter tuning* untuk efisiensi pencarian parameter optimal, serta membandingkan hasilnya dengan *GridSearchCV* guna memperoleh model dengan performa lebih baik. Selain itu, (4) mempertimbangkan metode ensemble lain, seperti *XGBoost* atau *deep learning*, untuk melihat apakah model yang lebih kompleks dapat meningkatkan akurasi prediksi.

DAFTAR PUSTAKA

[1] R. Bahtra, *Buku Ajar Permainan Sepakbola*, no. 156. 2022.

[2] INEOS, "Manchester United plc reaches agreement for Sir Jim Ratcliffe, Chairman of INEOS, to acquire up to a 25% shareholding in the Company," 2023. https://www.ineos.com/news/ineos-group/manchester-united-plc-reaches-agreement-for-sir-jim-ratcliffe-chairman-of-ineos-to-acquire-up-to-a-25-shareholding-in-the-company/

[3] S. Brito dan P. Ferreira, "The Impact of Performance Measures in Football Players ' Transfer Market Value Miguel da Silva Brito Pacheco Ferreira Supervised by Bruno Miguel Pinto Damásio," 2021.

[4] M. E. Kaukab, N. Falah, F. Ekonomi, U. Sains, dan A.- Qur, "FOOTBALL PLAYER MARKET VALUE : APAKAH USIA PEMAIN BERPERAN DALAM PENENTUAN HARGA PASAR ?," vol. 9, no. 1, hal. 24–37, 2021.

[5] A. Metelski, "Original Article Factors affecting the value of football players in the transfer market," vol. 21, no. 2, hal. 1150–1155, 2021, doi: 10.7752/jpes.2021.s2145.

[6] M. A. Al-asadi dan S. Tasdemir, "Predict the Value of Football Players Using FIFA Video Game Data and Machine Learning Techniques," *IEEE Access*, vol. 10, hal. 22631–22645, 2022, doi: 10.1109/ACCESS.2022.3154767.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

[7] G. G. P. K. Laros, "Predicting Transfer Value of Professional Football Players Based on Player Skills and Characteristics Using Multiple Linear Regression, Support Vector Regression, and Random Forest Regression," 2022.

[8] H. Lee, B. A. Tama, dan M. Cha, "Prediction of Football Player Value using Bayesian Ensemble," no. 2015, hal. 1–17, 2022.

[9] E. Fitri, "Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah," *Journal of Applied Computer Science and Technology*, vol. 4, no. 1, hal. 58–64, 2023, doi: 10.52158/jacost.v4i1.491.

[10] B. Setio dan P. Prasetyaningrum, "Penerapan Data Mining Dalam Mengelompokkan Kunjungan Wisatawan Di Kota Yogyakarta Menggunakan Metode K-Means," *Journal of Computer Science and Technology (JCS-TECH)*, vol. 1, no. 1, hal. 27–32, 2021, doi: 10.54840/jcstech.v1i1.9.

[11] H. Y. Pratama, "Powering Up Your Pandas Part II — Label Encoding and One Hot Encoding," *Data Folks Indonesia*, 2022. https://medium.com/data-folks-indonesia/powering-up-your-pandas-part-ii-label-encoding-and-one-hot-encoding-dac0fce045da

[12] A. J. Syahid dan D. Mahdiana, "Perbandingan Algoritma Untuk Klasifikasi Analisis Sentimen Terhadap Genose Pada Media Sosial Twitter," *semanTIK*, vol. 7, no. 1, hal. 9, 2021, doi: 10.55679/semantik.v7i1.18087.

[13] Furkou, "Football Manager Data Toplama ve Temizleme." 2023.