

**ANALISIS PERBANDINGAN ALGORITMA *RANDOM FOREST* DAN  
*SUPPORT VECTOR MACHINE* UNTUK KLASIFIKASI  
HASIL PERTANDINGAN TIM SEPAK BOLA  
LIGA UTAMA INGGRIS MUSIM 2022/2023**

**SKRIPSI**

**CHRISTIAN DUSTIN FRIZZI NAINGGOLAN**

**201401101**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2025**

**ANALISIS PERBANDINGAN ALGORITMA *RANDOM FOREST* DAN  
*SUPPORT VECTOR MACHINE* UNTUK KLASIFIKASI  
HASIL PERTANDINGAN TIM SEPAK BOLA  
LIGA UTAMA INGGRIS MUSIM 2022/2023  
SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer**

**CHRISTIAN DUSTIN FRIZZI NAINGGOLAN  
201401101**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2025**

## PERSETUJUAN

Judul :ANALISIS PERBANDINGAN ALGORITMA  
*RANDOM FOREST* DAN *SUPPORT VECTOR  
MACHINE* UNTUK KLASIFIKASI HASIL  
PERTANDINGAN TIM SEPAK BOLA LIGA  
UTAMA INGGRIS MUSIM 2022/2023

Kategori : SKRIPSI

Nama : CHRISTIAN DUSTIN FRIZZI NAINGGOLAN

Nomor Induk Mahasiswa : 201401101

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :


Pembimbing 2



Desilia Selvida S.Kom., M.Kom.

NIP. 198912052024062002

Pembimbing 1



Dian Rachmawati S.Si., M.Kom.

NIP. 198307232009122004

Diketahui/Disetujui Oleh

Program Studi S-1 Ilmu Komputer



Ketua  
Dr. Amalia ST., M.T.

NIP. 197812212014042001

**PERNYATAAN**

**ANALISIS PERBANDINGAN ALGORITMA *RANDOM FOREST* DAN  
*SUPPORT VECTOR MACHINE* UNTUK KLASIFIKASI  
HASIL PERTANDINGAN TIM SEPAK BOLA  
LIGA UTAMA INGGRIS MUSIM 2022/2023**

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Januari 2025



Christian Dustin Frizzi Nainggolam  
201401101

## PENGHARGAAN

Segala puji syukur dipanjatkan kepada Tuhan Yang Maha Esa atas segala kasih dan karunia-Nya sehingga penulis dapat berada di tahap penyusunan skripsi ini sebagai syarat untuk mendapatkan gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer, Universitas Sumatera Utara.

Dengan penuh rasa hormat pada kesempatan ini penulis mengucapkan terima kasih kepada Ibu tercinta, Mayuni R S Manurung atas segala bentuk perjuangan, kasih sayang, dan perlindungan dengan doa-doa yang dipanjatkan untuk penulis. Terima kasih juga untuk setiap dukungan yang telah diberikan hingga penulis dapat berada di titik ini.

Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, dan bimbingan dari banyak pihak. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan sebagai Dosen Pembimbing I yang telah memberikan banyak masukan, motivasi, serta dukungan kepada penulis selama penyusunan skripsi ini.
5. Ibu Dian Rachmawati S.Si., M.Kom. sebagai Dosen Pembimbing I yang telah memberikan banyak masukan, motivasi, serta dukungan kepada penulis selama penyusunan skripsi ini.
6. Ibu Desilia Selvida S.Kom., M.Kom. selaku dosen Pembimbing II yang telah memberikan banyak bimbingan dan masukan yang berharga kepada penulis selama proses penyusunan skripsi ini.
7. Ibu Sri Melvani Hardi S.Kom., M.Kom selaku Dosen Pembimbing Akademik yang telah memberikan saran, motivasi dan banyak dukungan kepada penulis selama perkuliahan.

Dan seluruh pihak yang telah memberi dukungan serta doa baik yang tidak dapat penulis sebutkan satu per-satu. Semoga Tuhan Yang Maha Esa senantiasa melimpahkan keberkahan serta kebaikan atas semua dukungan yang telah diberikan kepada penulis dan hasil penelitian ini dapat memberi manfaat maupun inspirasi untuk kedepannya.

Medan, Januari 2025

Penulis,



Christian Dustin Frizzi Nainggolan

## ABSTRAK

Keberlangsungan suatu pertandingan dalam olahraga sepak bola tidak hanya dilihat dari operan dan tendangan, melainkan juga turut memperhatikan komposisi dan strategi yang harus dimiliki setiap tim agar dapat memaksimalkan setiap pertandingan untuk memperoleh kemenangan. Oleh karena itu, Klasifikasi diperlukan untuk mengidentifikasi faktor yang berpengaruh pada klub untuk meraih kemenangan dalam pertandingan sepak bola. Maka dari itu, pendekatan yang dapat dilakukan adalah dengan menggunakan algoritma yang ada pada Machine learning yaitu Algoritma Random Forest dan Algoritma Support Vector Machine yang dapat mengidentifikasi dan mengklasifikasikan hasil pertandingan sepak bola serta memberikan informasi tambahan bagi klub dalam merancang strategi permainan dalam menghadapi klub lain sehingga efektif dalam setiap pertandingan. Hasil pengujian sistem menunjukkan bahwa Algoritma Random Forest didapatkan nilai total akurasi sebesar 94%. Sedangkan pada Algoritma Support Vector Machine dengan menggunakan beberapa kernel yaitu kernel Linear, Radial, dan Polynomial didapatkan nilai total akurasi masing-masing sebesar 100%, 88%, 97%. Ini menunjukkan bahwa klasifikasi menggunakan Support Vector Machine (SVM) sangat baik.

**Kata Kunci:** Liga Utama Inggris, Klasifikasi, *Machine Learning*, *Random Forest*, *Support Vector Machine*.

## ABSTRACT

*Football is not only about the passes and shots that must be taken into account in the course of a game, but also about the composition and strategy that each team must have in order to maximize each game and win. Therefore, a classification is needed to identify the factors that influence the team's victory in a football match. One approach that can be done is to use existing algorithms in machine learning, namely the Random Forest Algorithm and the Support Vector Machine Algorithm, which can identify and classify the results of football matches and provide additional information for clubs to design game strategies in facing other clubs so that they are effective in each match. The results of the system tests show that the Random Forest algorithm achieved an overall accuracy value of 94%. Meanwhile, the Support Vector Machine algorithm using multiple kernels, namely the linear, radial, and polynomial kernels, obtained a total accuracy value of 100%, 88%, and 97%, respectively. This shows that the Support Vector Machine (SVM) classification is excellent.*

**Keywords:** English Premier League, Classification, Machine Learning, Random Forest, Support Vector Machine.



## DAFTAR ISI

<b>PERSETUJUAN</b> .....	ii
<b>PERNYATAAN</b> .....	iii
<b>PENGHARGAAN</b> .....	iv
<b>ABSTRAK</b> .....	vi
<b>ABSTRACT</b> .....	vii
<b>DAFTAR ISI</b> .....	viii
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>BAB 1 PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	3
<b>1.3 Batasan Masalah</b> .....	4
<b>1.4 Tujuan Penelitian</b> .....	4
<b>1.5 Manfaat Penelitian</b> .....	5
<b>1.6 Metodologi Penelitian</b> .....	5
<b>1.7 Penelitian Relevan</b> .....	6
<b>1.8 Sistematika Penulisan</b> .....	7
<b>BAB 2 LANDASAN TEORI</b> .....	9
<b>2.1 Sepak Bola</b> .....	9
<b>2.2 Liga Utama Inggris</b> .....	9
<b>2.3 Data Mining</b> .....	10
<b>2.4 Machine Learning</b> .....	12
2.4.1 <i>Supervised Learning</i> .....	13
2.4.2 <i>Unsupervised Learning</i> .....	13
2.4.3 <i>Reinforcement Learning</i> .....	14
<b>2.5 Klasifikasi</b> .....	14
<b>2.6 Algoritma Random Forest</b> .....	14
<b>2.7 Algoritma Support Vector Machine (SVM)</b> .....	16
<b>BAB 3 ANALISIS DAN PERANCANGAN</b> .....	21
<b>3.1 Analisis Sistem</b> .....	21
3.1.1 Analisis Masalah .....	21

3.1.2	Analisis Data .....	21
3.1.3	Analisis kebutuhan .....	21
<b>3.2</b>	<b>Arsitektur Penelitian .....</b>	<b>23</b>
<b>3.3</b>	<b>Perancangan Sistem .....</b>	<b>24</b>
3.3.1	<i>Use Case Diagram</i> .....	24
3.3.2	<i>Activity Diagram</i> .....	26
3.3.3	<i>Sequence Diagram</i> .....	27
<b>3.4</b>	<b>Ishikawa Diagram .....</b>	<b>28</b>
<b>3.5</b>	<b>Flowchart .....</b>	<b>30</b>
<b>BAB 4</b>	<b>IMPLEMENTASI DAN PENGUJIAN .....</b>	<b>34</b>
4.1	Implementasi Sistem .....	34
4.2	Analisis Deskriptif .....	34
4.3	Klasifikasi Menggunakan Data Asli .....	36
4.4	Perbandingan Hasil Setiap Metode Klasifikasi.....	55
<b>BAB 5</b>	<b>PENUTUP.....</b>	<b>57</b>
5.1	Kesimpulan.....	57
5.2	Saran .....	57
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>58</b>

**DAFTAR TABEL**

<b>Tabel 4. 1</b> Klasemen Akhir Liga Utama Inggris 2022-2023 .....	35
<b>Tabel 4. 2</b> Hasil Akurasi, Precision, Recall, dan F1-Score.....	47
<b>Tabel 4. 3</b> Hasil Akurasi, Precision, Recall, dan F1-Score.....	52
<b>Tabel 4. 4</b> Hasil Akurasi, Precision, Recall, dan F1-Score.....	53
<b>Tabel 4. 5</b> Hasil Akurasi, Precision, Recall, dan F1-Score.....	55
<b>Tabel 4. 6</b> Perbandingan Akurasi Model Klasifikasi .....	56

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Pertandingan Sepak Bola .....	9
<b>Gambar 2. 2</b> Klub Liga Utama Inggris .....	10
<b>Gambar 2. 3</b> Arsitektur Data Mining (Algarni, A., 2016) .....	12
<b>Gambar 2. 4</b> Proses Algoritma Model ML .....	12
<b>Gambar 2. 5</b> Struktur Model Hyperplane Algoritma SVM (Huang, 2018).....	17
<b>Gambar 3. 1</b> Arsitektur Umum Sistem .....	24
<b>Gambar 3. 2</b> Use Case Diagram .....	25
<b>Gambar 3. 3</b> Activity Diagram .....	27
<b>Gambar 3. 4</b> Sequence Diagram .....	28
<b>Gambar 3. 5</b> Ishikawa Diagram .....	29
<b>Gambar 3. 6</b> Flowchart Sistem Random Forest.....	31
<b>Gambar 3. 7</b> Flowchart Sistem Support Vector Machine.....	32
<b>Gambar 4. 1</b> Perbandingan Persentase Kemenangan.....	36
<b>Gambar 4. 2</b> Tipe Data Variabel Penelitian .....	37
<b>Gambar 4. 3</b> Data Training .....	37
<b>Gambar 4. 4</b> Data Testing .....	38
<b>Gambar 4. 5</b> Fitur dan Target .....	38
<b>Gambar 4. 6</b> Data Latih dan Data Uji .....	38
<b>Gambar 4. 7</b> Model Random Forest .....	39
<b>Gambar 4. 8</b> Proses Menghitung Metrik Evaluasi .....	40
<b>Gambar 4. 9</b> Proses Evaluasi Model .....	40
<b>Gambar 4. 10</b> Proses Membuat Grafik Metrik Evaluasi.....	42
<b>Gambar 4. 11</b> Hasil Grafik Akurasi, F1-Score, Precision, dan Recall .....	42
<b>Gambar 4. 12</b> Hasil Confusion Matrix .....	43
<b>Gambar 4. 13</b> Fitur dan Target .....	48
<b>Gambar 4. 14</b> Data Latih dan Data Uji .....	48
<b>Gambar 4. 15</b> Normalisasi Fitur .....	49
<b>Gambar 4. 16</b> Model SVM .....	49
<b>Gambar 4. 17</b> Membuat model SVM .....	51
<b>Gambar 4. 18</b> Melatih model SVM .....	51
<b>Gambar 4. 19</b> Model SVC .....	51
<b>Gambar 4. 20</b> Membuat model RBF.....	52

<b>Gambar 4. 21</b> Melatih Model SVM.....	53
<b>Gambar 4. 22</b> Model Kernel RBF.....	53
<b>Gambar 4. 23</b> Membuat Model Poly .....	54
<b>Gambar 4. 24</b> Melatih Model Poly .....	54
<b>Gambar 4. 25</b> Model Kernel Poly.....	54

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Sepak bola merupakan salah satu cabang olahraga yang dapat dimainkan oleh seluruh kalangan dan juga merupakan olahraga yang paling banyak digemari dan diminati di seluruh negara. Hal yang mempengaruhi sepak bola begitu diminati di kalangan masyarakat terletak pada cara memainkannya dimana olahraga sepak bola mudah untuk dipahami dan tidak terlalu sulit untuk dilakukan.

Dalam olahraga sepak bola, keberhasilan suatu pertandingan tidak hanya bergantung pada operan dan tendangan, tetapi juga pada komposisi pemain serta strategi yang diterapkan oleh setiap tim. Strategi yang matang dan pemilihan komposisi yang tepat menjadi faktor penting dalam memaksimalkan peluang untuk meraih kemenangan. Oleh sebab itu, proses klasifikasi menjadi krusial untuk mengidentifikasi berbagai faktor yang memengaruhi hasil akhir pertandingan. Faktor-faktor ini dapat mencakup kemampuan individu pemain, kerja sama tim, hingga pola permainan yang digunakan. Salah satu pendekatan efektif untuk menganalisis dan mengklasifikasikan dengan memanfaatkan algoritma dalam machine learning. Dengan bantuan machine learning, tim dapat mengolah data pertandingan secara lebih mendalam, memahami pola-pola tersembunyi, dan memberikan rekomendasi strategis yang berbasis data untuk meningkatkan performa di lapangan.

Machine Learning (ML) ialah teknologi mesin yang memungkinkan komputer menganalisis dan memproses data secara cerdas, menyerupai cara manusia belajar. Teknologi ini dirancang untuk terus meningkatkan akurasi dan kemampuan melalui proses pembelajaran dari data yang tersedia (Jafar Alzubi et al., 2018). Model pembelajaran dalam ML dapat dikembangkan untuk berbagai tujuan, seperti melakukan klasifikasi data, membuat prediksi, serta menggali pola dan hubungan tersembunyi di dalam data (Guo & Hao, 2021).

Selain itu, penerapan ML telah meluas ke berbagai bidang, termasuk kesehatan, keuangan, dan olahraga, di mana kemampuan untuk menganalisis data secara efisien memainkan peran penting dalam pengambilan keputusan

berbasis informasi. Hal ini menjadikan ML sebagai salah satu pilar utama dalam perkembangan teknologi modern.

Dalam penelitian ini, algoritma yang diterapkan adalah Random Forest dan Support Vector Machine (SVM). Kedua algoritma ini dikenal luas dan sering digunakan dalam berbagai kasus klasifikasi karena keandalannya dalam menangani data yang kompleks (Azhari et al., 2021). Random Forest merupakan algoritma berbasis ensemble yang menggabungkan hasil dari banyak pohon keputusan untuk meningkatkan akurasi, sementara Support Vector Machine bekerja dengan mencari hyperplane optimal yang memisahkan data ke dalam kategori tertentu. Meskipun sama-sama digunakan untuk klasifikasi, kinerja kedua algoritma ini dapat menghasilkan tingkat akurasi yang berbeda tergantung pada karakteristik data yang dianalisis. Penelitian ini akan mengevaluasi kelebihan dan kelemahan masing-masing algoritma dalam konteks klasifikasi hasil pertandingan sepak bola, untuk menentukan algoritma yang lebih sesuai dengan data yang digunakan.

Random Forest adalah algoritma pembelajaran mesin yang sangat fleksibel dan mudah diterapkan, menjadikannya salah satu pilihan populer untuk berbagai tugas klasifikasi dan regresi. Algoritma ini dikenal karena kesederhanaannya dalam implementasi serta kemampuannya untuk menangani data yang kompleks dan beragam. Salah satu keunggulan utama Random Forest adalah kemampuannya meningkatkan akurasi meskipun terdapat data yang hilang atau pencilan. Selain itu, algoritma ini efisien dalam penyimpanan data dan mampu melakukan seleksi fitur untuk meningkatkan performa model klasifikasi secara keseluruhan (Wijaya et al., 2023). Dalam prosesnya, Random Forest menggunakan pendekatan ensemble dengan metode bagging, yaitu penggabungan prediksi pohon keputusan untuk mengurangi overfitting dan meningkatkan ketepatan (Azhari et al., 2021). Keunggulan ini menjadikan Random Forest sebagai salah satu algoritma andalan dalam analisis data yang membutuhkan prediksi akurat, terutama pada dataset yang besar dan kompleks.

Support Vector Machine (SVM) adalah algoritma machine learning yang berjenis supervised yang dapat digunakan untuk tugas klasifikasi maupun regresi, meskipun penerapannya lebih sering ditemui dalam konteks klasifikasi. Salah satu keunggulan utama SVM adalah kemampuannya dalam menemukan

hyperplane optimal yang memaksimalkan margin antara kelas data, sehingga menghasilkan pemisahan yang jelas dan akurat. Algoritma ini sangat efektif untuk data dengan dimensi tinggi, karena tetap dapat bekerja baik meskipun terdapat banyak fitur. Namun, SVM juga memiliki beberapa kelemahan, terutama terkait dengan pemilihan parameter, seperti kernel, C, dan gamma, yang sangat memengaruhi kinerja model. Selain itu, proses seleksi fitur yang tepat menjadi penting untuk memastikan algoritma dapat menangkap pola-pola signifikan dalam data. Oleh karena itu, kombinasi antara pemahaman mendalam tentang data dan pengaturan parameter yang optimal menjadi kunci untuk meningkatkan akurasi hasil klasifikasi yang dihasilkan oleh SVM (Azhari et al., 2021). SVM juga dapat diperluas untuk menangani masalah non-linear dengan menggunakan kernel tertentu, seperti radial basis function (RBF) atau polynomial, yang memungkinkan algoritma untuk memetakan data ke ruang dimensi yang lebih tinggi.

Berdasarkan uraian di awal maka penelitian ini akan menghasilkan nilai akurasi, presisi berupa label atau angka yang menunjukkan kelas suatu data ditempatkan. Klasifikasi dalam penelitian ini berdasarkan beberapa variabel yang mencakup shots, shots on target, corner kick, fouls, offsides, yellow cards, dan red cars. Variabel tersebut dipilih berdasarkan referensi dari penelitian Putra (2017) dan Laia, dkk (2019).

Oleh karena itu, penelitian ini dapat memberikan informasi tambahan kepada klub tentang cara membuat strategi permainan yang efektif untuk menghadapi permainan lawan agar dapat efektif dalam bermain dan memenangkan setiap pertandingan yang ada.

## **1.2 Rumusan Masalah**

Hasil akhir dari sebuah pertandingan sepak bola tidak hanya ditentukan oleh komposisi pemain dalam suatu tim, tetapi juga dipengaruhi oleh berbagai aspek penting yang berkontribusi terhadap keberhasilan klub dalam meraih kemenangan. Faktor-faktor seperti strategi permainan, performa individu pemain, dan dinamika tim menjadi elemen yang signifikan dalam menentukan hasil pertandingan. Untuk memahami hal ini lebih dalam, diperlukan proses



klasifikasi guna memprediksi hasil pertandingan sepak bola dengan menggunakan algoritma seperti Random Forest dan Support Vector Machine (SVM). Metode ini tidak hanya membantu dalam menentukan teknik klasifikasi yang paling efektif berdasarkan data yang tersedia, tetapi juga mengidentifikasi variabel-variabel utama yang memiliki pengaruh terbesar terhadap peluang kemenangan sebuah tim. Dengan demikian, analisis ini dapat memberikan wawasan strategis yang berguna bagi tim dalam mempersiapkan pertandingan.

### 1.3 Batasan Masalah

Pada penelitian, peneliti menetapkan sejumlah pembatasan terhadap lingkup penyelesaian masalah. Berikut adalah batasan-batasan dalam konteks penelitian ini:

1. Data yang digunakan yaitu data hasil pertandingan sepak bola Liga Utama Inggris musim 2022/2023.
2. Parameter yang digunakan ialah klub yang bertanding, jumlah gol, total shots, sepak pojok, offside, kartu kuning, kartu merah dan hasil akhir pertandingan menggunakan perbandingan Algoritma *Random Forest* dan *Support Vector Machine* (SVM) untuk menentukan nilai metrik evaluasi berupa akurasi, *precision*, *recall*, *f1-score*.
3. Dataset yang digunakan dalam penelitian ini berupa data *sekunder* yang diambil dari laman resmi *premier league* musim 2022/2023.
4. Sistem yang dibangun akan menampilkan output keluaran berupa akurasi, presisi, *recall*, dan *f1-score* terhadap data hasil pertandingan dan menggunakan bahasa pemrograman *Python* dan IDE berupa *Visual Studio Code*.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menganalisis hasil pertandingan Liga Utama Inggris guna memperoleh nilai akurasi dan presisi yang dapat mengungkap faktor-faktor kunci yang memengaruhi kemenangan suatu tim dalam pertandingan sepak bola. Penelitian ini akan menggunakan pendekatan machine learning untuk menggali informasi lebih dalam mengenai elemen-elemen yang berperan penting dalam hasil pertandingan.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu sebagai berikut

1. Dapat meningkatkan pemahaman mengenai cara mengklasifikasikan berbagai variabel yang terdapat dalam olahraga sepak bola.
2. Penelitian ini akan memberikan wawasan mengenai seberapa besar pengaruh masing-masing variabel terhadap hasil pertandingan sepak bola Liga Utama Inggris, sehingga memungkinkan analisis yang lebih mendalam mengenai faktor-faktor penentu kemenangan.
3. Penelitian ini diharapkan dapat memberikan informasi yang berguna bagi klub-klub sepak bola dalam merancang strategi permainan yang lebih efektif dan dapat membantu tim dalam mengambil keputusan yang lebih berbasis data untuk meningkatkan peluang kemenangan dalam setiap pertandingan.

## 1.6 Metodologi Penelitian

Beberapa metode yang diterapkan dalam penelitian ini mencakup diantaranya sebagai berikut:

### 1. Studi Pustaka

Studi pustaka dilakukan dengan tujuan mengumpulkan berbagai materi referensi yang terkait dengan klasifikasi, *data mining*, pengolahan data, serta *Random Forest* dan *Support Vector Machine* (SVM) dari sejumlah jurnal, artikel, buku, dan sumber referensi lainnya.

### 2. Analisis dan Perancangan Sistem

Pada tahap analisis dilakukan analisis kebutuhan sesuai dengan tujuan penelitian yang melibatkan pemahaman yang mendalam tentang proses Klasifikasi dengan memanfaatkan model *Random Forest* dan *Support Vector Machine* dengan Langkah selanjutnya melakukan perancangan sistem berupa Arsitektur Umum Sistem, Perancangan Sistem dan diagram alir (*flowchart*), dan diagram ishikawa.

### 3. Implementasi Sistem

Dalam proses implementasi, Diagram alir yang direncanakan akan berfungsi sebagai dasar untuk membuat aplikasi. Bahasa pemrograman *Python* dan

*Visual Studio Code* (IDE) akan digunakan dalam penelitian ini untuk membuat sistem.

#### 4. Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap sistem yang telah dirancang sesuai dengan kebutuhan dan spesifikasi yang diinginkan untuk memastikan program telah dapat berjalan sesuai dengan rancangan. Dan apabila jika terdapat kesalahan, melakukan perbaikan sejauh yang diperlukan.

#### 5. Dokumentasi Sistem

Dilakukan proses dokumentasi seluruh tahap dari analisis sampai ke tahap pengujian sistem dalam bentuk skripsi untuk menunjukkan hasil dari penelitian yang telah dilakukan.

### 1.7 Penelitian Relevan

Beberapa penelitian yang relevan dengan penelitian ini yaitu:

1. Penelitian dengan judul “*Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL)*” (Razali *et al.*, 2017). Penelitian ini menggunakan *Bayesian Network* untuk melakukan prediksi hasil pertandingan sepak bola. Uji prediksi dilakukan dengan menggunakan software WEKA. Prediksi dilakukan berdasarkan data dari tiga musim yang berbeda dan mendapatkan hasil akurasi rata-rata sebesar 75,09% selama tiga musim menggunakan Bayesian Network dalam kumpulan data ini jauh di atas persentase akurasi prediksi keseluruhan rata-rata sebesar 59,21%.
2. Penelitian dengan judul “*Fuzzy-Based Model for Predicting Football Match Results*” (Omomule *et al.*, 2020) mengimplementasikan algoritma logika fuzzy untuk melakukan model prediksi untuk hasil pertandingan sepak bola. Data yang digunakan yaitu Liga Inggris musim 2017/2018. Mendapatkan hasil rata-rata kesalahan pengujian sebesar 0,075 dengan akurasi prediksi yang ditingkatkan sebesar 89,27%.
3. Algoritma C.45 digunakan dalam penelitian berjudul “PENERAPAN DATA MINING DALAM MEMPREDIKSI PEMENANG KLUB SEPAK BOLA PADA AJANG LIGA CHAMPION DENGAN ALGORITMA C.45” (Laia *et al.*, 2019). Dalam penelitian ini, tim yang akan memenangkan liga champion

adalah variabel dependennya, sedangkan kualitas pemain, umur, kualitas pelatih, kekayaan, dan prestasi klub adalah variabel independen. Hasil dari pengolahan data mining dengan algoritma C.45 menunjukkan bahwa kondisi manajemen keuangan klub dan umur rata-rata pemain sepak bola adalah kriteria kemenangan klub dalam kompetisi Liga Champion.

## **1.8 Sistematika Penulisan**

Sistematika penulisan skripsi yang digunakan dalam penelitian ini adalah sebagai berikut:

### **BAB 1 PENDAHULUAN**

Pada bab ini mencakup hal latar belakang pemilihan judul, rumusan dan batasan masalah, tujuan, keuntungan, dan metodologi penelitian, serta metode penelitian relevan.

### **BAB 2 LANDASAN TEORI**

Bab ini menjelaskan beberapa teori yang berkaitan dengan penelitian, seperti pendahuluan *Machine Learning*, *Random Forest*, *Support Vector Machine* (SVM), *Data Mining*, sepak bola.

### **BAB 3 ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan bagaimana analisis pada algoritma dan perancangan diagram yang diperlukan, seperti diagram alir (*flowchart*), ishikawa diagram.

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi penjelasan mengenai implementasi algoritma pada sistem model identifikasi klasifikasi pertandingan sepak bola beserta hasil pengujian sistem yang dibangun dan analisisnya.

### **BAB 5 KESIMPULAN DAN SARAN**

Dalam bab ini memberikan informasi kesimpulan yang dapat diperoleh berdasarkan hasil penelitian beserta saran yang diberikan peneliti sebagai masukan untuk penelitian selanjutnya.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Sepak Bola**

Sepak bola merupakan salah satu cabang olahraga yang dapat dimainkan oleh seluruh kalangan dan juga merupakan olahraga yang paling banyak digemari dan diminati di seluruh negara. Hal yang mempengaruhi sepak bola begitu diminati di kalangan masyarakat terletak pada cara memainkannya dimana olahraga sepak bola mudah untuk dipahami dan tidak terlalu sulit untuk dilakukan. Untuk lebih memperjelas bagaimana sepak bola berlangsung, dapat dilihat pada gambar 2.1.



**Gambar 2. 1** Pertandingan Sepak Bola

#### **2.2 Liga Utama Inggris**

Liga Utama Inggris, yang juga dikenal sebagai Premier League, merupakan liga sepak bola tertinggi di Inggris. Liga ini terdiri dari 20 klub yang bersaing setiap musimnya dengan sistem promosi dan degradasi yang serupa dengan yang diterapkan pada English Football League (EFL).

Sistem promosi dan degradasi juga berlaku di antara Liga Utama

dan Kejuaraan EFL, di mana tiga tim terbawah Liga Utama akan turun ke Kejuaraan EFL, sementara dua tim teratas dari Kejuaraan EFL akan dipromosikan ke Liga Utama. Tim ketiga di Kejuaraan EFL akan mendapatkan kesempatan promosi melalui babak play-off yang melibatkan tim peringkat ketiga hingga keenam. Pada awalnya, Liga Utama Inggris terdiri dari 22 tim pada tahun 1992, namun jumlah ini dikurangi menjadi 20 tim pada tahun 1995. Pada gambar 2.2 tertera klub yang bermain di Liga Utama Inggris.



**Gambar 2. 2** Klub Liga Utama Inggris

### 2.3 Data Mining

Data mining adalah proses menganalisis dan mengolah sejumlah besar data dengan tujuan menemukan informasi penting dari berbagai sudut pandang atau dimensi, sehingga dapat digunakan untuk membuat keputusan (Algarni, A., 2016). Sistem data mining memiliki banyak bagian, seperti:

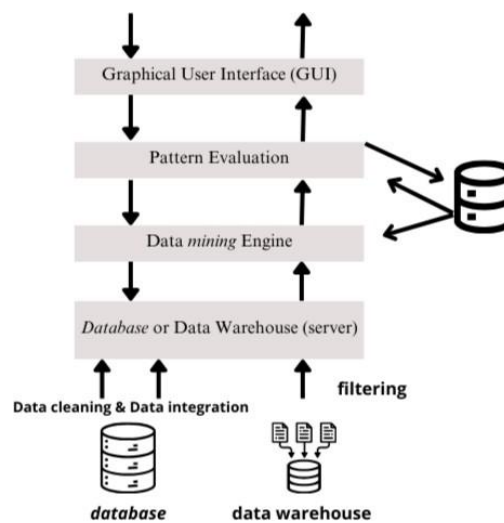
1. Database atau gudang data (data warehouse) merupakan sistem penyimpanan informasi yang berperan dalam mengelola dan mencari data yang relevan sesuai dengan kebutuhan pengguna atau user.
2. Basis pengetahuan (knowledge base) adalah sebuah sistem penyimpanan data yang terstruktur, yang berfungsi untuk mengumpulkan, mengelola, dan membagikan pengetahuan serta informasi yang relevan kepada pengguna. Sistem ini tidak hanya menyimpan data, tetapi juga memfasilitasi pengelolaan informasi yang dapat digunakan untuk mendukung pengambilan keputusan atau menyelesaikan masalah.
3. Sistem ini beroperasi dengan menjalankan program yang didasarkan pada algoritma yang telah dirancang sebelumnya, memungkinkan pengolahan

dan analisis data yang efisien untuk mengungkap pola atau informasi tersembunyi. Melalui proses ini, data mining dapat membantu dalam pengambilan keputusan dan memberikan wawasan yang lebih mendalam mengenai data yang ada.

4. Evaluasi pola, yang juga sering disebut sebagai pattern evaluation, merupakan komponen penting yang berinteraksi langsung dengan modul-modul dalam sistem data mining. Fungsi utama dari komponen ini adalah untuk mengidentifikasi pola-pola yang ada dalam database yang sedang dianalisis. Setelah pola-pola tersebut ditemukan, proses data mining akan melanjutkan untuk mengumpulkan informasi relevan yang dapat digunakan untuk tujuan lebih lanjut, seperti pengambilan keputusan atau prediksi. Evaluasi pola memainkan peran krusial dalam memastikan bahwa data yang ditemukan benar-benar memiliki nilai yang signifikan dan dapat memberikan wawasan yang berguna bagi pengguna atau sistem yang mengolahnya. Dengan demikian, proses ini tidak hanya mengidentifikasi pola, tetapi juga menilai relevansi dan kualitas data yang diekstraksi.
5. Antarmuka Pengguna Grafis (Graphical User Interface atau GUI) adalah komponen utama yang memungkinkan pengguna berkomunikasi dengan sistem. Modul ini dirancang untuk mempermudah interaksi antara pengguna dan sistem, serta memberikan kebebasan bagi pengguna untuk menentukan sendiri proses pengolahan data mining. Melalui GUI, pengguna dapat dengan mudah mengakses fitur-fitur yang tersedia dan melakukan pemrosesan data secara visual. Antarmuka ini menyediakan tampilan yang intuitif, sehingga mempermudah pemahaman dan penggunaan berbagai fungsi dalam sistem. Dengan demikian, GUI berperan penting dalam meningkatkan efisiensi dan efektivitas dalam pemanfaatan sistem.

Arsitektur *Data Mining* dapat dilihat dan ditunjukkan pada gambar 2.3 berikut



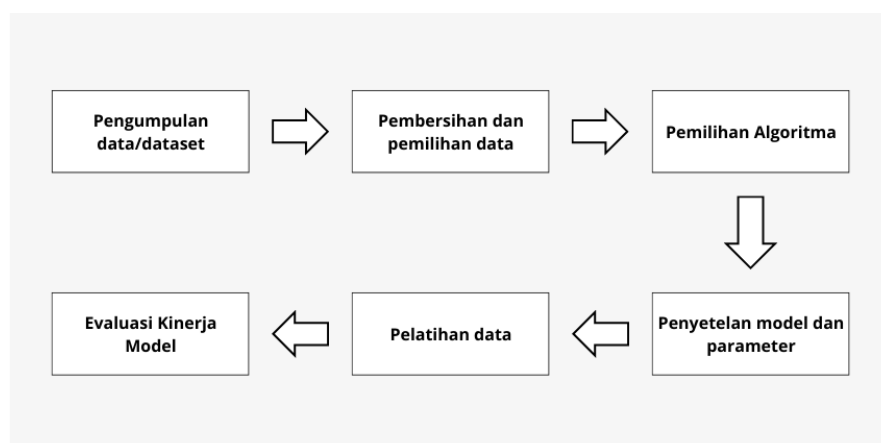


**Gambar 2. 3** Arsitektur Data Mining (Algarni, A., 2016)

## 2.4 Machine Learning

*Machine Learning* (ML) adalah teknologi mesin yang memungkinkan komputer dalam menganalisis data dan memproses data lainnya seperti halnya manusia dalam melakukan pembelajaran yang dapat meningkatkan akurasi dan kapabilitas melalui algoritma itu sendiri (Jafar Alzubi et al., 2018). Model Pembelajaran ML dapat dilatih untuk melakukan klasifikasi, prediksi, dan penambahan interaksi pada data (Guo & Hao, 2021).

Ada beberapa jenis metode dalam machine learning yaitu: supervised learning, unsupervised learning, dan reinforcement learning. Adapun proses-proses yang ada pada algoritma model *machine learning* dapat dilihat pada gambar 2.4.



**Gambar 2. 4** Proses Algoritma Model ML

1. **Pengumpulan data/dataset:** Tugas utama dalam proses pembelajaran mesin adalah untuk mengumpulkan dan menyiapkan data dalam format yang dapat diberikan sebagai masukan ke algoritma. Oleh karena itu, data perlu dikumpulkan dan diproses terlebih dahulu ke format yang terstruktur
2. **Pemilihan dan Pembersihan Data:** Berdasarkan dataset yang sudah diperoleh, mungkin tidak semuanya relevan dengan proses pembelajaran yang akan dirancang sehingga perlu dilakukan pemilihan dan pembersihan *dataset* yang akan digunakan
3. **Pemilihan Algoritma:** Setiap algoritma pembelajaran mesin memiliki kecocokan yang berbeda untuk berbagai jenis masalah. Beberapa algoritma lebih sesuai untuk kategori masalah tertentu, seperti yang dijelaskan sebelumnya.
4. **Penyetelan Model dan Parameter:** Untuk pemilihan atau penyetelan model dan parameter sebagian besar algoritma pembelajaran mesin memerlukan beberapa intervensi manual awal untuk menetapkan nilai yang paling tepat dari berbagai parameter
5. **Pelatihan Data:** Setelah memilih algoritma yang sesuai dengan topik analisis dan nilai parameter yang sesuai, model perlu dilatih dengan menggunakan sebagian dataset sebagai data pelatihan
6. **Evaluasi Kinerja Model:** Sebelum implementasi sistem secara *real-time*, model harus diuji terhadap data yang tidak terlihat untuk mengevaluasi seberapa banyak yang telah dipelajari dengan *metric* evaluasi. Pendekatan pada *machine learning* sebagai berikut

#### 2.4.1 Supervised Learning

Untuk belajar dan membuat prediksi atau klasifikasi pada data baru, *supervised learning* menggunakan dataset yang dilabeli sebagai model algoritma (Jafar Alzubi *et al.*, 2018)

#### 2.4.2 Unsupervised Learning

*Unsupervised Learning* adalah model *machine learning* yang menggunakan data tanpa label untuk mengidentifikasi pola atau struktur data.

### 2.4.3 Reinforcement Learning

*Reinforcement Learning* adalah model pembelajaran yang berada diantara keduanya (*supervised* dan *unsupervised*). Tipe algoritma ini hanya memberikan respon terhadap output yang dihasilkan apakah sudah benar atau tidak. Algoritma ini juga harus mencari dan mengeluarkan banyak kemungkinan untuk mendapatkan hasil yang benar.

## 2.5 Klasifikasi

Klasifikasi merupakan proses untuk mengelompokkan objek atau entitas ke dalam kategori atau kelas tertentu berdasarkan data yang telah ditentukan sebelumnya. Dalam konteks pengembangan model machine learning, klasifikasi digunakan untuk memprediksi kelas atau kategori dari data yang belum diketahui dengan memanfaatkan data yang telah dilabeli sebelumnya.

Teknik ini sangat berguna dalam berbagai aplikasi, seperti deteksi spam, pengenalan wajah, atau diagnosis medis, di mana tujuan utamanya adalah untuk memberikan label yang sesuai berdasarkan pola yang ditemukan dalam data pelatihan. Proses klasifikasi melibatkan berbagai algoritma yang dapat memilih fitur-fitur yang paling relevan untuk menghasilkan prediksi yang akurat. Dengan kemajuan teknologi dan teknik-teknik baru dalam machine learning, klasifikasi semakin efektif dalam menangani dataset besar dan kompleks.

## 2.6 Algoritma *Random Forest*

*Random forest* adalah algoritma pembelajaran mesin populer yang dapat digunakan untuk mengembangkan model prediksi. Pertama kali diperkenalkan oleh Breiman pada tahun 2001 (Breiman, 2001), *Random Forest* adalah kumpulan pohon klasifikasi dan regresi (Breiman, Friedman, Olshen, & Stone, 1984). Dalam penerapannya, Algoritma *Random Forest* menggunakan metode *ensemble learning* yaitu *Boosting* dan *Bootstrap Aggregating (bagging)*.

*Boosting* adalah proses membangun sekumpulan model, dimana setiap model mencoba untuk memperbaiki kesalahan model sebelumnya dalam urutan tertentu. Sedangkan *Bootstrap Aggregating* yang dikenal juga sebagai bagging merupakan metode *ensemble learning* yang didesain untuk meningkatkan stabilitas dan akurasi model terintegrasi dari suatu data. Dengan demikian,

*bagging* diakui lebih kuat terhadap masalah *overfitting* dibandingkan dengan pendekatan *boosting* (Sheykhmousa et al., 2020)

Manfaat dari penggunaan *random forest* untuk model prediksi adalah kemampuan untuk mengelola dataset dengan banyak variabel prediktor; namun, seringkali dalam praktik, jumlah variabel prediktor yang diperlukan untuk mendapatkan prediksi hasil harus dikurangi untuk meningkatkan efisiensi terhadap hasil data yang didapat (Speiser et al., 2019).

Untuk menjalankan model *Random Forest*, ada dua parameter yang harus ditetapkan: Jumlah pohon (*Ntree*) dan jumlah fitur yang dipilih secara acak (*Mtry*). Mengurangi parameter *Mtry* dapat menghasilkan komputasi yang lebih cepat, tetapi memiliki pengaruh terhadap akurasi klasifikasi sedangkan pada *Ntree* dapat dibuat parameter sebesar mungkin karena pengklasifikasi RF efisien secara komputasi dan tidak *overfit*. Untuk menghitung kesalahan uji dataset atau nilai *error* pada algoritma *random forest* dapat menggunakan rumus:

$$MSE = n^{-1} \sum_{i=1}^n (y_i - \bar{y}_i)^2$$

Keterangan :

- *MSE* : Mean Squared Error (MSE)
- *n* : Jumlah sampel pada dataset
- $y_i$  : Nilai terukur data ke-i
- $\bar{y}_i$  : Nilai hasil dari observasi data ke-i

Berikut adalah contoh cara menghitung MSE untuk set data dengan beberapa nilai variabel data:

Actual values ( $y_i$ ): [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Predicted values ( $\bar{y}_i$ ): [1.9, 3.9, 6.9, 8.9, 11.9, 13.9, 15.9, 17.9, 19.9, 21.9]

Kemudian, kita dapat mengkalkulasikan nilai MSE:

$$n = 10$$

$$\sum (y_i - \bar{y}_i)^2 = (2 - 1.9)^2 + (4 - 3.9)^2 + \dots + (20 - 19.9)^2$$

$$\sum (y_i - \bar{y}_i)^2 = 0.01 + 0.09 + \dots + 0.01$$

$$\Sigma (y_i - \bar{y}_i)^2 = 0.1$$

$$MSE = (1/n) * \Sigma (y_i - \bar{y}_i)^2 = (1/10) * 0.1 = 0.01$$

Dalam contoh ini, MSE adalah 0,01, yang mengindikasikan perbedaan kuadrat rata-rata antara nilai aktual dan prediksi.

## 2.7 Algoritma *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) adalah metode dalam machine learning yang dirancang untuk menangani tugas klasifikasi dan analisis regresi. Algoritma ini pertama kali dikembangkan oleh Vladimir Vapnik pada tahun 1963 berdasarkan teori statistik untuk membangun model klasifikasi linier (Huang, Sujun *et al.*, 2018).

Tujuan utama dari *Support Vector Machine* adalah menemukan *hyperplane* optimal yang memaksimalkan margin, sehingga dapat memisahkan data secara efektif ke dalam dua kelas yang berbeda.

Untuk melakukan pengoptimalan pada nilai *hyperplane* dapat menggunakan persamaan berikut :

$$wx_i + b = 0 \quad (1)$$

Keterangan :

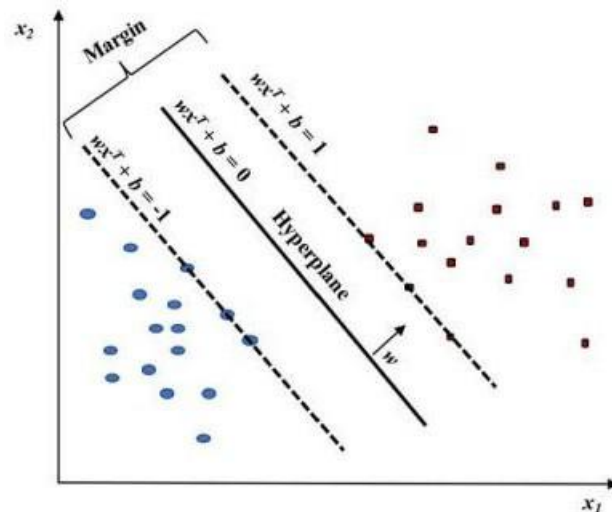
- $w$  : bobot vector (weight vector)
- $x$  : fitur input vector (x)
- $b$  : bias

Untuk nilai  $w$  dan  $b$  akan memenuhi ketidaksamaan berikut untuk semua elemen dari himpunan :

$$(w \cdot xi + b) \geq +1, \text{ jika } y_i = +1 \quad (2)$$

$$(w \cdot xi + b) \leq -1, \text{ jika } y_i = -1 \quad (3)$$

Sampel data pada persamaan (2) termasuk dalam kelas positif dan sampel data yang berada di bawah atau persamaan (3) termasuk dalam kelas negatif. Data yang terletak pada Persamaan (2) dan Persamaan (3) disebut dengan vektor pendukung (*support vector*). Struktur model *hyperplane* dapat ditunjukkan pada gambar 2.5 berikut



**Gambar 2. 5** Struktur Model *Hyperplane* Algoritma SVM (Huang, 2018)

Perolehan nilai dan pemisahan hyperplane pada Algoritma *Support Vector Machine*, dapat menggunakan persamaan berikut:

$$(w \cdot x_i) + b = 0$$

Keterangan :

- $w$  : vektor bobot (*weight vector*).
- $x$  : vektor fitur input ( $x$ ).
- $b$  : bias.

Untuk data  $x_i$  yang termasuk dalam kelas  $-1$ , persamaan yang berlaku adalah:

$$(w \cdot x_i + b) \leq -1$$

Sementara itu, untuk data  $x_i$  yang berada dalam kelas  $+1$ , persamaan yang digunakan adalah:

$$(w \cdot x_i + b) \geq 1$$

Persamaan ini memungkinkan SVM untuk menentukan margin optimal antara dua kelas dengan memisahkan data berdasarkan nilai-nilai bobot ( $w$ ) dan bias ( $b$ ), sehingga dapat menghasilkan hyperplane yang memaksimalkan jarak antar kelas.

Untuk perhitungan nilai *hyperplane* dapat menggunakan dataset yang berisikan dua kelas sebagai berikut:

Class 1: [1, 1], [2, 2], [3, 3], [4, 4]

Class 2: [5, 5], [6, 6], [7, 7], [8, 8]

Sekarang, kita dapat menguji persamaan *hyperplane* untuk setiap titik data:

1. Pertama hitung titik data pada kelas 1:

$$w * [1, 1] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [1, 1] + 0 = 0$$

$$1 + 1 + 0 = 2$$

Karena 2 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

2. Selanjutnya hitung titik data pada kelas 2:

$$w * [5, 5] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [5, 5] + 0 = 0$$

$$5 + 5 + 0 = 10$$

Karena 10 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

3. Hitung titik data kedua pada kelas 1:

$$w * [2, 2] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [2, 2] + 0 = 0$$

$$2 + 2 + 0 = 4$$

Karena 4 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

4. Hitung titik data kedua pada kelas 2:

$$w * [6, 6] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [6, 6] + 0 = 0$$

$$6 + 6 + 0 = 12$$

Karena 12 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

5. Hitung titik data ketiga pada kelas 1:

$$w * [3, 3] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [3, 3] + 0 = 0$$

$$3 + 3 + 0 = 6$$

Karena 6 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

6. Hitung titik data ketiga pada kelas 2:

$$w * [7, 7] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [7, 7] + 0 = 0$$

$$7 + 7 + 0 = 14$$

Karena 14 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

7. Hitung titik data keempat pada kelas 1:

$$w * [4, 4] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [4, 4] + 0 = 0$$

$$4 + 4 + 0 = 8$$

Karena 8 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

8. Hitung titik data keempat pada kelas 2:

$$w * [8, 8] + b = 0$$

Asumsikan data yang ada memiliki vektor bobot  $w =$  dan suku bias  $b = 0$ .

Persamaannya menjadi:

$$[1, 1] * [8, 8] + 0 = 0$$



$$8 + 8 + 0 = 16$$

Karena 16 tidak sama dengan 0, persamaan *hyperplane* gagal mengklasifikasikan titik data ini dengan benar.

Dalam contoh ini, persamaan *hyperplane* gagal mengklasifikasikan semua titik data dengan benar, mengindikasikan bahwa *hyperplane* mungkin bukan solusi optimal untuk set data ini.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Analisis bertujuan untuk mengidentifikasi kebutuhan dan persyaratan komponen yang diperlukan selama fase desain sistem dalam proses penelitian. Tahap ini berfungsi sebagai dasar dalam membangun sebuah sistem dengan memahami sebab dan akibat dari masalah yang dihadapi melalui analisis masalah pada tahap analisis sistem. Proses analisis ini menjadi langkah penting sebelum memasuki fase perancangan dan pengembangan sistem, karena memungkinkan identifikasi kebutuhan yang mendalam dan perencanaan yang lebih matang. Dengan melakukan analisis yang komprehensif, sistem yang dikembangkan tidak hanya lebih terstruktur tetapi juga lebih mampu memenuhi kebutuhan spesifik dan mencapai tujuan akhir secara efektif. Pendekatan ini juga membantu dalam mengantisipasi potensi hambatan selama implementasi, sehingga memastikan keberhasilan sistem yang dirancang.

##### **3.1.1 Analisis Masalah**

Dalam konteks untuk pengklasifikasian hasil pertandingan sepak bola, yang merupakan tugas prediktif dimana tujuan utama nya adalah memprediksi hasil pertandingan berdasarkan data historis. Untuk membandingkan suatu algoritma, akan menggunakan beberapa variabel fitur yang relevan seperti statistik tim, pemain, dan performa historis.

##### **3.1.2 Analisis Data**

Penelitian ini menggunakan *dataset* yang diambil dari website Premier League dan juga dari berbagai sumber publik atau dataset yang tersedia. Adapun dataset yang dihasilkan mencakup hasil pertandingan dan komponen-komponen yang telah ditentukan untuk proses klasifikasi.

##### **3.1.3 Analisis kebutuhan**

Analisis kebutuhan ditujukan untuk identifikasi dan mendefinisikan berbagai aspek yang diperlukan oleh pengguna dalam

merancang sebuah sistem. Proses ini dapat memaksimalkan bahwa sistem yang dikembangkan mampu memenuhi ekspektasi dan kebutuhan pengguna secara efektif.

Terdapat 2 jenis kebutuhan yang dapat diidentifikasi dalam perancangan sebuah sistem yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

#### 1. Kebutuhan fungsional

Merupakan kumpulan fitur atau fungsi yang harus ada dalam suatu sistem atau produk untuk memenuhi kebutuhan pengguna atau pemangku kepentingan. Ini mencakup spesifikasi fungsionalitas yang harus ada pada sistem untuk mencapai tujuan utama. Penelitian ini memiliki kebutuhan fungsional utama, yaitu:

- a. Sistem mampu mengumpulkan data dari berbagai sumber, termasuk statistik pertandingan, selisih gol, dan kondisi cuaca.
- b. Data yang dikumpulkan harus dalam format yang konsisten dan terstruktur untuk memudahkan analisis.
- c. Melakukan normalisasi atau standarisasi pada fitur-fitur yang digunakan agar algoritma dapat beroperasi dengan baik.
- d. Sistem harus mampu menerapkan algoritma Random Forest untuk klasifikasi, termasuk pengaturan parameter seperti jumlah pohon dan kedalaman maksimum.
- e. Implementasi algoritma SVM dengan kemampuan untuk memilih kernel yang tepat (*linear* atau *non-linear*) sesuai dengan karakteristik data.
- f. Pemisahan data ke dalam data latih dan data uji.
- g. Menentukan dan menggunakan metric evaluasi
- h. Menyediakan visualisasi yang jelas mengenai perbandingan akurasi dan metrik lainnya antara *Random Forest* dan *Support Vector Machine*.

## 2. Kebutuhan non-fungsional

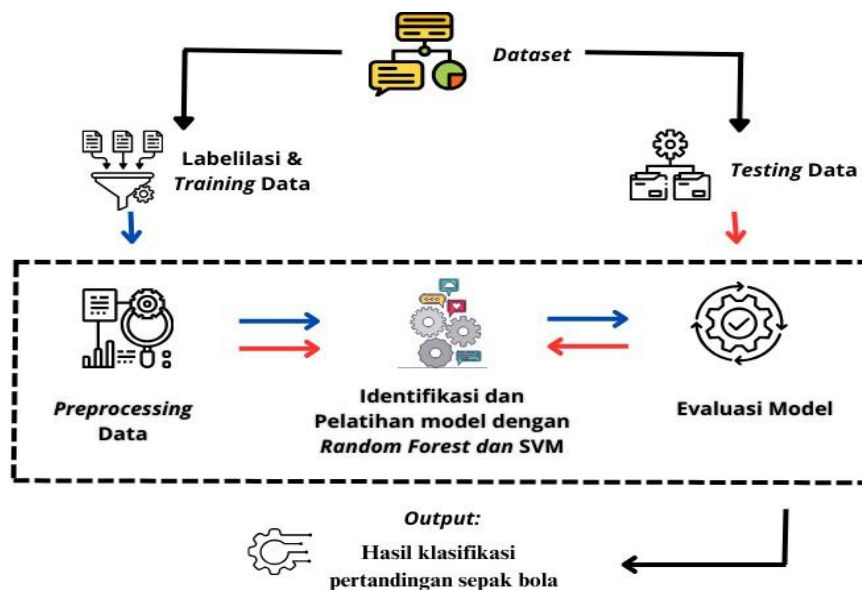
Kebutuhan non-fungsional dalam penelitian ini mengacu pada spesifikasi tambahan yang berfungsi sebagai pendukung sistem agar dapat berjalan secara optimal. Kebutuhan ini mencakup berbagai aspek seperti kinerja, yang memastikan sistem dapat bekerja dengan efisiensi tinggi, keamanan, yang melindungi data dan informasi dari akses tidak sah, serta batasan sistem yang menetapkan parameter operasional, seperti kapasitas penyimpanan atau waktu respons.

Selain itu, kebutuhan non-fungsional juga melibatkan skalabilitas untuk memastikan sistem dapat berkembang sesuai dengan peningkatan kebutuhan, serta keandalan untuk menjamin sistem tetap berfungsi meskipun menghadapi gangguan. Dengan memenuhi kebutuhan non-fungsional ini, sistem yang dirancang akan lebih stabil, aman, dan sesuai dengan ekspektasi pengguna.

- a. Model harus memberikan tingkat akurasi yang tinggi, misalnya di atas 85%, untuk memastikan prediksi hasil pertandingan yang dapat diandalkan.
- b. Hasil klasifikasi harus konsisten ketika model diuji dengan dataset yang berbeda.
- c. Sistem menggunakan manipulasi data menggunakan bahasa pemrograman *Python*.
- d. Kode sumber harus ditulis dengan baik dan terstruktur agar mudah dipelihara dan diperbarui.

## 3.2 Arsitektur Penelitian

Arsitektur umum sistem merupakan alur seluruh rangkaian sistem dalam penelitian. Pada penelitian ini, arsitektur umum sistem yang digunakan dapat dilihat pada gambar 3.1.



**Gambar 3. 1** Arsitektur Umum Sistem

- Menerima masukan berupa dataset latih dan uji
- Melakukan labelisasi *training data* dan *testing data*
- Preprocessing* data dilakukan pada data yang telah diproses, menggunakan algoritma *Random Forest* dan algoritma *Support Vector Machine* (SVM)
- Evaluasi model dilakukan pada dataset yang telah dilakukan *preprocessing data*
- Output* sistem berupa hasil klasifikasi pertandingan sepak bola Liga Utama Inggris

### 3.3 Perancangan Sistem

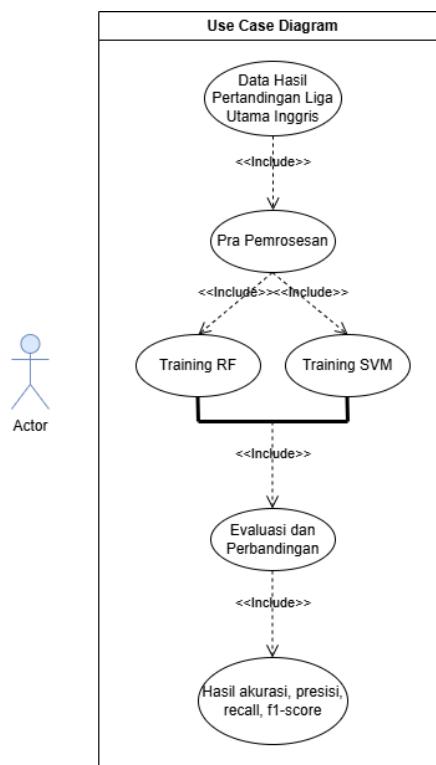
Pada penelitian ini menggunakan *use case diagram*, *activity diagram*, dan *sequence diagram* untuk menggambarkan proses pemodelan di seluruh tahap pemodelan sistem. Diagram ini dijelaskan sebagai berikut.

#### 3.3.1 Use Case Diagram

*Use Case Diagram* merupakan jenis diagram dalam Unified Modeling Language (UML) untuk memvisualisasikan interaksi antara aktor, yang dapat berupa pengguna eksternal atau sistem lain, dengan sistem yang sedang dirancang. Diagram ini bertujuan untuk memahami

dan menganalisis komponen-komponen utama yang diperlukan dalam pengembangan sistem. Dengan menyajikan hubungan antara aktor dan fungsi utama sistem, Use Case Diagram mempermudah identifikasi kebutuhan fungsional dan alur kerja yang relevan.

Pada penelitian ini, diagram tersebut dirancang untuk menggambarkan fungsi-fungsi sistem yang akan diimplementasikan. Diagram yang dihasilkan memberikan panduan visual yang jelas, sehingga proses perancangan dan pengembangan sistem menjadi lebih terarah dan terstruktur. Use Case Diagram dapat dilihat pada gambar 3.2.



**Gambar 3. 2** *Use Case Diagram*

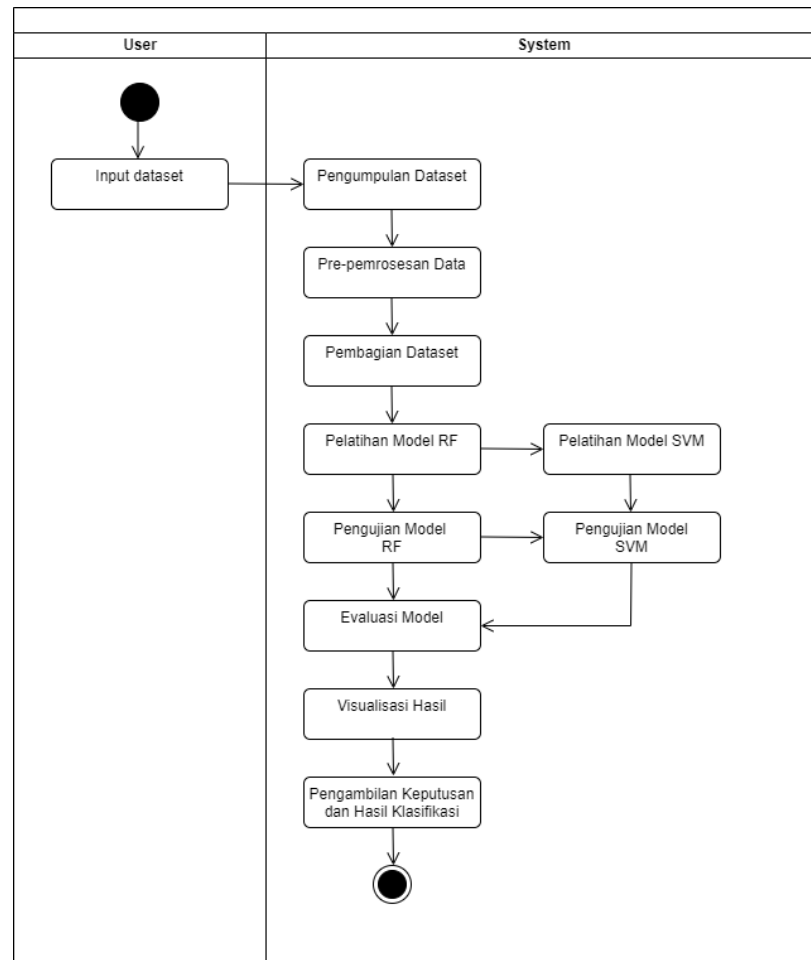
Adapun *narrative use case*, adalah metode untuk menilai dan mengkarakterisasi persyaratan perangkat lunak yang berusaha untuk mengklarifikasi dan mendeklarasikan bagaimana pengguna (aktor) terlibat dengan sistem yang sedang dikembangkan sebagai berikut.

- a. Data pertandingan Liga Utama Inggris berinteraksi dengan Sistem Analisis untuk mengumpulkan data dan melakukan pra-pemrosesan.

- b. Setelah data selesai dilakukan pre-pemrosesan, dilakukan penerapan pada kedua algoritma (*Random Forest* dan *Support Vector Machine*) untuk proses klasifikasi.
- c. Hasil dari RF dan SVM kemudian dibandingkan untuk menentukan klasifikasi hasil pertandingan sepak bola.
- d. Sistem memproses data dan menampilkan hasil klasifikasi pertandingan sepak bola untuk pengambilan keputusan.

### 3.3.2 Activity Diagram

Diagram aktivitas merupakan jenis diagram yang terdapat dalam *Unified Modeling Language* (UML) yang berfungsi untuk mengilustrasikan alur dari suatu sistem atau bagian dari sistem. Diagram ini menampilkan proses serta menunjukkan urutan dan kondisi yang mengarah pada transisi antara aktivitas-aktivitas tersebut. Dengan menggunakan *activity diagram*, pemangku kepentingan dapat memodelkan logika bisnis dan alur kerja sistem secara visual, sehingga memudahkan pemahaman tentang bagaimana suatu proses bekerja dan membantu dalam mengidentifikasi area untuk perbaikan atau peningkatan yang diperlukan. *Activity diagram* penelitian ini sebagai berikut.



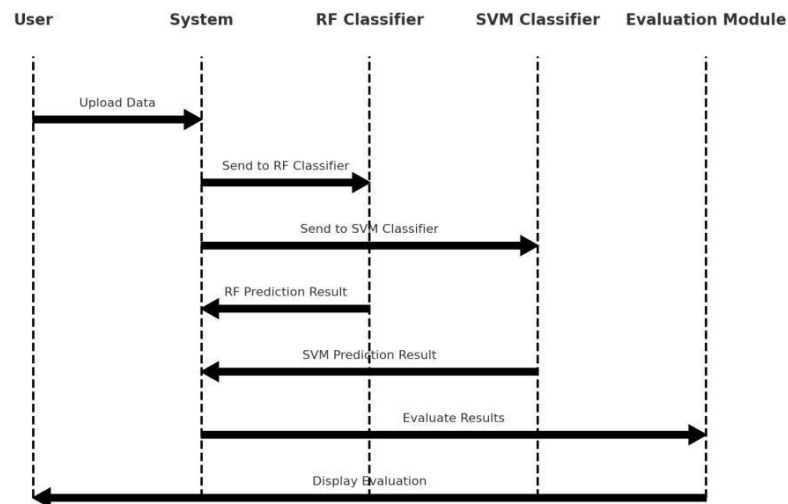
**Gambar 3. 3** *Activity Diagram*

Berdasarkan gambar 3.3 di atas, *activity diagram* ini memberikan gambaran tentang alur kerja dari proses perbandingan algoritma *Random Forest* dan *Support Vector Machine* dalam klasifikasi hasil pertandingan sepak bola, mulai dari pengumpulan data hingga pengambilan keputusan berdasarkan hasil analisis.

### 3.3.3 *Sequence Diagram*

Untuk dapat melihat interaksi setiap objek yang ada pada sistem, *sequence diagram* digambarkan pada gambar 3.4.





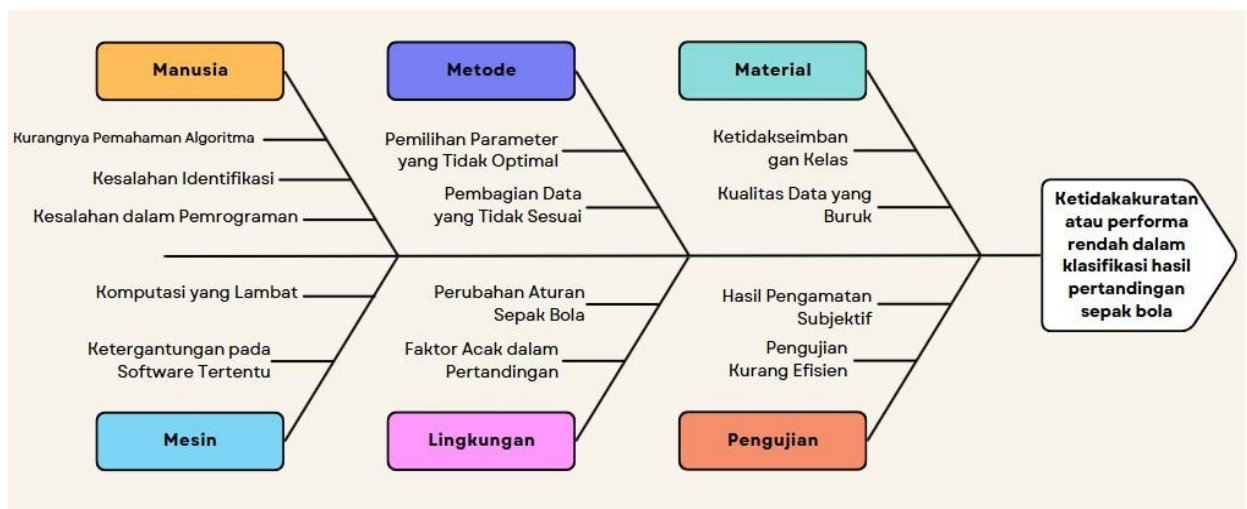
**Gambar 3. 4** *Sequence Diagram*

Pada gambar 3.4 merupakan *sequence diagram* yang dimulai dari *User* mengunggah data pertandingan sepak bola ke sistem. Kemudian, sistem mengirimkan data tersebut ke algoritma *Random Forest* untuk diproses dan mendapatkan hasil klasifikasi. Lalu, sistem juga mengirimkan data yang sama ke algoritma *Support Vector Machine* untuk diproses. Setelah itu *RF Classifier* dan *SVM Classifier* mengembalikan hasil klasifikasi berupa hasil menang, seri, atau kalah. Sistem mengirimkan kedua hasil klasifikasi ke modul evaluasi untuk dibandingkan berdasarkan kriteria performa berupa hasil metrik evaluasi. sistem mengirimkan kedua hasil klasifikasi ke hasil metrik evaluasi untuk dibandingkan berdasarkan kriteria performa.

### 3.4 Ishikawa Diagram

Diagram Ishikawa, juga dikenal sebagai *diagram fishbone* atau *diagram sebab-akibat*. Diagram ini memberikan pendekatan terstruktur dan sistematis untuk mengidentifikasi dan menyusun penyebab potensial dari suatu efek atau masalah tertentu. Proses dalam mengumpulkan data dan mengatur penyebab potensial dapat mencakup identifikasi hambatan, fasilitator dan insentif untuk suatu

perilaku, meninjau literatur, menganalisis diagram alir, melakukan analisis mode dan efek kegagalan (FMEA).



**Gambar 3.5** Ishikawa Diagram

Gambar 3.5 menunjukkan diagram ishikawa dimana penyebab utama dikelompokkan ke dalam beberapa kategori, seperti manusia, metode, material, mesin, lingkungan, pengujian, dan data. Untuk proses bagaimana diagram ishikawa bekerja dapat dilihat pada penjelasan berikut

**a. Manusia**

1. **Kurangnya Pemahaman Algoritma** : Peneliti atau pengembang kurang memahami cara kerja algoritma Random Forest dan SVM.
2. **Kesalahan dalam Pemrograman** : Terdapat bug atau implementasi kode yang tidak sesuai dengan prosedur.

**b. Metode**

1. **Pemilihan Parameter yang Tidak Optimal** : Parameter seperti C, kernel pada SVM, atau jumlah pohon pada Random Forest tidak diatur dengan benar.
2. **Pembagian Data yang Tidak Sesuai** : Data latih dan data uji mungkin tidak terdistribusi dengan baik.

### c. **Material**

1. **Ketidakseimbangan Kelas** : Jumlah data untuk kategori tertentu (misalnya, menang, seri, kalah) tidak merata.
2. **Kualitas Data yang Buruk** : Data yang digunakan memiliki banyak missing value atau noise.

### d. **Mesin**

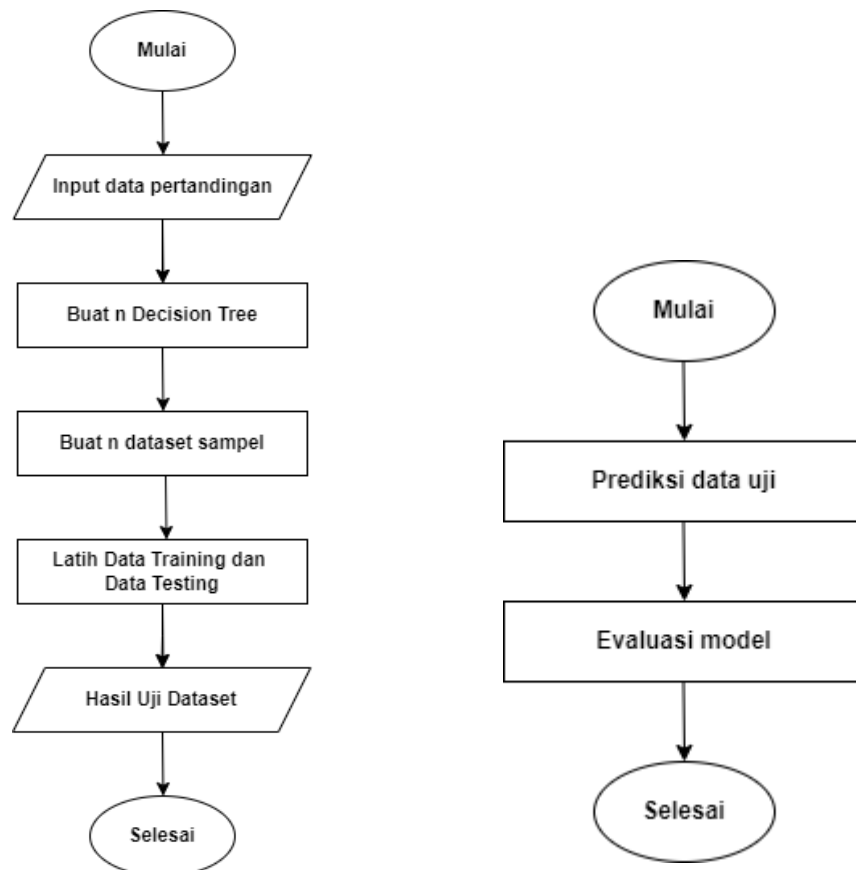
1. **Komputasi yang Lambat** : Performa perangkat keras tidak cukup untuk menangani dataset besar.
2. **Ketergantungan pada Software Tertentu** : Perbedaan versi pustaka (library) atau bug pada tools yang digunakan.

### e. **Lingkungan**

1. **Perubahan Aturan Sepak Bola** : Faktor eksternal seperti perubahan dalam format liga atau aturan permainan dapat memengaruhi hasil pertandingan.
2. **Faktor Acak dalam Pertandingan** : Cuaca, cedera pemain, atau keputusan wasit dapat memengaruhi hasil pertandingan secara signifikan.

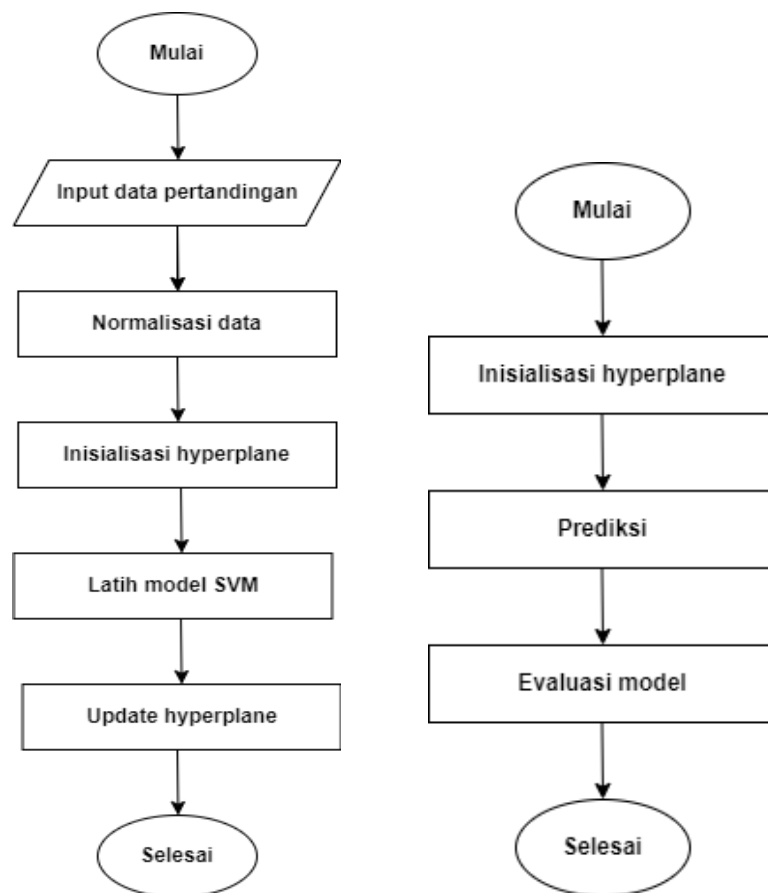
## 3.5 **Flowchart**

*Flowchart* dapat digunakan untuk menunjukkan urutan langkah-langkah dan keputusan yang diperlukan untuk melakukan alur proses dan rangkaian proses secara bertahap yang diimplementasikan pada algoritma yang digunakan dalam sistem.



**Gambar 3. 6** Flowchart Sistem Random Forest

Gambar 3.6 bagian *flowchart* dari algoritma *Random Forest* yang digunakan untuk proses klasifikasi hasil pertandingan liga utama inggris musim 2022/2023. Proses dimulai dengan menginputkan dataset yang dibutuhkan untuk proses klasifikasi, kemudian dilanjutkan dengan membuat *n decision tree*. Setelah itu, dataset tersebut dibagi menjadi *n* dataset sampel dengan penggantian. Proses kemudian berlanjut seperti sebelumnya, dengan setiap dataset sampel diproses secara terpisah, lalu latih data training dan data testing. Kemudian, akan didapatkan hasil uji dari dataset. Proses prediksi berdasarkan hasil uji yang didapat. Proses dimulai dengan melakukan prediksi data uji dari setiap sampel dataset dihasilkan. Setelah itu, hasil prediksi dari setiap dataset sampel digabungkan dan agregasi dilakukan untuk menghasilkan hasil prediksi akhir dan akan dilakukan evaluasi kinerja model menggunakan metrik evaluasi yang sesuai.



**Gambar 3. 7** Flowchart Sistem Support Vector Machine

Gambar 3.7 merupakan *flowchart* dari algoritma SVM yang digunakan untuk proses klasifikasi hasil pertandingan liga utama inggris musim 2022/2023. Proses dimulai dengan menginputkan dataset yang dibutuhkan untuk proses klasifikasi, kemudian dilanjutkan dengan normalisasi data. Setelah melakukan normalisasi data, dilakukan inisialisasi *hyperplane*. *Hyperplane* berfungsi sebagai pemisah antara dua kelas dalam data. SVM berusaha menemukan *hyperplane* yang tidak hanya memisahkan kelas-kelas tersebut, tetapi juga memaksimalkan margin antara titik data terdekat dari masing-masing kelas. Dengan memaksimalkan margin, SVM cenderung mengurangi risiko *overfitting*. *Hyperplane* yang optimal membantu memastikan bahwa model tidak hanya cocok dengan data pelatihan, tetapi juga dapat memprediksi dengan baik pada data yang tidak terlihat sebelumnya. Proses kemudian berlanjut seperti sebelumnya, dengan setiap dataset sampel diproses secara terpisah, lalu latih data training dan data testing. Kemudian, akan didapatkan hasil uji dari dataset.

Proses prediksi berdasarkan hasil uji yang didapat. Proses dimulai dengan melakukan prediksi data uji dari setiap sampel dataset dihasilkan. Setelah itu, hasil prediksi dari setiap dataset sampel digabungkan dan agregasi dilakukan untuk menghasilkan hasil prediksi akhir dan akan dilakukan evaluasi kinerja model.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi Sistem

Sebuah sistem untuk melatih model dikembangkan dengan beberapa komponen pendukung. Adapun komponen tersebut berfungsi untuk proses pelatihan dengan sistem mampu menjalankan prosedur pelatihan model Random Forest dan SVM dengan efisiensi dan efektivitas yang tinggi ketika perangkat terintegrasi sempurna.

##### 4.4.1 Perangkat Keras

Untuk proses kelancaran penelitian, digunakan perangkat yang sesuai dengan penelitian sebagai berikut:

- a. *AMD Ryzen 3 5300U with Radeon Graphics*                      2.60 GHz
- b. RAM 16GB.
- c. Sistem operasi *Windows 11 Home Single Language* 64-bit.
- d. *Browser Google Chrome* dan *Microsoft Edge*.

##### 4.4.2 Perangkat Lunak

Adapun perangkat lunak atau library yang digunakan pada penelitian ini yaitu:

- a. *Visual Studio Code*
- b. *Python 3.7*.
- c. *Library pandas, numpy, nltk, sklearn, gensim, transformers, tensorflow, torch, matplotlib, scikit-learn* dan *streamlit*

#### 4.2 Analisis Deskriptif

Musim Liga Utama Inggris 2022–2023 merupakan musim ke-124 dalam sejarah liga sepak bola profesional tingkat tertinggi di Inggris.

Pada 20 Mei 2023, Manchester City berhasil meraih gelar juara, menjadikannya kemenangan ketiga mereka dalam dua musim terakhir. Namun,

di sisi lain, tiga tim yaitu Leicester City, Leeds United, dan Southampton harus menerima kenyataan pahit terdegradasi dari liga utama. Dengan demikian, mereka akan melanjutkan perjuangan mereka di divisi Kejuaraan EFL pada musim berikutnya. Musim ini tidak hanya mencatatkan prestasi tetapi juga menjadi pelajaran penting bagi tim-tim yang berjuang untuk bertahan di kompetisi papan atas.

**Tabel 4. 1** Klasemen Akhir Liga Utama Inggris 2022-2023

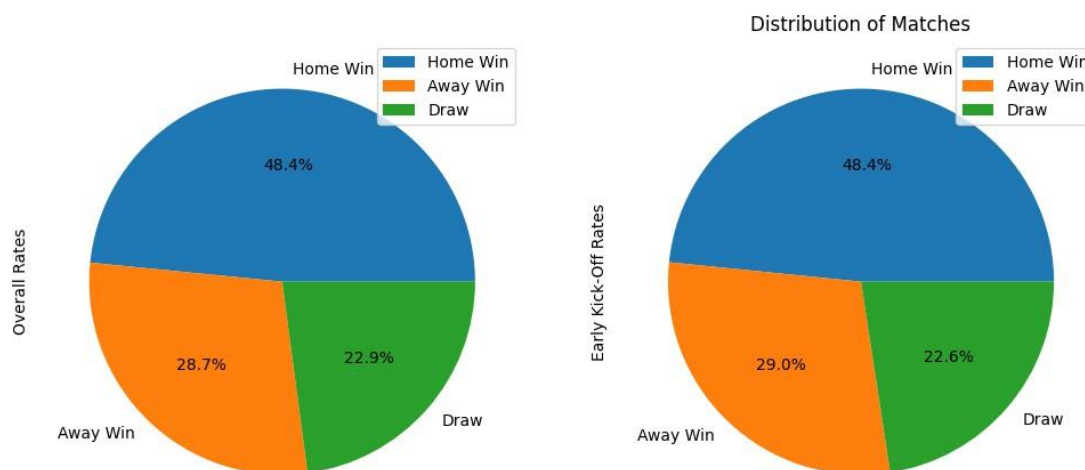
Position	Club	Played	Won	Drawn	Lost	GF	GA	GD	Points
1	Manchester City	38	28	5	5	94	33	61	89
2	Arsenal	38	26	6	6	88	43	45	84
3	Manchester Utd	38	23	6	9	58	43	15	75
4	Newcastle Utd	38	19	14	5	68	33	35	71
5	Liverpool	38	19	10	9	75	47	28	67
6	Brighton	38	18	8	12	72	53	19	62
7	Aston Villa	38	18	7	13	51	46	5	61
8	Tottenham	38	18	6	14	70	63	7	60
9	Brentford	38	15	14	9	58	46	12	59
10	Fulham	38	15	7	16	55	53	2	52
11	Crystal Palace	38	11	12	15	40	49	-9	45
12	Chelsea	38	11	11	16	38	47	-9	44
13	Wolves	38	11	8	19	31	58	-27	41
14	West Ham	38	11	7	20	42	55	-13	40
15	Bournemouth	38	11	6	21	37	71	-34	39
16	Nott'ham Forest	38	9	11	18	38	68	-30	38
17	Everton	38	8	12	18	34	57	-23	36
18	Leicester City	38	9	7	22	51	68	-17	34
19	Leeds United	38	7	10	21	48	78	-30	31
20	Southampton	38	6	7	25	36	73	-37	25

Pada table 4.1 Selama 380 pertandingan yang berlangsung di Liga Utama Inggris musim 2022–2023, tim tuan rumah mencatatkan kemenangan dalam 184 pertandingan, menunjukkan keunggulan bermain di kandang. Sementara itu, tim tamu berhasil meraih kemenangan dalam 109 pertandingan, menyoroti tantangan tambahan yang dihadapi saat bertanding di luar markas mereka. Sisanya, sebanyak 87 pertandingan, berakhir dengan hasil imbang, mencerminkan keseimbangan kekuatan di beberapa pertandingan.

Distribusi ini memberikan gambaran mengenai dinamika hasil pertandingan di musim tersebut. Untuk memberikan pemahaman lebih jelas, persentase



kemenangan dari masing-masing kategori dapat divisualisasikan melalui grafik yang ditampilkan pada gambar berikut. Analisis semacam ini berguna untuk memahami pola performa tim, baik di kandang maupun tandang, serta faktor yang memengaruhi hasil pertandingan.



**Gambar 4. 1** Perbandingan Persentase Kemenangan

#### 4.3 Klasifikasi Menggunakan Data Asli

##### 4.3.1 Struktur Data dan Pembagian Data *Training* dan *Testing*

Dalam proses klasifikasi menggunakan data asli, terdapat beberapa tahapan yang harus dilakukan untuk mengolah dan menganalisis data. Pada penelitian ini, struktur data yang digunakan mencakup 380 observasi, dengan satu variabel dependen (y) dan beberapa variabel independen (x). Variabel dependen terdiri dari tiga kategori hasil pertandingan, yaitu home, away, dan draw.

Sementara itu, variabel independen terdiri dari nilai integer yang mewakili beberapa indikator statistik dalam pertandingan, seperti total shots fouls, offsides, yellow card, dan red card. Data ini kemudian akan dibagi menjadi data training dan testing untuk melatih model klasifikasi sekaligus menguji performanya. Proses pembagian data ini bertujuan untuk memastikan bahwa model dapat memprediksi hasil pertandingan secara akurat berdasarkan variabel yang tersedia.

	Date	Time	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	...	AST	HF	AF	HC	AC	HY	AY	HR	AR	Result_Match
0	05/08/2022	20:00	Crystal Palace	Arsenal	0	2	A	0	1	A	...	2	16	11	3	5	1	2	0	0	True
1	06/08/2022	12:30	Fulham	Liverpool	2	2	D	1	0	H	...	4	7	9	4	4	2	0	0	0	False
2	06/08/2022	15:00	Bournemouth	Aston Villa	2	0	H	1	0	H	...	2	18	16	5	5	3	3	0	0	True
3	06/08/2022	15:00	Leeds	Wolves	2	1	H	1	1	D	...	6	13	9	6	4	2	0	0	0	False
4	06/08/2022	15:00	Newcastle	Nottingham	2	0	H	0	0	D	...	0	9	14	11	1	0	3	0	0	False

5 rows × 24 columns

**Gambar 4.2** Tipe Data Variabel Penelitian

Proses klasifikasi hasil pertandingan sepak bola ditunjukkan pada Gambar 4.2, di mana tipe data variabel penelitian digunakan. Sebelum diklasifikasikan, data yang pertama digunakan dibagi menjadi bagian data pelatihan dan data pengujian. Jumlah total data yang digunakan sebanyak 380 data. Untuk pembagian data, dapat dilihat pada gambar 4.3 dan gambar 4.4.

```
Data Training:
      HomeTeam  AwayTeam  FTHG  FTAG  HS  AS
89           8         2     2     2  19  10
176          8         5     2     1   8  20
147          6         8     0     3   4  23
211          0         3     1     1  23   9
92           1         5     0     2  18   8
89          D
176          H
147          A
211          D
92           A
Name: FTR, dtype: object
```

**Gambar 4.3** Data Training

```
Data Testing:
      HomeTeam  AwayTeam  FTHG  FTAG  HS  AS
266         16         17     3     3  19  17
261          4          6     1     0  11  11
265          3         10     1     1  11  14
39          15         17     0     2  17  18
33           5         10     2     1   7  17
266      D
261      H
265      D
39       A
33       H
Name: FTR, dtype: object
```

**Gambar 4. 4** *Data Testing*

### 4.3.2 Klasifikasi Metode Random Forest

#### A. Proses Pelatihan dan Pembentukan Model

Tahap proses dan pembentukan model, dibuat dengan menggunakan nilai parameter yang menentukan jumlah *trees* yang akan dibangun. Dalam tahap ini, parameter yang digunakan sebanyak 100 estimator. Sebelum melakukan pelatihan dan pembentukan model, terlebih dahulu menentukan fitur dan target.

```
# Menentukan fitur dan target
X = df[['HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'HS', 'AS']]
Y = df['FTR']
```

**Gambar 4. 5** Fitur dan Target

Pada gambar 4.5 merupakan fitur dan target yang digunakan pada sistem dan proses klasifikasi. Adapun fitur [X] merupakan variabel independen data yang digunakan model untuk membuat proses klasifikasi dan prediksi. Sedangkan target [Y] adalah variabel dependen yang menunjukkan kategori atau kelas dari data yang menghasilkan output model.

```
# Membagi data menjadi data latih dan uji
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)
```

**Gambar 4. 6** Data Latih dan Data Uji

Pada Gambar 4.6, proses pembagian dataset menjadi data latih dan data uji diperlihatkan untuk melatih dan mengevaluasi model. Pembagian data ini bertujuan agar model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Selain itu, untuk memastikan pembagian data yang konsisten setiap kali dilakukan, fungsi `random_state=42` diterapkan. Fungsi ini menetapkan nilai seed dalam proses pengacakan data, sehingga pembagian data latih dan uji dapat direproduksi dengan hasil yang sama pada setiap percobaan, memastikan keandalan dan keterulangan hasil eksperimen.

```
# Membuat model Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, Y_train)

0.1s
```

**Gambar 4. 7** Model *Random Forest*

Gambar 4.7 merupakan tahap pada sistem dalam membuat model untuk proses klasifikasi pada *random forest*. Model dilatih dengan menggunakan 100 pohon Keputusan serta parameter acak untuk memastikan hasil yang dapat direproduksi kembali.

#### **B. Implementasi Klasifikasi Hasil Pertandingan dengan *Random Forest***

Pada tahap sebelumnya telah dilakukan proses untuk menentukan beberapa tahap, mulai dari menentukan fitur dan tujuan, membuat data latihan dan data uji, dan membuat model hutan acak. Untuk menguji dan memastikan akurasi model yang dihasilkan, pengujian akan dilakukan pada data latihan dan data latihan.

Adapun tahapan nya yaitu dengan membuat fungsi yang digunakan untuk menghitung metrik evaluasi seperti *precision*, recall, f1-score, dan accuracy berdasarkan label sebenarnya. Rumus untuk menghitung metrik evaluasi yaitu.

$$\text{Akurasi} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F1-Score} = 2 * (\text{Presisi} * \text{Recall}) / (\text{Presisi} + \text{Recall})$$

Fungsi „*precision\_score*“, „*recall\_score*“, „*f1-score*“, dan „*accuracy\_score*“ dari library „*sklearn.metrics*“ secara otomatis menghitung nilai-nilai tersebut berdasarkan label sebenarnya („*y\_true*“) dan label prediksi („*y\_pred*“).

```
# Metrik evaluasi tambahan
print("\nMetrik Evaluasi:")
print(classification_report(Y_test, Y_pred))
```

**Gambar 4. 8** Proses Menghitung Metrik Evaluasi

### C. Evaluasi

Setelah dilakukan proses implementasi ekstraksi fitur dan klasifikasi hasil pertandingan menggunakan *Random Forest*, selanjutnya dilakukan proses evaluasi terhadap analisis klasifikasi hasil pertandingan sepak bola. Proses evaluasi model menggunakan data latih dan data uji.

```
# Prediksi dan evaluasi model
Y_pred = model.predict(X_test)

# Calculate the percentage of each outcome ('H', 'D', 'A')
ftr_counts = {outcome: list(Y_pred).count(outcome) for outcome in set(Y_pred)}
total_predictions = len(Y_pred)

# Print the percentages
for outcome, count in ftr_counts.items():
    percentage = (count / total_predictions) * 100
    print(f'{outcome}: {percentage:.2f}%')
```

**Gambar 4. 9** Proses Evaluasi Model

Pada Gambar 4.9, proses evaluasi model dilakukan dengan menggunakan data testing untuk mengukur kinerja model yang telah dilatih sebelumnya. Model yang telah dibangun digunakan untuk memprediksi hasil pada data uji (*X\_test*), dan hasil prediksi ini disimpan dalam variabel *Y\_pred*, yang berisi array berisi klasifikasi untuk setiap contoh dalam *X\_test*. Klasifikasi ini dapat berupa kelas-kelas seperti „H“ (Home Win), „D“ (Draw), dan „A“ (Away Win). Untuk menganalisis hasilnya, digunakan dictionary comprehension untuk menghitung frekuensi kemunculan masing-masing kelas dalam hasil prediksi. Kemudian,

persentase kemunculan setiap kelas dihitung dan dicetak, memberikan gambaran distribusi hasil klasifikasi.

Selanjutnya, menggunakan *dictionary comprehension* untuk menghitung jumlah kemunculan setiap kelas (outcome) dalam hasil prediksi „ $Y_{pred}$ “. Untuk setiap „*outcome*“ (kelas) dan „*count*“ (jumlah kemunculan), akan dihitung persentase kemunculan kelas lalu hasilnya dicetak dan mendapatkan output yang menunjukkan persentase setiap kelas. Setelah itu, dilakukan perhitungan metrik evaluasi seperti *precision*, *recall*, *F1-score*, dan akurasi.

Selanjutnya, proses yang dilakukan adalah menentukan dan membuat grafik metrik evaluasi seperti *precision*, *recall*, *F1-score*, dan akurasi. Untuk keseleruhan bagaimana proses menentukan grafik metrik evaluasi dapat dilihat pada gambar 4.10.

```

import matplotlib.pyplot as plt
import numpy as np

# Data from the classification report
classes = ['A', 'D', 'H']
precision = [1.00, 0.91, 1.00]
recall = [0.98, 1.00, 0.97]
f1_score = [0.99, 0.95, 0.99]

# Macro and weighted averages
macro_avg = [0.97, 0.98, 0.98]
weighted_avg = [0.98, 0.98, 0.98]

# Create a figure for the plots
plt.figure(figsize=(18, 8))

# Bar width
bar_width = 0.2
index = np.arange(len(classes))

# Plotting Precision, Recall, and F1-score
plt.subplot(1, 2, 1)
plt.bar(index, precision, bar_width, label='Precision', color='blue')
plt.bar(index + bar_width, recall, bar_width, label='Recall', color='orange')
plt.bar(index + 2 * bar_width, f1_score, bar_width, label='F1-score', color='green')

# Adding Macro Averages
plt.bar(len(classes), macro_avg[0], bar_width, label='Macro Avg Precision', color='cyan', alpha=0.5)
plt.bar(len(classes), macro_avg[1], bar_width, color='magenta', alpha=0.5, bottom=macro_avg[0])
plt.bar(len(classes), macro_avg[2], bar_width, color='yellow', alpha=0.5, bottom=np.array(macro_avg[:2]).sum())

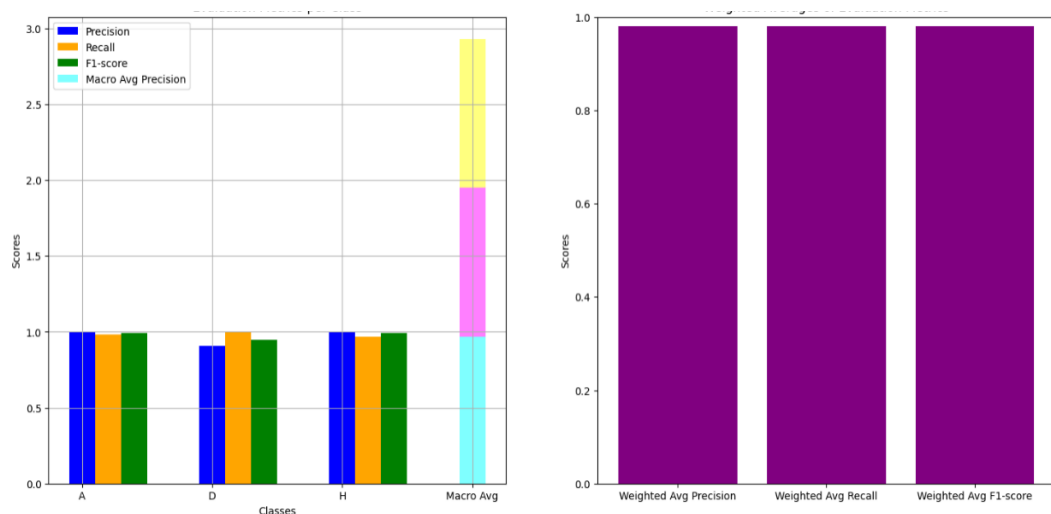
# Adjusting the x-ticks
plt.xlabel('Classes')
plt.ylabel('Scores')
plt.title('Evaluation Metrics per Class')
plt.xticks(np.append(index, len(classes)), classes + ['Macro Avg'])
plt.legend()
plt.grid()

# Plotting Weighted Averages
plt.subplot(1, 2, 2)
plt.bar(['Weighted Avg Precision', 'Weighted Avg Recall', 'Weighted Avg F1-score'], weighted_avg, color='purple')
plt.ylabel('Scores')
plt.title('Weighted Averages of Evaluation Metrics')
plt.ylim(0, 1)

plt.show()

```

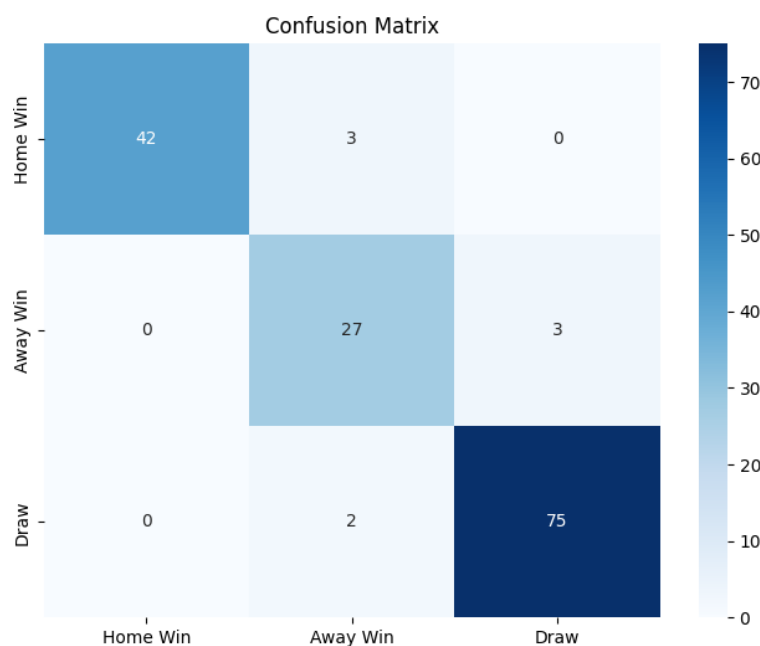
**Gambar 4. 10** Proses Membuat Grafik Metrik Evaluasi



Pada **Gambar 4. 11** Hasil Grafik Akurasi, F1-Score, Precision, dan Recall

gambar 4.11, akurasi pelatihan mencapai rata-rata mencapai angka 98% dengan 40% dari total dataset yang digunakan sebagai data uji, sedangkan 80% akan digunakan untuk melatih model. Hal ini menunjukkan bahwa model belajar untuk mengklasifikasikan data dengan cukup baik.

*Confusion Matrix* merupakan tabel yang memvisualisasikan kinerja suatu model klasifikasi. Dengan begitu, dapat dilihat sejauh mana model mampu membedakan antara hasil yang benar dan salah dalam klasifikasi.



**Gambar 4. 12** Hasil Confusion Matrix

Gambar 4.12 memperlihatkan hasil dari confusion matrix yang dihasilkan, yang menggambarkan pembagian antara True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Matrix ini memberikan gambaran yang jelas tentang bagaimana model mengklasifikasikan hasil pertandingan, baik dengan benar maupun keliru. Dengan menggunakan informasi dari confusion matrix tersebut, berbagai metrik evaluasi dapat dihitung untuk setiap kelas dalam klasifikasi hasil pertandingan sepak bola. Perhitungan ini sangat penting untuk menilai sejauh mana model mampu memprediksi hasil pertandingan dengan tepat. Tabel 4.1 memberikan informasi yang diperlukan



untuk menghitung metrik evaluasi ini, yang selanjutnya digunakan untuk mengevaluasi kinerja keseluruhan model dalam tugas klasifikasi.

a. Akurasi (*Accuracy*)

Berdasarkan *Confusion Matrix* yang telah dijabarkan sebelumnya, nilai akurasi (*Accuracy*) dihitung menggunakan rumus yang telah ditetapkan. Akurasi memberikan indikasi sejauh mana model pada sistem dapat melakukan klasifikasi dengan tepat.

Metrics for Home Win (H):

$$Akurasi = \frac{\text{Banyak data uji yang sesuai}}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (a)$$

$$Akurasi = \frac{42+27+75}{42+107+0+3} \times 100\% = 94\%$$

Metrics for Away Win (A):

$$Akurasi = \frac{\text{Banyak data uji yang sesuai}}{TP+TN+FP+FN} \times 100\%$$

$$Akurasi = \frac{42+27+75}{75+72+3+2} \times 100\% = 94\%$$

Metrics for Draw (D):

$$Akurasi = \frac{\text{Banyak data uji yang sesuai}}{TP+TN+FP+FN} \times 100\%$$

$$Akurasi = \frac{42+27+75}{27+117+5+3} \times 100\% = 94\%$$

Dari hasil perhitungan nilai akurasi di atas, didapatkan nilai akurasi sebesar 94%.

b. Presisi (*Precision*)

Precision (presisi) merupakan salah satu metrik evaluasi penting dalam analisis klasifikasi yang mengukur tingkat ketepatan model dalam mengidentifikasi hasil positif. Metrik ini dihitung dengan membandingkan jumlah True Positive (TP) dengan total prediksi positif yang dilakukan oleh

model, termasuk baik yang benar maupun yang salah. Dalam hal ini, False Positive (FP) merujuk pada jumlah kasus yang salah diklasifikasikan sebagai positif oleh model, meskipun sebenarnya mereka termasuk dalam kategori negatif. Precision yang tinggi menunjukkan bahwa model sangat akurat dalam mengklasifikasikan hasil positif, mengurangi kemungkinan kesalahan yang disebabkan oleh prediksi yang salah.

Metrics for Home Win (H):

$$Precision = \frac{True\ Positif}{TP+FP} \times 100\% \dots\dots\dots (b)$$

$$Precision = \frac{42}{42+0} \times 100\% = 100\%$$

Metrics for Away Win (A):

$$Precision = \frac{True\ Positif}{TP+FP} \times 100\%$$

$$Precision = \frac{75}{75+3} \times 100\% = 96\%$$

Metrics for Draw (D):

$$Precision = \frac{True\ Positif}{TP+FP} \times 100\%$$

$$Precision = \frac{27}{27+5} \times 100\% = 84\%$$

Dari hasil perhitungan nilai presisi di atas, didapatkan nilai presisi untuk setiap metric evaluasi *home*, *away*, *draw* sebesar 100%, 96%, 84%.

### c. Recall

*Recall* adalah metrik yang memberikan informasi tentang kemampuan model dalam mengidentifikasi sebanyak mungkin kasus positif pada model klasifikasi yang sebenarnya positif.

Metrics for Home Win (H):

$$Recall = \frac{True\ Positif}{TP+FN} \times 100\% \dots\dots\dots (c)$$

$$Recall = \frac{42}{42+3} \times 100\% = 93\%$$

Metrics for Away Win (A):

$$Recall = \frac{True\ Positif}{TP+FN} \times 100\%$$

$$Recall = \frac{75}{75+2} \times 100\% = 97\%$$

Metrics for Draw (D):

$$Recall = \frac{True\ Positif}{TP+FN} \times 100\%$$

$$Recall = \frac{27}{27+3} \times 100\% = 90\%$$

Dari hasil perhitungan nilai *recall* di atas, didapatkan nilai *recall* untuk setiap metrik evaluasi *home*, *away*, *draw* sebesar 93%, 97%, 90%.

d. *F1-Score*

*F1-Score* adalah metrik rata-rata harmonis antara Precision dan Recall.

Metrics for Home Win (H):

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\% \dots\dots\dots (d)$$

$$F1 - Score = \frac{2 \times (100 \times 93)}{100+93} \times 100\% = 96,37\%$$

Metrics for Away Win (A):

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\%$$

$$F1 - Score = \frac{2 \times (96 \times 97)}{96+97} \times 100\% = 96,49\%$$

Metrics for Draw (D):

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\%$$

$$F1 - Score = \frac{2 \times (84 \times 90)}{84 + 90} \times 100\% = 86,89\%$$

Dari hasil perhitungan nilai *F1-Score* di atas, didapatkan nilai *F1-Score* untuk setiap metrik evaluasi *home*, *away*, *draw* sebesar 96,37%, 96,49%, 86,89%.

**Tabel 4. 2** Hasil Akurasi, Precision, Recall, dan F1-Score

Kelas	Akurasi (TP + TN) / (TP + TN + FP + FN)	<i>Precision</i> TP / (TP + FP)	<i>Recall</i> TP / (TP + FN)	<i>F1-Score</i> 2 * (Presisi * Recall) / (Presisi + Recall)
Home	94%	100%	93%	96,37%
Away	94%	96%	97%	96,49%
Draw	94%	84%	90%	86,89%

Tabel 4.2 menunjukkan hasil metrik evaluasi dari proses klasifikasi hasil pertandingan sepak bola dengan variabel („H“, „A“, „D“) yaitu *F1-Score* sebesar 96,37%, 96,49%, 86,89%, *precision* sebesar 100%, 96%, 84%, dan *recall* sebesar 93%, 97%, 90%.

### 4.3.3 Klasifikasi Metode Support Vector Machine

Beberapa jenis kernel yang umum digunakan dalam SVM antara lain kernel linear, kernel radial basis function (RBF), dan kernel polynomial.

Setiap jenis kernel ini memiliki keunggulan masing-masing, tergantung pada karakteristik data yang digunakan. Kernel linear biasanya digunakan ketika data dapat dipisahkan secara linier, sementara kernel RBF lebih efektif untuk menangani data yang memiliki hubungan non-linier. Sementara itu, kernel polynomial memungkinkan model untuk menangani hubungan yang lebih kompleks antara fitur-fitur data. Pemilihan kernel yang tepat akan sangat

mempengaruhi kemampuan model dalam membuat prediksi yang akurat dan efisien

Pemilihan jenis kernel bertujuan untuk menyesuaikan model dengan karakteristik data, sehingga memungkinkan evaluasi yang akurat terhadap kinerja model. Dengan menggunakan kernel yang tepat serta pengaturan parameter yang optimal, model yang dihasilkan tidak hanya mampu mengklasifikasikan data dengan baik, tetapi juga memberikan tingkat akurasi yang tinggi. Selain itu, pemahaman mendalam mengenai masing-masing jenis kernel dapat membantu dalam mengidentifikasi pola-pola dalam data yang mungkin tidak terlihat secara langsung, sehingga meningkatkan kemampuan generalisasi model.

#### a. Proses Pembentukan dan Pelatihan Model

Tahap proses dan pembentukan model, dibuat dengan menggunakan nilai parameter yang menentukan jumlah *trees* yang akan dibangun. Dalam tahap ini, parameter yang digunakan sebanyak 100 estimator. Sebelum melakukan pelatihan dan pembentukan model, terlebih dahulu menentukan fitur dan target.

```
# Memilih fitur dan label
X = df[['HomeTeam', 'AwayTeam', 'FTHG', 'FTAG', 'HS', 'AS']]
y = df['FTR']
```

**Gambar 4. 13** Fitur dan Target

Gambar 4.13 merupakan fitur dan target yang digunakan pada sistem dan proses klasifikasi. Adapun fitur [X] merupakan variabel independen data yang digunakan model untuk membuat proses klasifikasi dan prediksi. Sedangkan target [y] adalah variabel dependen yang menunjukkan kategori atau kelas dari data yang menghasilkan output model.

```
# Membagi dataset menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8, random_state=42)
```

**Gambar 4. 14** Data Latih dan Data Uji

Pada gambar 4.14 menentukan dan membagi dataset menjadi data latih dan data uji agar model dapat dilatih dan dievaluasi. Proporsi data yang digunakan untuk data uji, yaitu 40% dari total data sedangkan untuk data latih sebanyak 60% yang digunakan untuk melatih model. Adapun fungsi `random_state=42` digunakan untuk menentukan *seed* pada proses pengacakan agar pembagian data selalu konsisten dan dapat digunakan secara berulang-ulang.

```
# Normalisasi fitur
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

**Gambar 4. 15** Normalisasi Fitur

Gambar 4.15 merupakan tahap untuk menormalisasai fitur pada sistem klasifikasi *support vector machine* (SVM) dimana proses ini mengubah nilai-nilai fitur pada dataset sehingga fitur memiliki nilai yang seragam dan skala yang sama agar model lebih stabil dan tidak terlalu sensitif terhadap perubahan nilai fitur.

„StandardScaler“ adalah sebuah teknik normalisasi fitur yang mengubah nilai-nilai fitur menjadi nilai-nilai yang memiliki rata-rata 0 dan standar deviasi 1. Dengan demikian, kodingan di atas memastikan bahwa semua fitur

```
# Membuat model SVM
svm = SVC(kernel='linear', C=1.0, random_state=42)
```

**Gambar 4. 16** Model SVM

dalam „X\_train“ dan „X\_test“ memiliki skala yang sama, sehingga model pembelajaran mesin dapat bekerja lebih efektif.

Gambar 4.16 ditujukan untuk membuat model *support vector machine* (SVM) dengan parameter yang sudah ditentukan. „**svm = SVC**“ Membuat objek „**svc**“ yang merupakan implementasi dari algoritma SVM dalam library „**scikit-learn**“. „**C=1.0**“ Menentukan nilai parameter C, yang merupakan parameter regularization dalam model SVM. Nilai C yang lebih besar akan

membuat model SVM lebih kompleks dan lebih rentan terhadap overfitting dan seed generator bilangan acak yang telah ditentukan.

**b. Implementasi Klasifikasi Hasil Pertandingan dengan *Support Vector Machine***

Proses dalam metode Random Forest serupa dengan langkah-langkah yang dilakukan pada metode lain, di mana langkah pertama adalah menentukan fitur dan target yang relevan dan data dibagi menjadi data latih dan data uji untuk membangun dan menguji model. Dengan menggunakan data latih, model Random Forest dibangun dan dilatih untuk mengenali pola yang ada dalam data.

Setelah model selesai dibangun, langkah selanjutnya adalah melakukan pengujian baik pada data latih maupun data uji untuk mengevaluasi seberapa baik kinerja model tersebut. Pengujian ini bertujuan untuk memperoleh tingkat akurasi model yang telah dibuat, serta untuk melihat bagaimana model mampu menggeneralisasi data yang belum pernah dilihat sebelumnya. Proses evaluasi ini sangat penting untuk memastikan bahwa model yang dihasilkan dapat memberikan prediksi yang akurat dan dapat diandalkan.

Adapun tahapan nya yaitu dengan membuat fungsi yang digunakan untuk menghitung metrik evaluasi berdasarkan label sebenarnya. Dalam implementasi nya, juga akan digunakan beberapa kernel yaitu kernel *linear*, *radial*, dan *polynomial*.

**a. *Kernel Linear***

Fungsi yang digunakan dalam klasifikasi data yang dapat dipisahkan secara linier berperan untuk menghitung jarak atau titik antara dua vektor dalam ruang fitur. Proses ini memungkinkan Support Vector Machine (SVM) untuk mengidentifikasi dan menentukan hyperplane terbaik yang dapat memisahkan dua kelas data dengan margin terbesar. Dengan menggunakan fungsi ini, SVM dapat mengoptimalkan posisi hyperplane, sehingga memaksimalkan perbedaan antara dua kelas yang berbeda.

Hal ini sangat penting dalam meningkatkan akurasi klasifikasi, karena margin yang lebih besar antara kelas-kelas yang berbeda cenderung menghasilkan model yang lebih robust dan mampu membuat prediksi yang lebih akurat. Fungsi ini menjadi dasar dalam pendekatan SVM untuk data yang

terpisah secara linier, dengan tujuan untuk menciptakan pemisahan yang jelas dan efektif antara dua kategori data.

Pada penelitian yang dilakukan, untuk menentukan proses klasifikasi hasil pertandingan sepak bola, diperlukan adanya model yang ada pada kernel *linear*. Dapat dilihat pada gambar 4.16.

```
# Membuat model SVM
svm = SVC(kernel='linear', C=1.0, random_state=42)
```

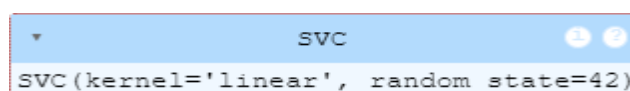
**Gambar 4. 17** Membuat model SVM

Gambar 4.17 menunjukkan proses untuk membuat model SVM yang akan dipakai pada klasifikasi. „**svm = SVC()**“ Membuat objek „**svc**“ yang merupakan implementasi dari algoritma SVM dalam library „**scikit-learn**“. „**kernel='linear'**“ untuk menentukan jenis kernel yang digunakan dalam model SVM. Pada model SVM akan mencari hyperplane yang memisahkan data dalam ruang fitur. „**C=1.0**“ Menentukan nilai parameter C, yang merupakan parameter regularization dalam model SVM. Nilai C yang lebih besar akan membuat model SVM lebih kompleks dan lebih rentan terhadap overfitting.

```
# Melatih model SVM
svm.fit(X_train, y_train)
```

**Gambar 4. 18** Melatih model SVM

Tahap pelatihan model SVM ditunjukkan pada gambar 4.18. Dengan menjalankan "svm.fit(X\_train, y\_train)", Anda meminta model SVM untuk belajar dari data pelatihan yang diberikan. Model akan mencari hyperplane yang ideal untuk membedakan kelas dalam data berdasarkan karakteristiknya. Model SVM akan siap untuk digunakan untuk membuat prediksi pada data baru setelah proses pelatihan selesai. Gambar 4.18 menunjukkan gambar model SVC kernel linear.



```
SVC
SVC(kernel='linear', random_state=42)
```

**Gambar 4. 19** Model SVC



Perhitungan nilai evaluasi untuk keseluruhan kelas klasifikasi hasil pertandingan sepak bola dapat dihitung berdasarkan informasi yang diberikan pada tabel 4.3.

**Tabel 4. 3** Hasil Akurasi, Precision, Recall, dan F1-Score

Kelas	Akurasi (TP + TN) / (TP + TN + FP + FN)	<i>Precision</i> TP / (TP + FP)	<i>Recall</i> TP / (TP + FN)	F1-Score 2 * (Presisi * Recall) / (Presisi + Recall)
Home	100%	100%	100%	100%
Away	100%	100%	100%	100%
Draw	100%	100%	100%	100%

Tabel 4.3 menunjukkan hasil metrik evaluasi dari proses klasifikasi hasil pertandingan sepak bola dengan variabel („A“, „D“, „H“) yaitu F1-score sebesar 100%, 100%, 100%, precision sebesar 100%, 100%, 100%, dan recall sebesar 100%, 100%, 100%.

#### b. Kernel Radial

Kernel Radial adalah jenis fungsi kernel yang digunakan untuk mengukur tingkat kesamaan antara dua titik data. Fungsi ini bekerja dengan cara menghitung jarak antara dua titik data dalam ruang fitur untuk menentukan seberapa mirip keduanya. Dengan pendekatan ini, kernel radial memungkinkan model untuk menangani data yang lebih kompleks, di mana pemisahan linier tidak dapat dilakukan secara langsung.

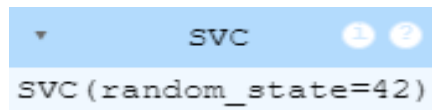
```
# Membuat model SVM
svm = SVC(kernel='rbf', C=1.0, random_state=42)
```

**Gambar 4. 20** Membuat model RBF

Pada gambar 4.20 model menggunakan **Radial Basis Function (RBF) kernel**, yang bekerja dengan menghitung kesamaan antara data berdasarkan jarak Euclidean dengan parameter regulasi (regularization parameter) dalam SVM. Dengan menetapkan seed yaitu `random_state = 42` untuk generator angka acak, memastikan hasil yang dapat direproduksi.

```
# Melatih model SVM
svm.fit(X_train, y_train)
```

**Gambar 4. 21** Melatih Model SVM



```
SVC
```

```
SVC(random_state=42)
```

**Gambar 4. 22** Model Kernel RBF

Tahap melatih model SVM ditunjukkan pada gambar 4.21 dan 4.22. Dengan menjalankan "svm.fit(X\_train, y\_train)", Anda meminta model SVM untuk belajar dari data pelatihan yang diberikan. Untuk memisahkan kelas-kelas dalam data berdasarkan fitur yang ada, model akan mencari hyperplane yang optimal. Setelah proses pelatihan selesai, model akan siap untuk digunakan untuk membuat prediksi pada data baru.

Perhitungan akurasi, *precision*, *recall*, dan *F1-score* untuk keseluruhan kelas klasifikasi hasil pertandingan sepak bola dapat dihitung berdasarkan informasi yang diberikan pada tabel 4.3.

**Tabel 4. 4** Hasil Akurasi, Precision, Recall, dan F1-Score

Kelas	Akurasi (TP + TN) / (TP + TN + FP + FN)	<i>Precision</i> TP / (TP + FP)	<i>Recall</i> TP / (TP + FN)	<i>F1-Score</i> 2 * (Presisi * Recall) / (Presisi + Recall)
Home	88%	88%	96%	92%
Away	88%	100%	84%	92%
Draw	88%	70%	70%	70%

Tabel 4.4 menunjukkan hasil metrik evaluasi dari proses klasifikasi hasil pertandingan sepak bola dengan variabel („A“, „D“, „H“) yaitu F1-score sebesar 98%, 95%, 89%, precision sebesar 99%, 100%, 83%, dan recall sebesar 97%, 91%, 97%.

### c. Kernel *Polynomial*

Kernel *Polynomial* memetakan data ke dalam ruang berdimensi tinggi menggunakan transformasi berbasis polinomial, sehingga memungkinkan pemisahan data yang tidak linier di ruang fitur aslinya. Dibandingkan dengan kernel RBF, kernel *polynomial* lebih terbatas pada jenis pola tertentu yang bisa dimodelkan.

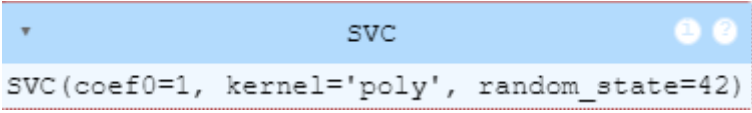
```
# Membuat model SVM
svm = SVC(kernel='poly', degree=3, coef0=1, C=1.0, random_state=42)
```

**Gambar 4. 23** Membuat Model *Poly*

Gambar 4.23 menggunakan kernel *polynomial*, yang memetakan data ke dalam ruang berdimensi tinggi dengan transformasi berbasis polinomial. Pada kodingan yang tertera, kernel *polynomial* akan membentuk hubungan kubik.

```
# Melatih model SVM
svm.fit(X_train, y_train)
```

**Gambar 4. 24** Melatih Model *Poly*



**Gambar 4. 25** Model Kernel *Poly*

Gambar 4.24 dan 4.25 merupakan tahap untuk melatih model SVM. Dengan menjalankan „svm.fit(X\_train, y\_train)”, Anda memberi tahu model SVM untuk belajar dari data pelatihan yang diberikan. Model akan mencari hyperplane yang optimal untuk memisahkan kelas dalam data berdasarkan fitur-fitur yang ada. Setelah proses pelatihan selesai, model akan siap untuk digunakan dalam membuat prediksi data baru.

Perhitungan akurasi, *precision*, *recall*, dan *F1-score* untuk keseluruhan kelas klasifikasi hasil pertandingan sepak bola dapat dihitung berdasarkan informasi yang diberikan pada tabel 4.4.

**Tabel 4. 5** Hasil Akurasi, Precision, Recall, dan F1-Score

Kelas	Akurasi $(TP + TN) / (TP + TN + FP + FN)$	Precision $TP / (TP + FP)$	Recall $TP / (TP + FN)$	F1-Score $2 * (Presisi * Recall) / (Presisi + Recall)$
Home	97%	100%	96%	98%
Away	97%	98%	98%	98%
Draw	97%	88%	97%	92%

Tabel 4.5 menunjukkan hasil metrik evaluasi dari proses klasifikasi hasil pertandingan sepak bola dengan variabel („H“, „A“, „D“) yaitu F1-score sebesar 98%, 98%, 92%, precision sebesar 100%, 98%, 88%, dan recall sebesar 96%, 98%, 97%.

#### 4.4 Perbandingan Hasil Setiap Metode Klasifikasi

Dalam tahap ini, hasil dari Algoritma *Random Forest* dan Algoritma *Support Vector Machine* dibandingkan. Tujuan dari perbandingan ini adalah untuk mengevaluasi kinerja masing-masing algoritma dengan menggunakan berbagai metric evaluasi. Tujuan lainnya adalah untuk mengetahui kelebihan dan kelemahan kedua pendekatan dalam menangani masalah klasifikasi.

Selain itu, analisis ini juga memperhitungkan bagaimana setiap metode beradaptasi dengan karakteristik data, seperti adanya ketidakseimbangan kelas, jumlah fitur yang digunakan, dan tingkat kompleksitas pola yang terdapat dalam dataset. Evaluasi ini menggunakan data uji yang representatif untuk melihat seberapa baik algoritme mampu mengidentifikasi pola di kelas mayoritas dan minoritas, dan bagaimana kinerjanya ketika diterapkan pada data yang belum pernah ada sebelumnya.

Hasil perbandingan ini tidak hanya memberikan informasi mengenai tingkat efisiensi dan efektivitas masing-masing metode, namun juga memberikan dasar untuk menentukan pendekatan klasifikasi yang optimal untuk digunakan dalam penelitian ini. Pendekatan seperti oversampling dan penetapan parameter model juga dipertimbangkan untuk memastikan

perbandingan benar-benar obyektif dan relevan dengan tujuan penelitian. Untuk perbandingan akurasi model metode klasifikasi dapat dilihat pada tabel 4.5.

**Tabel 4. 6** Perbandingan Akurasi Model Klasifikasi

Metode	Data			
	Hasil Pertandingan	<i>Sensitivity</i>	<i>Precision</i>	Total Akurasi
<i>Random Forest</i>	<i>Home</i>	93,33%	100%	<b>94%</b>
	<i>Away</i>	97,40%	96,15%	
	<i>Draw</i>	90%	84,37%	
<i>Support Vector Machine</i>	Hasil Pertandingan	<i>Sensitivity</i>	<i>Precision</i>	Total Akurasi
<i>Linear</i>	<i>Home</i>	100%	100%	<b>100%</b>
	<i>Away</i>	100%	100%	
	<i>Draw</i>	100%	100%	
<i>Radial</i>	<i>Home</i>	97%	99%	<b>88%</b>
	<i>Away</i>	91%	100%	
	<i>Draw</i>	97%	83%	
<i>Polynomial</i>	<i>Home</i>	100%	100%	<b>97%</b>
	<i>Away</i>	100%	100%	
	<i>Draw</i>	100%	100%	

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Hasil pengujian dan implementasi yang dilakukan untuk membandingkan kinerja Algoritma *Random Forest* dan *Support Vector Machine* (SVM) dalam mengklasifikasikan hasil pertandingan sepak bola, dapat disimpulkan sebagai berikut.

1. Dari hasil pengujian sistem, dengan metode evaluasi *confusion matrix* pada Algoritma *Random Forest* didapatkan nilai total akurasi sebesar 94%. Sedangkan pada Algoritma *Support Vector Machine* menggunakan beberapa kernel yaitu kernel *Linear*, *Radial*, dan *Polynomial* didapatkan nilai total akurasi masing-masing sebesar 100%, 88%, 97%. Maka dengan demikian, untuk penggunaan Algoritma yang tepat didapatkan Algoritma *Support Vector Machine*.
2. Metode yang paling tepat dalam pengklasifikasian untuk perbandingan algoritma yaitu Algoritma *Support Vector Machine* dengan menggunakan kernel fungsi *linear* dan mendapatkan tingkat akurasi sebesar 100%..

#### 5.2 Saran

Saran yang dapat diberikan untuk menjadi pertimbangan dalam pengembangan dan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian ini memiliki luaran berupa hasil klasifikasi hasil pertandingan sepak bola Liga Utama Inggris yang menggunakan Algoritma *Random Forest* dan Algoritma *Support Vector Machine* (SVM), peneliti selanjutnya diharapkan dapat menggunakan algoritma perbandingan yang sama untuk kasus yang berbeda.
2. Penelitian selanjutnya diharapkan dapat memperbanyak kuantitas variabel yang mungkin berpengaruh dalam tingkat kemenangan suatu tim dalam memenangkan pertandingan.
3. Penelitian selanjutnya diharapkan memperbanyak kuantitas dataset sehingga akurasi dari sistem klasifikasi bisa lebih baik lagi.

## DAFTAR PUSTAKA

- Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, 1142(1). <https://doi.org/10.1088/1742-6596/1142/1/012012>
- Azhari, M., Situmorang, Z., & Rosnelly, R. (2021). Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(2), 640. <https://doi.org/10.30865/mib.v5i2.2937>
- Chowdhury, M. S. (2024). Comparison of accuracy and reliability of random forest, support vector machine, artificial neural network and maximum likelihood method in land use/cover classification of urban setting. *Environmental Challenges*, 14. <https://doi.org/10.1016/j.envc.2023.100800>
- Devikanniga, D., Ramu, A., & Haldorai, A. (2020). Efficient diagnosis of liver disease using support vector machine optimized with crows search algorithm. *EAI Endorsed Transactions on Energy Web*, 7(29). <https://doi.org/10.4108/EAI.13-7-2018.164177>
- Ge, Z., Song, Z., Ding, S. X., & Huang, B. (2017). Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*, 5, 20590–20616. <https://doi.org/10.1109/ACCESS.2017.2756872>
- Guo, X., & Hao, P. (2021). Using a random forest model to predict the location of potential damage on asphalt pavement. *Applied Sciences (Switzerland)*, 11(21). <https://doi.org/10.3390/app112110396>
- Hokijuliandy, E., Napitupulu, H., & Firdaniza, F. (n.d.). *Analisis Sentimen Menggunakan Metode Klasifikasi Support Vector Machine (SVM) dan Seleksi Fitur Chi-Square*.
- Huang, S., Nianguang, C. A. I., Penzuti Pacheco, P., Narandes, S., Wang, Y., & Wayne, X. U. (2018). Applications of support vector machine (SVM) learning in cancer genomics. In *Cancer Genomics and Proteomics* (Vol. 15, Issue 1, pp. 41–51). International Institute of Anticancer Research. <https://doi.org/10.21873/cgp.20063>
- J, I. A. (2020). Fuzzy-Based Model for Predicting Football Match Results.

- Mahmudur Rahman Khan, M., Bente Arif, R., Abu Bakr Siddique, M., & Rahman Oishe, M. (n.d.). *Study and Observation of the Variation of Accuracies of KNN, SVM, LMNN, ENN Algorithms on Eleven Different Datasets from UCI Machine Learning Repository*.
- Mantovani, R. G., Rossi, A. L. D., Alcobaça, E., Vanschoren, J., & de Carvalho, A. C. P. L. F. (2019). A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences*, 501, 193–221. <https://doi.org/10.1016/j.ins.2019.06.005>
- Mentch, L., & Zhou, S. (2019). *Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success*. <http://arxiv.org/abs/1911.00190>
- Razali, N., Mustapha, A., Yatim, F. A., & Ab Aziz, R. (2017). Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL). *IOP Conference Series: Materials Science and Engineering*, 226(1). <https://doi.org/10.1088/1757-899X/226/1/012099>
- Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *Stata Journal*, 20(1), 3–29. <https://doi.org/10.1177/1536867X20909688>
- Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (Vol. 13, pp. 6308–6325). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/JSTARS.2020.3026724>
- Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. In *Expert Systems with Applications* (Vol. 134, pp. 93–101). Elsevier Ltd. <https://doi.org/10.1016/j.eswa.2019.05.028>
- Wu, W. T., Li, Y. J., Feng, A. Z., Li, L., Huang, T., Xu, A. D., & Lyu, J.



(2021).

Data mining in clinical big data: the frequently used databases, steps, and methodological models. In *Military Medical Research* (Vol. 8, Issue 1). BioMed Central Ltd. <https://doi.org/10.1186/s40779-021-00338-z>

Zhang, X., Shen, H., Huang, T., Wu, Y., Guo, B., Liu, Z., Luo, H., Tang, J., Zhou, H., Wang, L., Xu, W., & Ou, G. (2024). Improved random forest algorithms for increasing the accuracy of forest aboveground biomass estimation using Sentinel-2 imagery. *Ecological Indicators*, 159. <https://doi.org/10.1016/j.ecolind.2024.111752>