PREDIKSI HASIL PERTANDINGAN SEPAKBOLA *ENGLISH PREMIER LEAGUE* DENGAN MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS* DAN *NAÏVE BAYES CLASSIFIER*

SKRIPSI

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1

Disusun oleh:

Darmastyo Bagas Prabowo

15524002

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2020

LEMBAR PENGESAHAN

PREDIKSI HASIL PERTANDINGAN SEPAKBOLA *ENGLISH PREMIER LEAGUE* DENGAN MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS* DAN *NAÏVE BAYES CLASSIFIER*

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Darmastyo Bagas Prabowo
15524002

Yogyakarta, 07 Desember 2020

Menyetujui,

Pembimbing Skripsi

Dzata Farahiyah, S.T., M.Sc.
155220509

LEMBAR PENGESAHAN

SKRIPSI

PREDIKSI HASIL PERTANDINGAN SEPAKBOLA *ENGLISH PREMIER LEAGUE* DENGAN MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBORS* DAN *NAÏVE BAYES CLASSIFIER*

Dipersiapkan dan disusun oleh:

Darmastyo Bagas Prabowo

15524002

Telah dipertahankan di depan dewan penguji Pada tanggal: 07 Desember 2020

Susunan dewan penguji

Ketua Penguji : Dzata Farahiyah,S.T.,M.Sc., _____

Anggota Penguji 1: Elvira Sukma Wahyuni,S.Pd.,M.Eng., _____

Anggota Penguji 2: Firdaus,S.T.,M.T.,Ph.D., _____

Skripsi ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana

Tanggal: 07 Desember 2020

Ketua Program Studi Teknik Elektro

Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 07 Desember 2020

Darmastyo Bagas Prabowo

KATA PENGANTAR

Assalamu 'alaikum Wr. Wb

Segala puji dan syukur penulis panjatkan pada Kehadirat Allah SWT yang senantiasa melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul " *PREDIKSI HASIL PERTANDINGAN SEPAKBOLA ENGLISH PREMIER LEAGUE DENGAN MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBORS DAN NAÏVE BAYES CLASSIFIER* " ini dengan baik dan lancar.

Skripsi ini wajib ditempuh oleh mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia sebagai salah satu syarat untuk menyelesaikan jenjang studi Strata Kelancaran dalam mempersiapkan dan menyelesaikan Skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis haturkan kepada:

1. Bapak Dardiri dan Ibu Eko Supriyanti, selaku orangtua penulis yang tidak henti memberikan do'a, motivasi, dan kasih sayang yang tulus Kepada Penulis. Serta Kakak penulis yaitu Lina Arum Sari, beserta keponakan penulis Muhammad Fadhil Gadaffi dan Muhammad Aulian Attafaris yang menjadi motivasi penulis untuk segera menyelesaikan tugas akhir ini.
2. Ibu Dzata Farahiyah, S.T., M.Sc., selaku Dosen Pembimbing Skripsi yang tidak pernah lelah memberikan bimbingan dan saran kepada penulis hingga selesainya skripsi ini.
3. Bapak Yusuf Aziz Amrullah, S.T., M.Eng., Ph.D., selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia.
4. Seluruh Dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama menempuh kuliah dari semester pertama hingga akhir di Jurusan Teknik Elektro.
5. Isna Fitria Abdillah yang menjadi motivasi penulis untuk segera menyelesaikan skripsi ini.
6. Almer Apparel beserta team yang membuat penulis sanggup menyelesaikan skripsi ini. Terimakasih pelajaran dan pengalaman selama ini.
7. Class of 2015 yang selalu ada untuk penulis. Khususnya Rozan Andru dan Fata Attamami yang memotivasi dan mendukung penulis untuk menyelesaikan skripsi ini
8. Teman-teman seperjuangan yang telah membantu saya dalam mengerjakan skripsi yaitu Fikri Prayoga, M. ikhsan, dan Genta Bayu.
9. Teman-teman Teknik Elektro Angkatan 2015.

10. Semua pihak yang telah memberikan masukan, dorongan dan semangat dalam menyelesaikan skripsi ini.

Penulis menyadari akan kekurangan dari skripsi yang jauh dari kata sempurna ini, baik dari segi materi maupun segi penulisan. Segala saran dan kritik yang disampaikan akan sangat bermanfaat bagi penulis untuk membuat karya tulis berikutnya dengan hasil yang lebih baik. semoga skripsi ini bermanfaat bagi yang membacanya.

Wassalamu 'alaikum warahmatullahi wabarakatuh.

ARTI LAMBANG DAN SINGKATAN

KNN	: *K-Nearest Neighbors*
NBC	: Naïve Bayes Classifier
EPL	: *English Premiere League*
ML	: *Machine Learning*
HS	: *Home Shot*
AS	: *Away Shot*
HST	: *Home Shot on Target*
AST	: *Away Shot on Target*
HTAG	: *Half Time Away Goal*
HTHG	: *Half Time Home Goal*
HF	: *Home Foul*
AF	: *Away Foul*
HC	: *Home Corner*
AC	: *Away Corner*
HY	: *Home Yellow Card*
AY	: *Away Yellow Card*
HR	: *Home Red Card*
AR	: *Away Red Card*
AI	: *Artificial Intelligence*
FTR	: *Fulltime Result*
P	: *Positive*
N	: *Negative*
TN	: *True Negative*
TP	: *True Positive*
FP	: *False Positive*
FN	: *False Negative*

ABSTRAK

Sepak bola adalah salah satu olahraga yang sangat popular di dunia, bahkan sepak bola mampu mencakup banyak aspek seperti kesehatan, hiburan, mata pencaharian, maupun dunia bisnis. Salah satu kompetisi sepak bola terbesar adalah *English Premier League*. Skripsi ini bertujuan untuk mengetahui prediksi hasil pertandingan sepak bola *English Premier League* berupa Home Win (H), Away Win (A), dan Draw (D) meggunakan metode *Machine learning* dengan membandingkan Algoritma *K-Nearest Neighbors* dan *Naïve Bayes Classifier*. KNN digunakan untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel dari training data. Sedangkan *Naive bayes* digunakan karena hanya membutuhkan jumlah data latih yang kecil untuk menentukan estimasi parameter yang diperlukan pada proses pengklasifikasian. Pada penelitian ini menggunakan 3 skenario pembagian data latih dan data uji yaitu skenario 1 (75%-25%), skenario 2 (80%-20), dan skenario 3 (90%-10%). Pada keseluruhan skenario menghasilkan nilai akurasi NB1 60,5%, NB2 60,5%, NB3 63,5%. Kemudian pada pengujian KNN1 60,3%, KNN2 59,6%, KNN3 62,7%. Pada penelitian ini algoritma *Naïve Bayes Classifier* menghasilkan nilai akurasi yang lebih baik secara keseluruhan dibanding dengan model *K-Nearest Neighbors.*

Kata Kunci: *Machine Learning, Naïve Bayes Classifier, K-Nearest Neighbors.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Dewasa ini sepak bola bukan hanya menjadi sebuah olahraga, tetapi sepak bola menjadi sebuah industri yang mencakup banyak aspek, seperti bisnis, hiburan, dan perkembangan teknologi serta pengetahuan sepak bola. Salah satu faktor penting dalam dunia sepak bola adalah data statistik. Melalui pendekatan ilmu statistik para pelatih klub sepak bola dunia dapat melakukan Analisa tindakan yang perlu agar bisa meraih kemenangan.

Statistik adalah sebuah aplikasi (pengembangan) ilmiah dari prinsip matematika untuk mengolah data input, analisis, dan penyajian data. Dalam machine learning Teknik statistik digunakan dalam proses data mining [1]. Dalam dunia sepak bola data mining dilakukan dengan menganalisa jalannya laga, mulai dari jumlah shoot on target, maupun off target hingga statistik umum suatu pertandingan sepak bola.

Machine Learning (ML) atau pembelajaran mesin merupakan metode pendekatan dalam *Artificial Intelligence* (AI) yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah. Kegunaan utama dalam *Machine Learning* (ML) adalah untuk klasifikasi dan prediksi suatu objek. *Machine Learning* (ML) memiliki proses pelatihan, pembelajaran, atau training [2]. Dalam ML, algoritma yang digunakan untuk menghitung dan menganalisa suatu data statistik sesuai dengan teorema dasarnya adalah *Naïve Bayes Classifier* dan *K-Nearest Neighbors* untuk proses klasifikasi suatu label berdasarkan data training. Penelitian ini akan menggunakan metode Algoritma *Naïve Bayes* dan KNN untuk melakukan prediksi hasil pertandingan sepakbola EPL dengan dataset dan fitur yang didapatkan dari statistik pertandingan EPL. Hasil pertandingan yang dimaksud adalah Home Win (H), Away Win (A), dan Draw (D). kedua algoritma tersebut dipilih karena dataset berbentuk statistik dan data yang digunakan sangat banyak.

Naïve Bayes Classifier adalah sebuah metode klasifikasi yang berakar dari teorema bayes. Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari Naïve Bayes Classifier ini adalah asumsi yang sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian [3]. *Naïve Bayes Classifier* dipilih karena metode pengklasifikasiannya menggunakan metode probabilitas dan statistik.

K-Nearest Neighbors adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pemelajaran yang jaraknya paling dekat dengan objek tersebut. Data pemelajaran digambarkan ke ruang berdimensi banyak dengan tiap-tiap dimensi mewakili tiap ciri/fitur dari data. Klasifikasi data baru dilakukan dengan mencari label k tetangga terdekat. Label terbanyak yang muncul menjadi label data baru. Bila k = 1, data baru dilabeli dengan label tetangga terdekat [4]. Penggunaan metode *K-Nearest Neighbors* dikarenakan metode ini memiliki pelatihan yang sangat cepat, sederhana dan mudah dipelajari, dan efektif meskipun data pelatihannya besar, sehingga metode *K-Nearest Neighbors* dipilih pada penelitian ini.

Pada penelitian ini metode *Naïve Bayes Classifier* dan *K-Nearest Neighbors* digunakan untuk memprediksi hasil pertandingan *English Premier League* (EPL), karena EPL merupakan kompetisi sepak bola dunia yang memiliki banyak penggemar diseluruh dunia dan merupakan liga sepak bola yang paling kompetitif. Kemudian kedua algotitma tersebut dibandingkan untuk menentukan akurasi algoritma yang lebih baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan sebelumnya, maka didapatkan beberapa rumusan masalah, diantaranya :

1. Bagaimana kinerja dari model prediksi hasil pertandingan EPL dengan menggunakan algoritma *Naïve bayes Classifier* ?
2. Bagaimana kinerja dari model prediksi hasil pertandingan EPL dengan menggunakan algorima *K-Nearest Neighbors* ?

1.3 Batasan Masalah

Batasan masalah berisi hal-hal yang membatasi lingkup penelitian.

1. Dataset pada penelitian ini menggunakan stastistik hasil pertandingan EPL selama 6 musim (2014/2015 – 2019/2020).
2. Dataset didapatkan dari www.football-data.co.uk .
3. Atribut yang digunakan untuk prediksi adalah *Home Team, Away Team, Home Shot* (HS), *Away Shot* (AS), *Home Shot on Target* (HST), *Away Shot on Target* (AST), *Home Foul* (HF), *Away Foul* (AF), *Half Time Away Goal* (HTAG), *Half Time Home Goal* (HTHG) *Home Corner* (HC), *Away Corner* (AC), *Home Yellow Card* (HY), *Away Yellow Card* (AY), *Home Red Card* (HR), *Away Red Card* (AR).
4. Pembagian data latih dan data uji dilakukan secara acak menggunakan bantuan *Scikit-*

Learn Library.

5. Perhitungan komputasi algorima *Naïve Bayes Classifier* dan *K-nearest Neighbors* menggunakan bantuan *Scikit-Learn Library*.
6. Rasio skenario pembagian data latih dan data uji yang digunakan, yaitu 75:25,80:20, dan 90:10.
7. Parameter K yang ditetapkan bernilai 17 (K=17), didapatkan dengan bantuan *GridSearchCV* untuk menentukan nilai K terbaik
8. Parameter untuk mengevaluasi kinerja model adalah akurasi, presisi, *recall,* dan fscore.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini dibagi menjadi beberapa hal, yaitu :

1. Untuk mengetahui kinerja dari model prediksi hasil pertandingan EPL dengan menggunakan algoritma *Naïve Bayes Classifier*.
2. Untuk mengetahui kinerja dari model prediksi hasil pertandingan EPL dengan menggunakan algoritma *K-Nearest Neighbors*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Untuk mengetahui algoritma terbaik antara *Naïve Bayes Classifier* dengan *K-nearest Neighbors* dalam melakukan prediksi hasil pertandingan sepak bola.
2. Membantu jajaran pelatih untuk melakukan prediksi hasil pertandingan yang berguna untuk menyiapkan latihan atau strategi guna mencapai hasil maksimal dalam pertandingan sepak bola.
3. Menjadi acuan untuk memprediksi hasil pertandingan sepakbola bagi penggemar sepak bola secara efektif.

BAB 2
TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian yang dilakukan oleh Hijmans, A., 2016 [5], tentang prediksi timnas sepakbola Belanda setelah gagal dalam kualifikasi *Euro Cup* 2016, sehingga penelitian ini diharapkan dapat membantu pembina timnas sepak bola Belanda untuk mengambil keputusan kedepan. Pada penelitian ini metode yang digunakan adalah GBM (*Gradient Boosting Machine), K-nearest Neighbors, dan Naïve Bayes Classifier.* hasil dari penelitian ini adalah GBM melakukan prediksi lebih baik dari *Naïve Bayes* dan *K-nearest Neighbors.* GBM mendapatkan prediksi rata-rata 60,22 % sedangkan Naïve bayes 42% dan K-nearest neighbors 58,62 %. Hasil ini mengacu pada dataset yang hanya mencakup informasi tentang timnas sepakbola Belanda dan tidak ada informasi tentang lawan kecuali hanya peringkat FIFA. Dari variable *dataset* penelitian tersebut, variable taktis tidak banyak memiliki nilai prediktif sedangkan selisih ranking FIFA memiliki nilai predikitif paling banyak.

Kemudian menurut Razali, N. *et al .,* 2017 [6], dalam penelitiannya memaparkan bahwa prediksi hasil pertandingan sepakbola telah menarik begitu banyak orang yang memiliki passion pada sepakbola, dari managerial tim sepakbola hingga para penggemarnya sendiri. Penelitian ini menjadi menarik, antara lain karena kesulitannya yang disebabkan oleh banyak faktor yang mempengaruhi hasil pertandingan sepakbola, seperti kerjasama tim, skill, cuaca, keunggulan tuan rumah dan banyak lainnya. Penelitian yang dilakukan adalah dengan membandingkan hasil prediksi pertandingan *English premier league* (EPL) dalam tiga musim, yaitu musim 2010/2011, 2011/2012, 2012/2013. Peneliti menggunakan algoritma *Bayesian Networks* (BNs), kemudian ada 3 label class yang dipilih yaitu Kemenangan Kandang (H), kemenangan tandang (A), dan Seri (D). pada penelitiannya *K-fold Cross Validation* digunakan untuk menguji keakuratan model prediksi. Atribut yang dipilih oleh peneliti dari *dataset* adalah *Home Team, Away Team, Home Team Shot, Away Team Shot, Home Team Shot on Target, Away Team Shot on Target, Home Team Corner, Away Team Corner, Home Team Foul, Away Team Foul, Home Team Yellow Card, Away Team Yellow Card, Home Team Red Card, Away Team Red Card, Half Time Home Goals, Half Time Away Goals, Full Time Home Goals, Full Time Away Goals.* Kemudian dari penelitian, didapatkan akurasi prediksi untuk musim 2010/2011 75,26%, musim 2011/2012 79,47%, dan musim 2012/2013 70,53%.

Kemudian Alfredo, Yoel F. dan Isa, Sani M., 2019 [7], melakukan penelitian tentang prediksi hasil pertandingan EPL selama 10 musim (2007/2008 - 2016/2017) dengan menggunakan algoritma berbasis pohon yaitu *C5.0, Random Forest,* dan *Extreme Gradient Boosting.* Penelitimengatakan

bahwa fitur terbaik untuk mengoptimalkan akurasi model prediksi hasil sepakbola adalah fitur *Full Time, Half Time Home Goal, Half Time Away Goal, Home Shot, Away Shot, Home Shot Target, Away Shot Target, Home fouls, Away Fouls, Home Corner, Away Corner, Home Yellow Card, Away Yellow Card, Home Red Card, Away Red Card.*

Pada penelitian tersebut ditentukan pembagian data latih dan data uji sebanyak 80%-20%, Kemudian ditentukan juga nilai *K-fold Cross Validation* dengan nilai variable K = 10. Hasil dari penelitian tersebut menunjukan *C5.0, Random Forest,* dan *Extreme Gradient Boosting* menghasilkan nilai akurasi 64,87%, 68,55%, dan 67,89%.

2.2 Tinjauan Teori

2.2.1 English Premier League (EPL)

English Premier League (EPL) adalah liga tertinggi dalam sistem liga sepakbola di Inggris dan kompetisi yang paling kompetitif di dunia. Kompetisi ini diikuti oleh 20 klub tiap musimnya dan menerapkan sistem promosi dan degradasi dari kasta liga dibawahnya yaitu *Championship English Football League* [8]. Faktor yang membuat liga Inggris menjadi sangat kompetitif adalah efek industrial sepak bola sejak memakai konsep *Premier League* yang diresmikan mulai tahun 1992. Efek industrialisasi paling nyata adalah hak siar liga yang sangat mahal dan berkontribusi besar pada pemasukan semua klub EPL.

Alasan lain EPL dinobatkan sebagai liga paling kompetitif di dunia adalah tidak adanya klub yang selalu mendominasi liga selama 10 tahun atau lebih seperti *La Liga* (liga sepakbola Spanyol) dengan dominasi antara klub *Barcelona* dan *Real Madrid, Seria A* (liga sepakbola Italia) dengan dominasi *Juventus, Bundesliga* (liga sepakbola Jerman) dengan dominasi *Bayern Munchen.* Terbukti dengan klub yang menjuarai EPL selama 10 tahun terakhir adalah 5 klub yang berbeda. Dengan alasan tersebut, penelitian ini menjadi lebih menarik.

2.2.2 Machine Learning

Machine Learning (ML) adalah sebuah aplikasi dari *Artificial Intelligence* (AI) atau kecerdasan buatan yang berfokus pada pengembangan sebuah system yang mampu belajar "sendiri" tanpa harus berulang kali di program oleh manusia [3]. Ada dua aplikasi utama dalam ML yaitu klasifikasi dan prediksi. Aplikasi ML membutuhkan data sebagai bahan belajar atau disebut juga dengan data latih (*train set)* untuk melatih model *Machine Learning,* kemudian untuk menguji suatu model *Machine Learning* yang telah dilatih, digunakan sebuah dataset baru yang disebut dengan data uji atau *test set*. Klasifikasi adalah metode dalam ML oleh mesin untuk memilah objek berdasarkan ciri tertentu atau

membedakan benda satu dengan yang lain. Sedangkan prediksi digunakan oleh mesin untuk menentukan keluaran dari suatu data latih yang telah dimasukan sebelumnya.

2.2.3 *Naïve Bayes Classifier* (NBC)

Naïve Bayes adalah metode klasifikasi berdasarkan teori probabilitas dan teorema *Bayesian* dengan anggapan bahwa setiap variable penentu keputusan bersifat bebas (*independence*) sehingga keberadaan setiap variable tidak ada kaitannya dengan keberadaan atribut yang lain. Teorema *Bayes* memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya. Teorema tersebut dikombinasikan dengan Naïve dimana diasumsikan kondisi antar atribut saling bebas [7]. Teorema *Bayes* ditunjukkan oleh persamaan (2.1)

$$P(A|B) = \frac{(P(B|A) \times P(A))}{P(B)} \tag{2.1}$$

Keterangan :

A : Sampel data yang label kelasnya tidak diketahui

B : fitur untuk menentukan label kelas

P(A|B) : Probabilitas label A berdasarkan fitur B (*posterior probability)*

P(B|A) : Probabilitas fitur B berdasarkan label A (*likelihood)*

P(A) : Probabilitas label A (*prior probability)*

P(B) : Probabilitas fitur B (*evidence*)

Penentuan hasil prediksi label untuk suatu sampel data dilakukan dengan membandingkan *Posterior Probability* masing-masing label, kemudian label yang memiliki *Posterior Probability* paling tinggi akan dijadikann hasil prediksi. Pada penelitian ini ada 3 label kelas "FTR" yang digunakan, yaitu "H" atau kemenangan tim kandang, "A" atau kemenangan tim tandang, dan "D" atau seri. Adapun untuk fitur yang digunakan pada penelitian ini, penulis menggunakan hasil statistik pertandingan yang umum dalam suatu pertandingan karena dari statistik dapat menggambarkan apa yang terjadi dalam suatu pertandingan dan statistik menjadi tolak ukur objektif dalam menggambarkan performa suatu tim. Fitur yang dipilih yaitu *Home Team, Away Team, Half Time Home Goal, Half Time Away Goal, Home Shot , Away Shot , Home Shot on Target , Away Shot on Target , Home Foul, Away Foul, Home Corner, Away Corner , Home Yellow Card, Away Yellow Card , Home Red Card, Away Red Card,* Penerapan persamaan (2.1) dengan menggunakan fitur dan label yang digunakan pada penelitian, menghasilkan persamaan (2.2), (2.3), dan (2.4) untuk menghitung masing-masing *Posterior Probability* label :

$$P(H|X_1,X_2,\ldots\ldots,X_{16}) = \frac{P(X_1,X_2,\ldots\ldots,X_{16}|H)\times P(H)}{P(X_1,X_2,\ldots\ldots,X_{16})} \quad (2.2)$$

$$P(A|X_1,X_2,\ldots\ldots,X_{16}) = \frac{P(X_1,X_2,\ldots\ldots,X_{16}|A)\times P(A)}{P(X_1,X_2,\ldots\ldots,X_{16})} \quad (2.3)$$

$$P(D|X_1,X_2,\ldots\ldots,X_{16}) = \frac{P(X_1,X_2,\ldots\ldots,X_{16}|D)\times P(D)}{P(X_1,X_2,\ldots\ldots,X_{16})} \quad (2.4)$$

Keterangan :

X_1 : *Hometeam*

X_2 : *Awayteam*

X_3 : *Half Time Home Goals* (HTHG)

X_4 : *Half Time Away Goals* (HTAG)

X_5 : *Home Shot* (HS)

X_6 : *Away Shot* (AS)

X_7 : *Home Shot on Target* (HST)

X_8 : *Away Shot on Target* (AST)

X_9 : *Home Foul* (HF)

X_{10} : *Away Foul* (AF)

X_{11} : *Home Corner* (HC)

X_{12} : *Away Corner* (AC)

X_{13} : *Home Yellow* (HY)

X_{14} : *Away Yellow* (AY)

X_{15} : *Home Red* (HR)

X_{16} : *Away Red* (AR)

Pada penelitian ini, perhitungan *Posterior Probability* dilakukan dengan menggunakan bantuan *Scikit-Learn Library.* Pada *Scikit-Learn Library* memiliki algoritma komputasi yang mempermudah melakukan perhitungan pada jumlah data yang banyak.

2.2.4 *K-Nearest Neighbors* (KNN)

K-Nearest Neighbors (KNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data latih yang jaraknya paling dekat dengan objek tersebut. Algoritma ini juga merupakan salah satu teknik *Lazy Learning* karena tidak mempelajari cara mengkategorikan data, melainkan hanya megingat data yang sudah ada [8].

Pada KNN, parameter K merupakan jumlah dari tetangga terdekat. Algoritma ini menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru. Pada saat diberikan sampel uji, maka algoritma akan menemukan sejumlah K objek (titik latih) yang paling dekat dengan titik uji. Pada penelitian ini, nilai K dan metode pengukuran jarak didapatkan dengan bantuan *GridSearchCV* yang berguna untuk menentukan nilai K dan metode pengukuran jarak terbaik yaitu K = 17 dan jarak *Manhattan.* Jarak *Manhattan* digunakan untuk menghitung perbedaan absolut antara koordinat sepasang objek. Penggunaan *Manhattan Distance* untuk pengukuran jarak memperoleh akurasi yang tinggi dibandingkan dengan *Euclidean Distance* sehingga dalam penelitian ini menggunakan klasifikasi metode *K-Nearest Neighbors* dengan menggunakan *Manhattan Distance*. untuk menghitung jarak terdekat dalam metode klasifikasi Perhitungan jarak pada penelitian ini dirumuskan oleh persamaan (2.5).

$$d(x,y) = \sum_{i=1}^{n} |X_i - y_i| \tag{2.5}$$

Dimana :

$d(x, y)$: jarak

x_i : sampel data latih

y_i : sampel data uji

i : variable data

n : dimensi data

Berdasarkan fitur yang digunakan pada penelitian ini, maka perhitungan jarak antara sampel data latih dan data uji diperlihatkan dalam persamaan (2.6)

$$d(x_i, y_i) = |A_{x_i} - A_{y_i}| + |B_{x_i} - B_{y_i}| + \ldots\ldots + |Q_{x_i} - Q_{y_i}| \quad (2.6)$$

Dimana :

$d(x_i, y_i)$: jarak antara suatu data uji dan data latih

x_i : Data latih

y_i : Data uji

A : Home team

B : Away team

C : *Half Time Home Goals* (HTHG)

E : *Home Time Away Goals* (HTAG)

F : *Home Shot* (HS)

G : *Away Shot* (AS)

H : *Home Shot on Target* (HST)

I : *Away Shot on Target* (AST)

J : *Home Foul* (HF)

K : *Away Foul* (AF)

L : *Home Corner* (HC)

M : *Away Corner* (AC)

N : *Home Yellow* (HY)

O : *Away Yellow* (AY)

P : *Home Red* (HR)

Q : *Away Red* (AR)

Perhitungan jarak pada penelitian ini, menggunakan bantuan *Scikit-Learn Library* yang mana di dalamnya memiliki algoritma komputasi yang dapat mempermudah perhitungan data dalam jumlah yang banyak.

2.2.5 Confusion Matrix

Confusion matrix adalah suatu metode yang digunakan untuk melakukan perhitungan dan meggambarkan kinerja model klasifikasi (*classifier)* pada satu set data uji yang nilai sebenarnya diketahui. *Confusion matrix* dapat membantu dalam visualisasi kinerja suatu algoritma. Seperti yang pada tabel 2.1 yang menampilkan hasil prediksi pada masalah klasifikasi, jumlah prediksi benar dan salah dirangkum dengan nilai-nilai dan dipecah kepada masing-masing label. Hal ini dapat membantu untuk mengetahui kesalahan yang dibuat oleh *classifier.*

Tabel 2.1 *Confusion Matrix*

		Predict Label	
		Positive (P)	*Negative* (N)
Actual Label	P	*True Positive* (TP)	*False Negative* (FN)
	N	*False Positive* (FP)	*True Negative* (TN)

Dimana :

True Positive: merupakan data positif dan diprediksi benar

True Negative: merupakan data negatif dan diprediksi benar

False Positive: merupakan data negatif tetapi diprediksi sebagai data positif (error tipe 1)

False Negative: meupakan data positif tetapi diprediksi sebagai data negatif (error tipe 2)

Pada penelitian ini menggunakan 3 label klasifikasi, sehingga Tabel 2.1 berubah sesuai label klasifikasi yang ditunjukan pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix* 3×3

		Predict Label		
		Label D	Label H	Label A
Actual Label	Label D	DD	DH	DA
	Label H	HD	HH	HA
	Label A	AD	AH	AA

2.2.6 Akurasi dan Error

Akurasi merupakan jumlah prediksi yang benar dibagi dengan keseluruhan data prediksi. Akurasi dapat diperoleh meggunakan persamaan (2.7).

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.7}$$

Pada penelitian ini label yang digunakan berjumlah 3 label sehingga *Confusion matrix* menjadi 3×3. Sehingga penerapan persamaan (2.7) dengan menggunakan label Tabel 2.2 menghasilkan persamaan (2.8).

$$\text{Akurasi} = \frac{DD+HH+AA}{DD+DH+DA+HD+HH+HA+AD+AH+AA} \tag{2.8}$$

Error merupakan jumlah seluruh data prediksi yang salah dan dibagi dengan keseluruhan data prediksi. Dinyatakan dalam pesamaan (2.9).

$$\text{Error} = \frac{FP + FN}{TP + TN + FP + FN} \tag{2.9}$$

Kemudian sesuai dengan data label Tabel 2.2, persamaan (2.9) dituliskan menjadi persamaan (2.10).

$$\text{Error} = \frac{DH+DA+HD+HA+AD+AH}{DD+DH+DA+HD+HH+HA+AD+AH+AA} \tag{2.10}$$

Atau persamaan (2.10) dapat dituliskan menjadi persamaan (2.11).

$$\text{Error} = 1 - \text{Akurasi} \tag{2.11}$$

2.2.7 Presisi

Presisi adalah rasio jumlah prediksi yang benar-benar positif dari semua kelas positif yang diprediksi benar. Nilai presisi dapat dihitung dengan persamaan (2.12).

$$\text{Presisi} = \frac{\text{TP}}{\text{TP+FP}} \tag{2.12}$$

Dikarenakan setiap label memiliki prediksi positif, maka untuk menghitung presisi dari *classifier* secara keseluruhan harus dilakukan perhitungan nilai presisi setiap labelnya. Kemudian mengikuti tabel 2.2 didapatkan persamaan (2.13)

$$\text{Presisi} = \frac{\text{D}i}{\text{D}i+\text{H}i+\text{A}i} \tag{2.13}$$

Kemudian untuk menghitung presisi total dapat menggunakan persamaan (2.14).

$$\text{Presisi} = \frac{\text{Presisi D+Presisi H+ Presisi A}}{3} \tag{2.14}$$

2.2.8 Recall

Recall atau *Sensitivity* menggambarkan seberapa banyak prediksi yang benar dari seluruh kelas positif. Nilai *Recall* dapat dihitung dengan menggunakan persamaan (2.15)

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2.15}$$

Dalam penelitian ini setiap label memiliki prediksi positif, maka untuk menghitung *Recall* dari *Classifier* secara keseluruhan harus menghitung terlebih dahulu masing-masing prediksi labelnya mengikuti tabel 2.2 dan didapatkan persamaan (2.16).

$$\text{Recall} = \frac{i\text{D}}{i\text{D}+i\text{H}+i\text{A}} \tag{2.16}$$

Maka *Recall* dari *Classifier* diperoleh menggunakan persamaan (2.17)

$$\text{Recall} = \frac{\text{Recall D+Recall H+Recall A}}{3} \tag{2.17}$$

2.2.9 F-Score

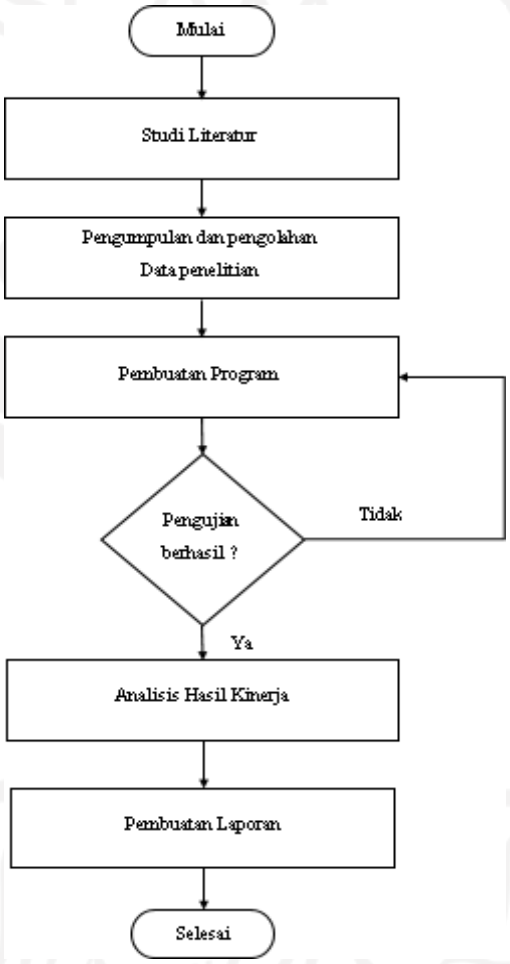
Pada pengukuran kinerja model, presisi dan *recall* merupakan pengukuran yang sama pentingnya. Presisi dan recall juga saling bertolak belakang. Recall bisa ditingkatkan semaksimal mungkin dengan cara memperbanyak prediksi sampel pada kelas positif yang mengakibatkan FP juga bertambah. Hal ini akan membuat presisi mejadi semakin berkurang karena tujuan utama dari evaluasi model menggunakan presisi adalah mengurangi jumlah FP. Berlaku juga sebaliknya untuk recall yang dipengaruhi oleh FN. Oleh sebab itu, dibutuhkan suatu cara untuk mendapatkan model yang seimbang. F-Score merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan [9]. *F-Score* dapat dihitung dengan menggunakan persamaan (2.18)

$$F - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.18)$$

BAB 3
METODOLOGI

3.1 Alur Penelitian

Pada penelitian ini, penulis meggunakan tahapan atau langkah-langkah agar penelitian ini menjadi lebih terstruktur, diilustrasikan oleh bagan alur seperti pada Gambar 3.1 beserta penjelasan mengenai setiap tahapannya.



Gambar 3.1 Bagan alur penelitian

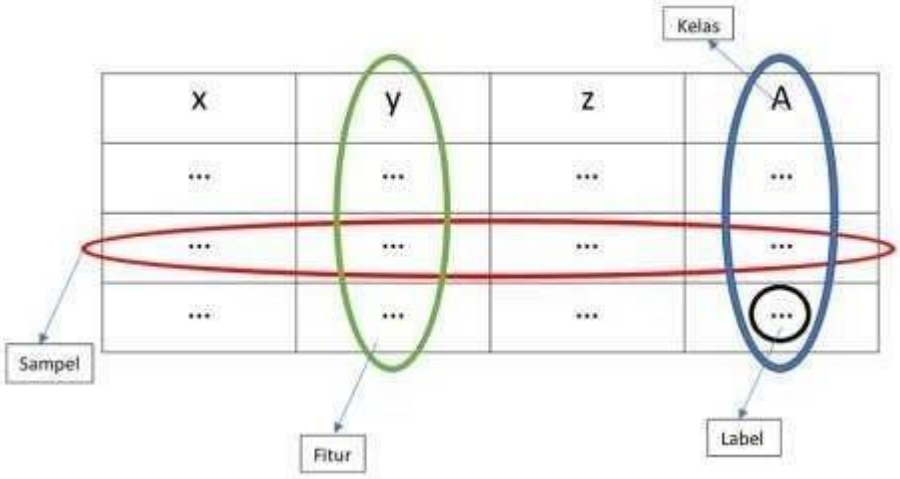
3.1.1 Studi Literatur

Pada tahapan ini, penulis mencari dan mempelajari sumber literatur yang berkaitan dengan topik penelitian yang dilakukan. sumber literatur ini antara lain berupa penelitian yang diterbitkan

sebelumnya yang dapat membantu proses penelitian yang dilakukan oleh penulis. Sumber literatur lain berupa Jurnal, *Paper*, dan buku yang berkaitan dengan topik pembahasan EPL, *Machine Learning*, *Naïve Bayes Classifier*, dan *K-Nearest Neighbors* serta metode klasifikasi lainnya.

3.1.2 Pengumpulan dan Pengolahan Data Penelitian

Pada tahap ini dilakukan proses pengumpulan dan pengolahan data. Tahapan ini cukup penting karena metode *Machine Learning* memerlukan data untuk melakukan pembelajaran agar dapat menentukan suatu keputusan atau prediksi. Jenis pembelajaran yang dilakukan pada penelitian ini adalah klasifikasi (*classification*) dimana *variable* output yang coba diprediksi berupa kategori. Pada dataset dalam *Machine Learning* terdapat fitur atau atribut yang dibutuhkan untuk proses pembelajaran *Dataset* pada *Machine Learning* terdiri dari beberapa bagian seperti yang ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Struktur Datasheet

Baris pada dataset disebut dengan sampel atau data point dan kolom pada dataset disebut dengan fitur atau atribut, kemudian kolom terakhir pada dataset disebut sebagai kelas. Fitur merupakan properti individual yang terukur atau karakteristik dari fenomena objek yang diamati (input) sedangkan kelas merupakan suatu hal yang coba diprediksi (output). Faktor yang mempengaruhi nilai prediksi pertandingan sepak bola adalah *Home Team*, *Away Team*, *Half Time Home Goal* (HTHG), *Half Time Away Goal* (HTHG), *Home Shot* (HS), *Away Shot* (AS), *Home Shot on Target* (HST), *Away Shot on Target* (AST), *Home Foul* (HF), *Away Foul* (AF), *Home Corner* (HC), *Away Corner* (AC), *Home Yellow Card* (HY), *Away Yellow Card* (AY), *Home Red Card* (HR), *Away Red Card* (AR). Beberapa faktor dipilih untuk menjadi fitur (*input*) yang diambil dari data selama 6 musim (2014/2015 – 2019/2020) setelah melakukan pengolahan diperoleh sebanyak 2280 sampel data.

Fitur-fitur tersebut dipilih karena mempengaruhi secara keseluruhan hasil suatu pertandingan. Jika salah satu fitur tersebut tidak digunakan maka membuat hasil akurasi penelitian menjadi lebih rendah.

Tabel 3.1 Data Input dan Output

Atribut data	Keterangan
Home team	Fitur (*input*)
Away team	Fitur (*input*)
Half Time Home Goal	Fitur (*input*)
Half Time Away Goal	Fitur (*input*)
Home Shot	Fitur (*input*)
Away Shot	Fitur (*input*)
Home Shot on Target	Fitur (*input*)
Away Shot on Target	Fitur (*input*)
Home Foul	Fitur (*input*)
Away Foul	Fitur (*input*)
Home Corner	Fitur (*input*)
Away Corner	Fitur (*input*)
Home Yellow Card	Fitur (*input*)
Away Yellow Card	Fitur (*input*)
Home Red Card	Fitur (*input*)
Away Red Card	Fitur (*input*)
FTR (*Full Time Result*)	Kelas (*output*)

- ***Home Team (HT)***

Home team adalah daftar nama tim kandang atau tim tuan rumah yang bertanding pada kompetisi *English Premier League*. Fitur ini dipilih karena penting untuk menjadi pebandingan dan membedakan antara satu tim dengan tim lainnya. Nama-nama tim kandang sudah dibuat *encoding* agar dapat dibaca oleh mesin. Data dan hasil encoding terdapat pada lampiran.

- ***Away Team (AT)***

Away team adalah daftar nama tim tandang atau tim tamu yang bertanding pada kompetisi *English Premier League*. Fitur ini dipilih karena penting untuk menjadi pebandingan dan membedakan antara satu tim dengan tim lainnya. Nama-nama tim tandang sudah dibuat *encoding* agar dapat dibaca oleh mesin. Data dan hasil encoding terdapat pada lampiran

- ***Halftime Home Goals (HTHG)***

  Halftime Home Goals adalah jumlah goal yang dicetak oleh tim kandang pada babak pertama.

  Fitur ini dipilih dikarena untuk menggambarkan gol yang dibuat pada babak petama dan meningkatkan performa model.

- ***Halftime Away Goals (HTAG)***

  Halftime Away Goals adalah jumlah goal yang dicetak oleh tim tandang pada babak pertama. Fitur ini dipilih dikarena untuk menggambarkan gol yang dibuat pada babak petama dan meningkatkan performa model.

- ***Home Shot (HS)***

Home Shot adalah jumlah tendangan kearah gawang yang dilakukan oleh tim kandang selama 90 menit pertandingan pada kompetisi *English Premiere League*. Fitur ini dipilih karena menggambarkan upaya untuk mencetak gol yang dilakukan oleh tim kandang.

- ***Away Shot (AS)***

Away Shot adalah jumlah tendangan kearah gawang yang dilakukan oleh tim tandang selam 90 menit pertandingan pada kompetisi *English Premiere League.* Fitur ini dipilih karena menggambarkan upaya untuk mencetak gol yang dilakukan oleh tim tandang.

- ***Home Shot on Target (HST)***

Home Shot on Target adalah jumlah tendangan tim kandang yang tepat sasaran menuju gawang lawan selama pertandingan bergulir. Fitur ini dipilih karena meggambar peluang untuk mencetak yang dilakukan oleh tim kandang.

- ***Away Shot on Target (AST)***

Away Shot on Target adalah jumlah tendangan tim tandang yang tepat sasaran menuju gawang lawan selama pertandingan bergulir. Fitur ini dipilih karena meggambar peluang untuk mencetak yang dilakukan oleh tim tandang.

- ***Home Foul (HF)***

Home Foul adalah jumlah pelanggaran yang dilakukan tim kandang selama pertandingan berlangsung. Fitur ini dipilih karena menggambarkan jumlah pelanggaran yang dilakukan oleh tim kandang. Semakin banyak jumlah pelanggaran maka akan semakin menguntungkan tim lawan.

- ***Away Foul (AF)***

Away Foul adalah jumlah pelanggaran yang dilakukan tim tandang selama pertandingan berlangsung. Fitur ini dipilih karena menggambarkan jumlah pelanggaran yang dilakukan oleh tim tandang. Semakin banyak jumlah pelanggaran maka akan semakin menguntungkan tim lawan.

- ***Home Corner (HC)***

  Home Corner adalah jumlah tendangan pojok yang dilakukan oleh tim kandang. Fitur ini dipilih karena melalui tendangan pojok tim memiliki peluang yang besar untuk mencetak gol.

- ***Away Corner (AC)***

  Away Corner adalah jumlah tendangan pojok yang dilakukan oleh tim tandang. Fitur ini dipilih karena melalui tendangan pojok tim memiliki peluang yang besar untuk mencetak gol.

- ***Home Yellow Card (HY)***

Home Yellow Card adalah jumlah kartu kuning yang didapatkan tim kandang selama pertandingan berlangsung. Fitur ini dipilih karena semakin banyak jumlah kartu yang didapat suatu tim maka akan mempengaruhi permainan pemain yang mendapatkan kartu kuning tersebut, dan membuat permainan tim tidak berjalan lancar. Jadi menguntungkan untuk tim lawan.

- ***Away Yellow Card (AY)***

Away Yellow Card adalah jumlah kartu kuning yang didapatkan tim tandang selama pertandingan berlangsung. Fitur ini dipilih karena semakin banyak jumlah kartu yang didapat suatu tim maka akan mempengaruhi permainan pemain yang mendapatkan kartu kuning tersebut, dan membuat permainan tim tidak berjalan lancar. Jadi menguntungkan untuk tim lawan.

- ***Home Red Card (HR)***

Home Red Card adalah jumlah kartu merah yang didapatkan tim kandang selama pertandingan berlangsung. Fitur ini dipilih karena tim yang mendapatkan kartu merah akan berkurang jumlah pemainnya yang berada di lapangan dan sangat menguntungkan untuk tim lawan.

- ***Away Red Card (AR)***

Away Red Card adalah jumlah pelanggaran kartu merah yang dikumpulkan tim tandangselama pertandingan berlangsung. Fitur ini dipilih karena tim yang mendapatkan kartu merah akan berkurang jumlah pemainnya yang berada di lapangan dan sangat menguntungkan untuk tim lawan.

- ***Fulltime Result (FTR)***

Fulltime Result) adalah hasil suatu pertandingan yang menampilkan hasil kemenangan Home/Away/Draw. Pada penelitian ini FTR dipilih untuk menjadi label output.

- ***Labeling***

Data statistik yang diperoleh dari [sitasi] menampilkan 3 label yaitu Home (kemenangan tim kandang), Away (kemenangan tim Tandang), dan Draw (seri). Ada dua tipe dataset yang ditemui dalam masalah pengklasifikasian , yaitu dataset seimbang (balanced dataset) dan dataset tidak

seimbang (imbalanced dataset). Pada penelitian ini dataset yang disiapkan adalah imbalanced dataset. Jumlah label pada penelitian ini ditunjukan pada table berikut.

Tabel 3.2 Label klasifikasi

Label	**jumlah**
Home	1042
Away	692
Draw	546

Algoritma *Machine Lerning* dapat menghasilkan prediksi lebih cepat ketika diberikan format data numerik. Oleh karena itu, *label encoding* digunakan untuk mengubah label menjadi bentuk numerik sehinga dapat diproses lebih cepat oleh mesin. Proses encoding dilakukan dengan menggunakan bantuan perangkat lunak *Microsoft excel*. Adapun label FTR yang telah dirubah menjadi bentuk numerik ditampilkan pada tabel 3.3.

Tabel 3.3 Label encoding

Label	**Hasil Encoding**
Home	1
Away	2
Draw	0

3.1.3 Pembuatan Program

Pembuatan program dilakukan dengan bantuan perangkat lunak (*software*) *Spyder*. *Software* ini digunakan karena memiliki *Interface* yang mudah bagi pemula dan dapat diintegrasikan dengan *virtual environment Anaconda. Anaconda Environment* digunakan untuk meng-install paket *library* seperti *pandas* dan *scikit-learn* yang digunakan pada penelitian ini. *Pandas* merupakan software *library* yang ditulis untuk bahasa pemrograman *Python* yang bertugas untuk memanipulasi dan menganalisis data. Sedangkan *scikit-learn* merupakan *library* yang di dalamnya terdapat berbagai macam algoritma pembelajaran untuk *Machine Learning*.

3.1.4 Pembagian Data Latih dan Data Uji

Dataset yang telah diolah kemudian dipecah menjadi data latih dan data uji. Data latih adalah data yang akan digunakan untuk membangun model, sedangkan data uji digunakan untuk menguji seberapa baik sistem bekerja. Data uji harus dipisahkan dari data yang akan dilatih dikarenakan model dapat mengingat data yang digunakan untuk melatihnya sehingga prediksi akan selalu bernilai benar untuk data yang telah digunakan sebelumnya. Pada penelitian ini pembagian data latih dan data uji dilakukan dengan 3 macam skenario pengujian seperti pada table 3.2.

Tabel 3.4 Label Klasifikasi

Pengujian	Data Latih	Data Uji
Skenario 1	75%	25%
Skenario 2	80%	20%
Skenario 3	90%	10%

Berdasarkan pengujian skenario yang ditampilkan pada Tabel 3.2, maka menghasilkan jumlah dari data latih dan data uji untuk masing-masing skenario seperti yang ditampilkan pada table 3.3. Pembagian data latih dan data uji dilakukan secara acak (*random*) dengan menggunakan bantuan *scikit-learn library*.

Tabel 3.5 Data Latih dan Data Uji

Pengujian	Data Latih	Data Uji	Jumlah
Skenario 1	1710	570	2280
Skenario 2	1824	456	2280
Skenario 3	2052	228	2280

Berikut adalah contoh dataset yang digunakan pada penelitian ini :

Tabel 3.6 Dataset

NO	HT	AT	HTHG	HTAG	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	FTR
1	1	2	1	1	14	4	6	2	13	19	9	3	2	2	0	1	1
2	3	4	1	2	11	13	3	3	16	10	3	6	1	1	0	0	0
3	5	6	0	1	14	5	5	4	14	20	4	0	2	4	0	0	2
4	7	8	0	0	19	11	6	4	10	10	8	9	1	2	0	0	2
5	9	10	0	0	12	7	2	2	14	9	2	8	0	3	0	0	2
6	11	12	1	1	10	7	5	2	18	9	6	3	3	1	0	0	0
7	13	14	0	0	18	10	4	4	12	10	8	5	1	0	1	1	2
8	15	16	1	0	12	12	5	6	8	11	2	6	1	2	0	0	1
9	17	18	0	1	12	13	0	5	8	11	3	3	1	5	0	0	2
10	19	20	1	3	9	11	2	3	6	7	4	3	1	1	0	0	2
Dst.																	

3.1.5 Analisis Hasil Kinerja

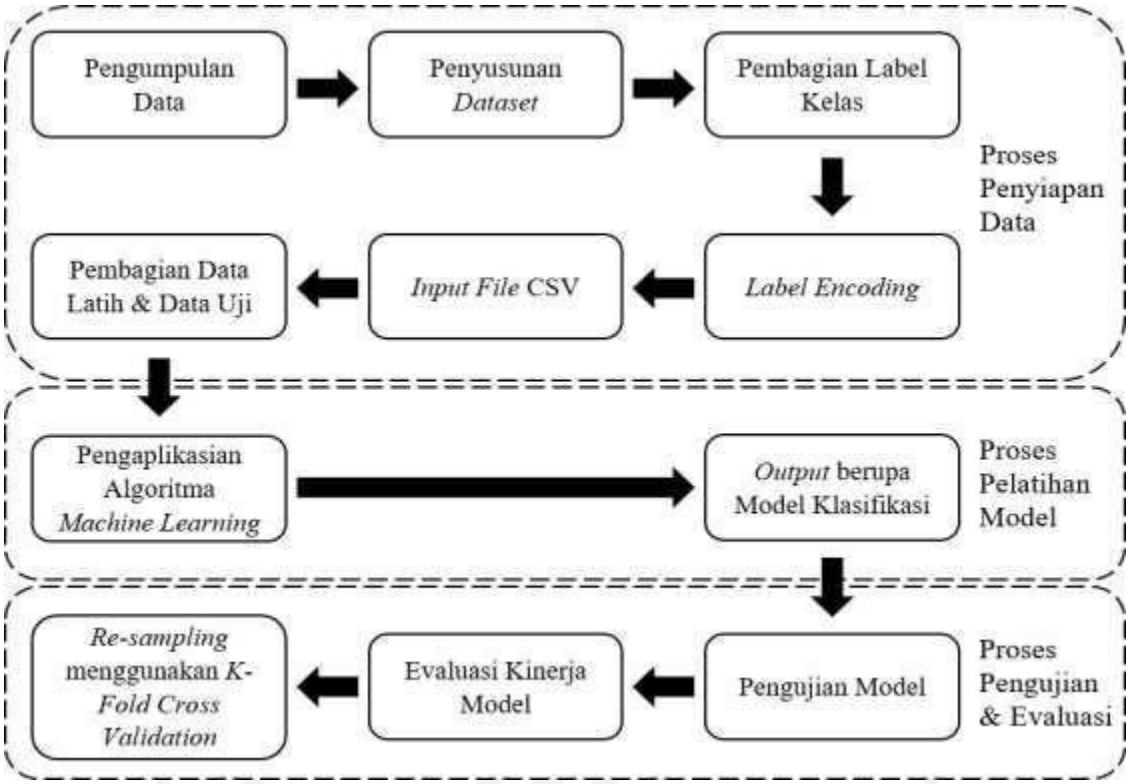
Analisa kinerja dari simulasi yang dilakukan pada penelitian kali ini dibagi menjadi 2 bagian. Pertama membandingkan performa masing-masing skenario pada setiap algortima, kemudian membandingkan performa antar algoritma secara keseluruhan. Parameter yang digunakan untuk menganalisisa performa kedua bagian tersebut adalah akurasi, presisi, *recall*, dan *f*-score.

3.1.6 Pembuatan Laporan

Pada tahap ini, penelitian telah selesai dilakukan, kemudian dilanjutkan dengan membuat laporan yang berisi kegiatan selama penelitian dari awal studi literatur hingga akhir yang berisi analisa dan hasil kesimpulan dan lampiran dari penelitian dengan lengkap

3.2 Perancangan Sistem

Perancangan sistem dibutuhkan agar sistem berjalan dengan sistematis dan tidak terjadi eror pada saat dilakukan pengujian. Adapun blok diagram dari sistem ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Blok Diagram Sistem

3.2.1 Proses Penyimpan Data

Tahap pertama pada sistem berisi proses penyiapan *dataset* sebelum kemudian masuk ke dalam algoritma *Machine Learning* untuk dilatih. Dimulai dengan menggumpulkan setiap atribut data dari sumber data yang telah dilampirkan sebelumnya untuk membuat suatu dataset yang terdiri dari 16 input dan 1 *output*. *Output* atau kelas dibagi menjadi 3 label kategori untuk mempermudah proses klasifikasi, kemudian masing-masing label tersebut diubah menjadi bentuk numerik untuk mempercepat proses komputasi dengan bantuan perangkat lunak *Microsoft Excel*. Pada proses terakhir. di fase ini, dataset dibagi menjadi 2 bagian, yaitu data latih dan data uji berdasarkan beberapa skenario pengujian.

3.2.2 Proses Pelatihan Model

Data latih yang telah dipisah akan digunakan untuk membangun model klasifikasi dengan mengaplikasikannya ke dalam algoritma *Machine Learning*. Algoritma yang digunakan pada penelitian adalah *Naïve Bayes* dan *K-Nearest Neighbors*.

3.2.3 Proses Pengujian dan Evaluasi

Pada tahap ini, model klasifikasi yang telah dibangun menggunakan data latih kemudian diuji hasil prediksi label kelasnya ketika diberikan fitur (*input)* data uji. Hasil prediksi kemudian dibandingkan dengan nilai label data uji yang sesungguhnya untuk melihat kinerja model klasifikasi. Parameter yang digunakan untuk menganalisis kinerja model adalah akurasi, presisi, *recall*, dan *f-score*.

Proses terakhir pada penelitian ini adalah melakukan *re-sampling* untuk melihat model tergeneralisasi dengan baik atau tidak (terjadi *overfitting & underfitting*).pengertian *Overfitting* yaitu suatu keadaan dimana data yang digunakan untuk pelatihan itu adalah yang "terbaik". Sehingga apabila dilakukan tes dengan menggunakan data berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai harapan). Sedangkan *Underfitting* merupakan keadaan dimana model pelatihan data yang dibuat tidak mewakilkan keseluruhan data yang akan digunakan nantinya. Sehingga menghasilkan performa buruk dalam pelatihan data. Oleh karena itu, pada penelitian ini akan digunakan teknik *K-Fold Cross Validation*.

K-Fold Cross Validation adalah metode statistik untuk mengevaluasi kinerja generalisasi yang lebih stabil dan menyeluruh daripada menggunakan pemisahan dataset menjadi satu set pelatihan dan pengujian. Dalam *K-Fold Cross Validation*, data akan dibagi beberapa bagian berulang kali (*iterasi*) yang kemudian model dilatih. Nilai dari parameter K menentukan seberapa banyak dataset akan dibagi dan seberapa banyak iterasi proses *training* dan *testing*. Untuk penentuan nilai K tidak ada penilaian khusus. Namun, biasanya nilai K yang ditetapkan adalah K=5 atau K=10. Pada penelitian ini, nilai K yang digunakan adalah 10 (K=10)

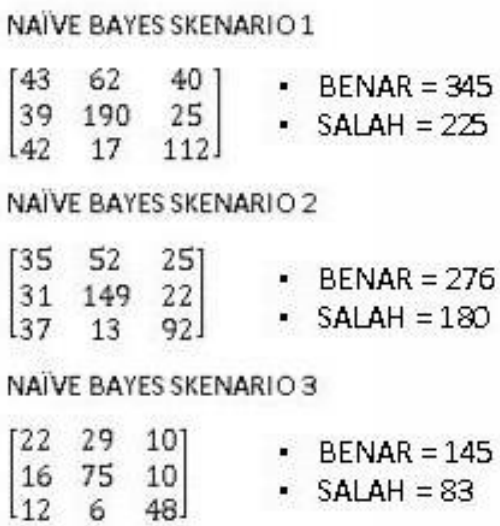
BAB 4
HASIL DAN PEMBAHASAN

4.1 Simulasi *Naïve Bayes*

Pada bagian ini, model *Machine Learning* telah selesai dilatih dengan algoritma *Naïve Bayes*. Kemudian prediksi yang dihasilkan dianalisis menggunakan *Confusion Matrix*. Pada *Confusion Matrix* terdapat beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model. Diantaranya adalah akurasi, eror, presisi, *recall,* dan *f-score.* Hasil dari setiap skenario dianalisa berdasarkan informasi yang dihasilkan dari *Confusion Matrix*.

4.1.1 Akurasi dan Error Pada Model *Naïve Bayes*

Simulasi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.1.

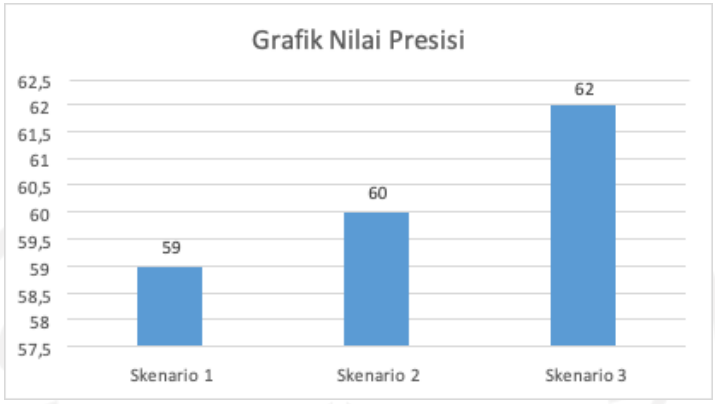


Gambar 4.1 Hasil *Confusion Matrix* Nilai Akurasi dan Error Pada Model *Naïve Bayes*

Dari hasil Gambar 4.1 dapat dilihat pada skenario 1 menghasilkan benar 345 dari 570 data uji atau akurasinya 60,5% dan eror 39,5%, skenario 2 menghasilkan benar 276 dari 456 data uji atau akurasinya 60,5% dan eror 39,5%, skenario 3 menghasilkan benar 145 dari 228 data uji atau akurasinya 63,5% dan eror 36,5%. Hasil akurasi yang berbeda pada ketiga skenario menunjukan bahwa semakin banyak data latih pada model maka akan meningkatkan nilai akurasi.

4.1.2 Presisi Pada Model *Naïve Bayes*

Pada simulasi ini dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan dari hasil *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.2

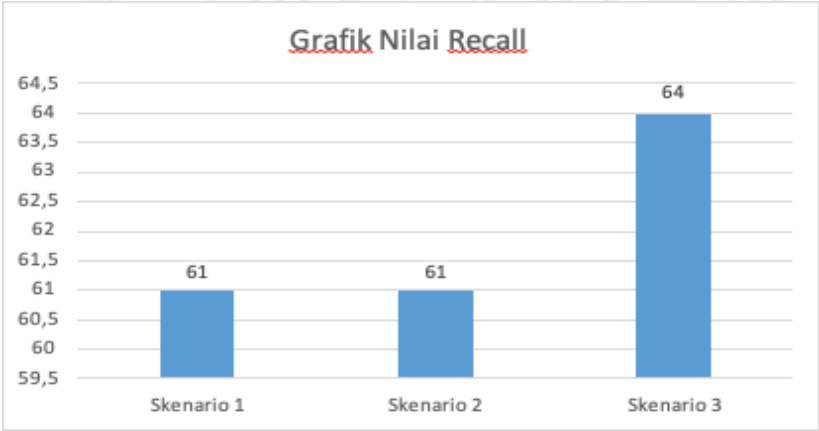


Gambar 4.2 Grafik Nilai Presisi

Dari hasil Gambar 4.2. terlihat bahwa skenario 1,2 dan 3 terus mengalami kenaikan. Pada skenario 1 bernilai 59% kemudian pada skenario 2 bernilai 60% dan skenario ketiga menghasilkan 62%. Terlihat bahwa pada skenario 1 memiliki nilai yang paling rendah, hal ini menunjukan bahwa skenario 1 dengan rasio pembagian data latih sebesar 75:25 banyak menghasilkan *False Positive* sehingga nilai yang dihasilkan rendah.

4.1.3 *Recall* Pada Model Naïve Bayes

Pada simulasi Recall dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *Recall* dapat dilihat pada Gambar 4.3.

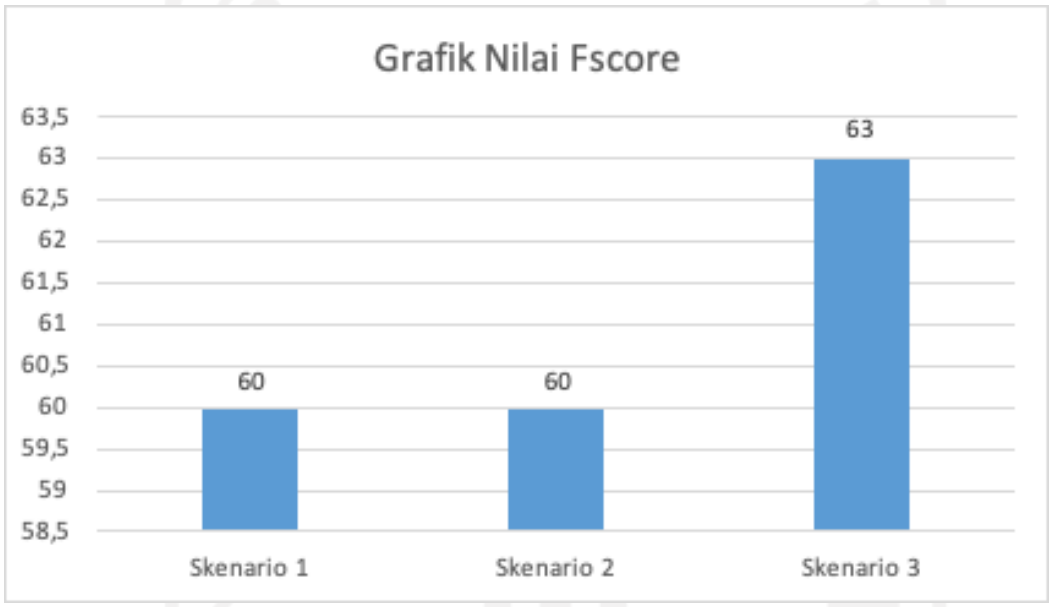


Gambar 4.3 Grafik Nilai Recall

Dari hasil Gambar 4.3. terlihat pada skenario 3 memiliki nilai recall terbaik. Pada skenario 1 dan 2 bernilai sama yaitu 61% sedangkan skenario 3 bernilai 64%. Hasil ini menunjukan bahwa skenario 1 dan 2 tidak dilatih dengan terlalu baik karena cukup banyak menghasilkan prediksi *False Positive*.

4.1.4 F-Score Pada Model *Naïve Bayes*

Pada simulasi *F-score* dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Grafik Nilai *F-score*

Pada hasil Gambar 4.4 dapat dilihat bahwa skenario 3 memiliki hasil yang paling tinggi yaitu 63%. Sedangkan skenario 1 dan 2 memiliki kesamaan nilai yaitu 60%. Hasil dari *F-score* rata-rata hampir sama dengan presisi nilai *Recall*.

4.2 Simulasi *K-Nearest Neighbors*

Pada simulasi ini, model *machine learning* telah selesai dilatih dengan algoritma KNN. Kemudian prediksi yang dihasilkan akan dianalisis menggunakan *matrix evaluation* atau disebut juga dengan *Confusion Matrix* .*Confusion Matrix* memiliki beberapa informasi yang bisa dimanfaatkan untuk mengevaluasi kinerja model antara lain adalah akurasi, error, presisi, *recall*, dan *f-Score*. Hasil dari setiap skenario dianalisa berdasarkan informasi yang didapat dari *confusion matrix.*

4.2.1 Akurasi dan Error Model *K-Nearest Neighbors*

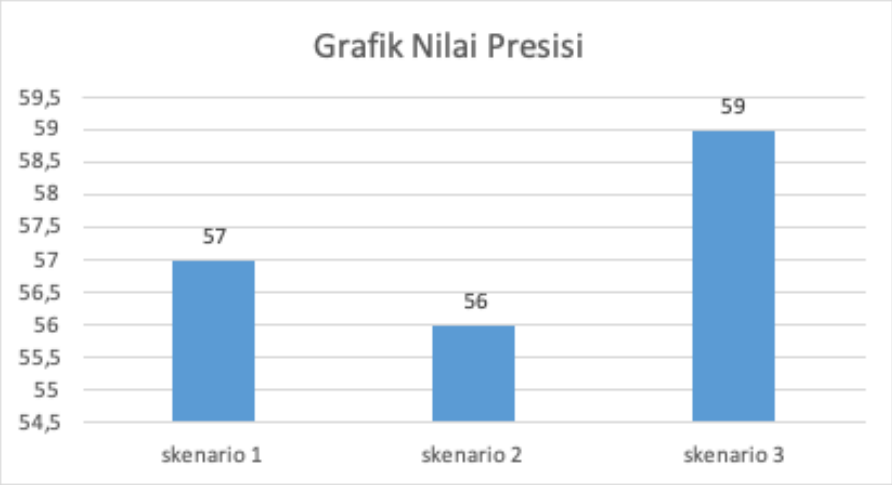
Pada simulasi akurasi dan eror dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter akurasi dan eror dapat dilihat pada Gambar 4.5.

Gambar 4.5 *Confusion Matrix* pada model KNN

Dari hasil Gambar 4.5 dapat dilihat skenario 1 menghasilkan benar 344 dari 570 data uji atau nilai akurasinya 60,3% dan eror 39,7%. Pada penelitian 2 menghasilkan benar 272 dari 456 data uji atau akurasinya 59,6% dan eror 40,4 %, kemudian pada penelitian 3 menghasilkan nilai benar 143 dari 228 data uji atau akurasinya 62,7% dan eror 37,3 %. Skenario 3 dengan rasio pembagian data uji dan latih sebesar 90:10 dapat dianggap paling optimal dalam membagi *dataset* pada *machine learning*. Kemudian pada data error pada skenario ke 3 berbanding terbalik dengan akurasi karena memiliki nilai error paling sedikit yaitu sebesar 37,3%

4.2.2 Presisi Model *K-Nearest Neighbors*

Pada simulasi presisi dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga model skenario. Hasil dari simulasi menurut parameter presisi dapat dilihat pada Gambar 4.6

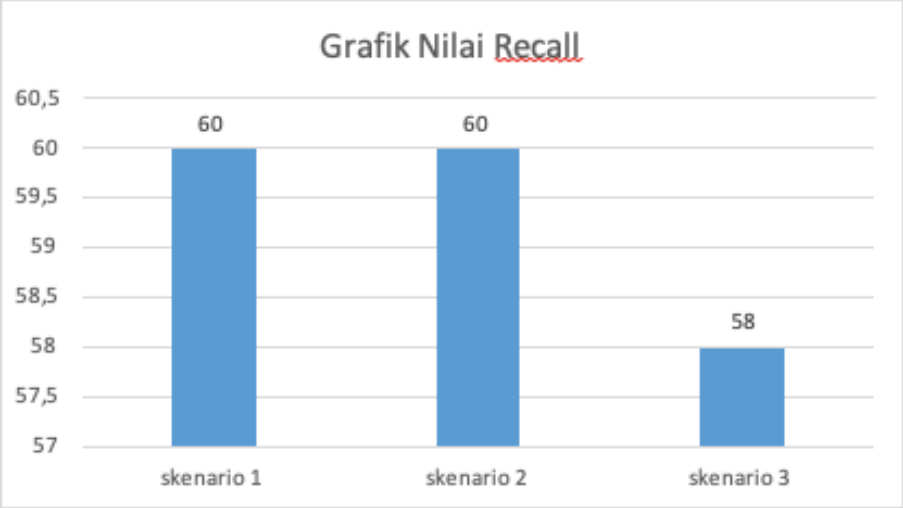


Gambar 4.6 Grafik Nilai Presisi

Dari Gambar 4.6 dapat dilihat bahwa skenario 3 memiliki nilai presisi yang paling tinggi dengan nilai 59%, kemudian skenario 1 memiliki nilai 57% dan skenario 2 memiliki nilai paling rendah yaitu 56%. Hasil ini menggambarkan bahwa pada saat skenario 2 banyak menghasilkan prediksi *False Positive* sehingga nilai presisi rendah.

4.2.3 Recall Model *K-Nearest Neighbors*

Pada simulasi recall dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *Confusion Matrix* ketiga model skenario. Hasil dari simulasi menurut parameter *recall* dapat dilihat pada Gambar 4.7.

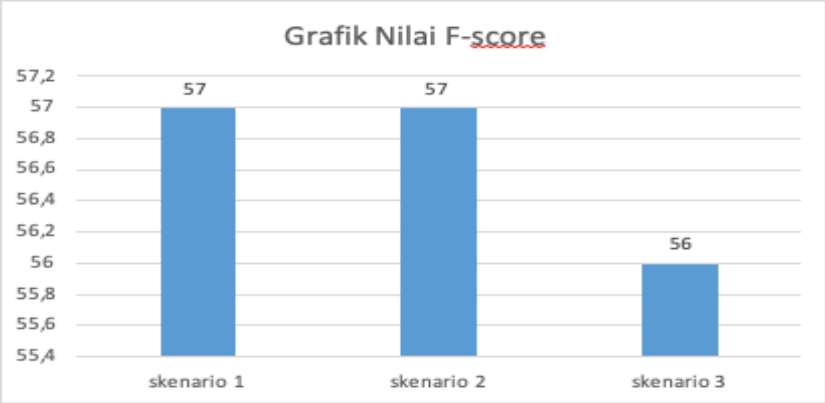


Gambar 4.7 Hasil Grafik Nilai *Recall*

Dari hasil Gambar 4.7 dapat dilihat, pada skenario 1 dan 2 memiliki kesamaan nilai yaitu 60%. Kemudian penurunan nilai terjadi pada skenario 3 yaitu bernilai 58%. Hal ini menunjukan skenario 3 memiliki banyak*false negative* sehingga nilai *recall* rendah.

4.2.4 F-Score Model *K-Nearest Neighbors*

Pada simulasi *f-score* dilakukan dengan menggunakan 3 skenario pembagian data uji dan data latih seperti yang ditunjukkan oleh Tabel 3.4. Analisa dilakukan pada hasil dari *confusion matrix* ketiga percobaan skenario. Hasil dari simulasi menurut parameter *f-score* dapat dilihat pada Gambar 4.8.

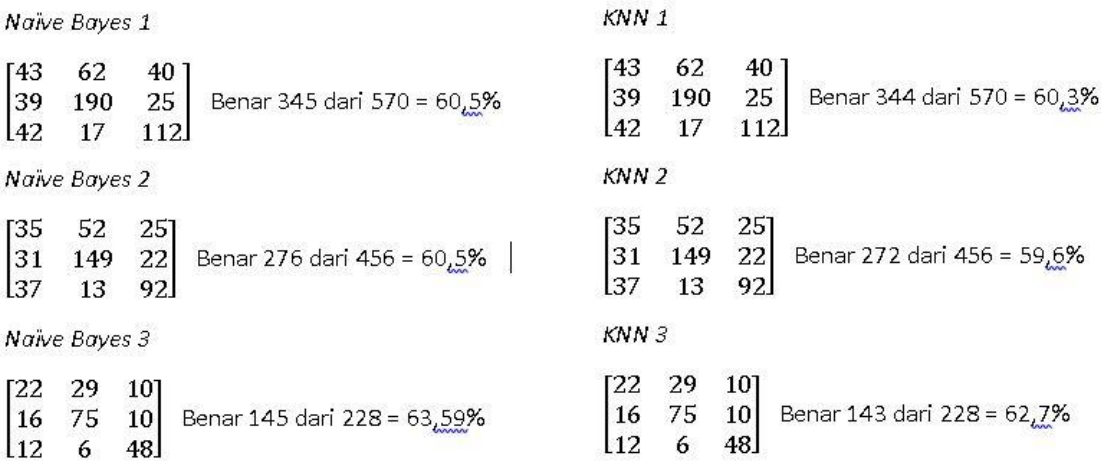


Gambar 4.8 Hasil Grafik Nilai *F-score*

Dari hasil Gambar 4.8 dapat dilihat, bahwa nilai skenario 1 dan 2 memiliki kesamaan nilai yaitu 57%. Kemudian pada skenario 3 memiliki nilai paling rendah sebesar 56%.

4.3 Perbandingan Performa Antar Algoritma

Perbandingan dari hasil model klasifikasi dari setiap skenario pengujian yang telah dilatih menggunakan algoritma *Naïve Bayes* dan *K-Nearest Neighbors* dilakukan untuk melihat kinerja dari masing-masing model algoritma. Untuk menganalisis masing-masing model, maka digunakan bantuan dari *confusion matrix* dengan parameter berupa akurasi. Hasil simulasi dari parameter tersebut dapat dilihat pada Gambar 4.9.



Gambar 4.9 *Confusion Matrix* hasil Pengujian kedua Model

Secara umum, pada pembuatan model *machine learning* memiliki rasio pembagian untuk data latih dan data uji berkisar pada nilai (90%-10%), (80%-20%), (70%-30%), dan (50%-50%) [9] dari keseluruhan dataset, hal ini masih dapat berubah ubah tergantung dari karakteristik data masing-masing dataset. Dataset yang digunakan pada penelitian ini memiliki 2280 sampel. Banyaknya data sampel akan mempengaruhi hasil pengujian, Oleh karena itu, pada penelitian ini bertujuan untuk mengetahui hasil dari berbagai skenario pengujian dengan berbagai macam rasio pembagian data latih dan data uji serta pengaruhnya terhadap dataset yang sudah dikumpulkan. data uji semakin sedikit maka model akan menghasilkan persentase yang rendah dari setiap parameter. Hal ini disebabkan karena pengaruh dari prediksi model yang salah terhadap data uji baik itu *false positive* maupun *false negative* akan sangat mempengaruhi dari presentase parameter.

Pada pelatihan model skenario pengujian terbaik untuk algoritma *Naïve Bayes* terjadi pada skenario 3 (rasio 90:10) dimana skenario ini memiliki nilai yang cukup tinggi. Berbeda dengan penelitian milik Razali [6] , pada penelitiannya dengan meggunakan model *Bayesian network* dengan rasio pembagian data latih dan data uji 90:10, menghasilkan nilai 75,09%. Nilai akurasi yang didapatkan oleh penulis cenderung lebih rendah dikarenakan peluang yang sering muncul pada seluruh fitur atau variabel menghasilkan nilai gain ratio yang tinggi dan mengakibatkan terjadinya kesalahan klasifikasi. Menurut socrates [10], pembobotan atribut kelas dapat meningkatkan pengaruh prediksi, dengan memperhitungkan bobot atribut terhadap kelas, maka yang menjadi dasar ketepatan klasifikasi bukan hanya probabilitas melainkan juga dari bobot setiap atribut tehadap kelas.

Pada skenario 2 (rasio 80:20) *Naive Bayes* mendapatkan nilai akurasi 60,5% sedangkan *KNN* menghasilkan nilai yang lebih rendah dari skenario lain. Namun, nilai yang didapatkan masih lebih baik dari penelitian milik Hijmans [5], dimana pada penelitiannya dengan data training dan data uji rasio 80:20 model Naïve Bayes menghasilkan akurasi prediksi 42% dan KNN 58,62%. Pada penelitian milik Hijmans [5], dijelaskan bahwa hasil yang didapatkan tejadi karena pemilihan fitur atau variabel yang kurang tepat dan masih perlu penambahan variabel informasi tentang lawan yang akan dihadapi untuk meningkatkan nilai akurasi.

Dari hasil pengujian kedua Algoritma seperti yang terlihat dari gambar 4.11 , model *Naïve bayes* menghasilkan True positive yang lebih banyak dibandingkan model *KNN*. *Naïve Bayes* memiliki kelebihan dalam pengujian menggunakan data dalam jumlah besar. Akan tetapi, perlu dilakukan pembobota atribut kelas serta optimasi data tambahan untuk meningkatkan akurasi dan pengaruh prediksi.

4.4 Pengujian *K-Fold Cross Validation*

Setelah melakukan re-sampling menggunakan *K-Fold Cross Validation*, maka didapati hasil pengujian dari setiap algoritma seperti yang ditampilkan Tabel 4.1. Pengujian *K-Fold Cross Validation* dilakukan dengan tujuan untuk menghindari *overfitting* dan *underfitting* pada suatu data model yang dilatih sebelumnya. *Overfitting* yaitu suatu keadaan dimana data yang digunakan untuk pelatihan itu adalah yang "terbaik". Sehingga apabila dilakukan tes dengan menggunakan data berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai harapan). Sedangkan *Underfitting* merupakan keadaan dimana model pelatihan data yang dibuat tidak mewakilkan keseluruhan data yang akan digunakan nantinya. Sehingga menghasilkan performa buruk dalam pelatihan data. Pada hasil pengujian, didapati bahwa skenario 3 *Naïve Bayes* memiliki nilai rata-rata akurasi tertinggi di antara 6 skenario pengujian yang dilakukan dengan nilai akurasi sebesar 63% dan Skenario 1,2 dan 3 *K-Nearest Neighbors* merupakan model dengan nilai rata-rata yang stabil dengan nilai 62% untuk semua skenario.

Tabel 4.1 Akurasi *K-Fold Cross Validation*

Iterasi	Akurasi *K-Fold Cross Validation*					
	NB 1	NB 2	NB 3	KNN 1	KNN 2	KNN 3
1	64%	67%	57%	57%	57%	57%
2	63%	59%	63%	63%	63%	63%
3	63%	65%	63%	63%	63%	63%
4	54%	57%	65%	65%	65%	65%
5	66%	59%	55%	55%	55%	55%
6	63%	66%	65%	65%	65%	65%
7	64%	62%	62%	62%	62%	62%
8	59%	62%	60%	60%	60%	60%
9	62%	62%	64%	64%	64%	64%
10	65%	64%	62%	62%	62%	62%
Rata-rata	**62%**	**62%**	**63%**	**62%**	**62%**	**62%**

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, kesimpulan yang dapat diambil dari penelitian ini adalah :

1. Pada penelitian ini, dapat disimpulkan bahwa model yang dilatih dengan menggunakan algoritma *Naïve Bayes* memiliki kinerja yang lebih bagus dibanding dengan algoritma *K-Nearest Neighbors*.
2. Kenaikan akurasi performa model terjadi ketika pembagian data latih dan data uji mengikuti skenario pengujian 3 (rasio 90:10).
3. Dalam melakukan prediksi hasil pertandingan sepakbola *English Premier League* diperlukan pemilihan dan pembobotan fitur atau atribut yang sesuai untuk meningkatkan nilai prediksi.

5.2 Saran

Setelah melakukan penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya agar lebih baik:

1. Untuk melakukan prediksi hasil pertandingan sepakbola *English Premiere League* dapat menambahkan sampel data yang baru dan lebih lengkap lagi.
2. Memilih atribut yang berbobot yang dapat meningkatkan nilai prediksi.
3. Menambahkan model *Deployment* agar hasil prediksi dapat diakses secara *online*.

DAFTAR PUSTAKA

[1] A. Saleh, "Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga," *Creat. Inf. Technol. J.*, vol. 2, no. 3, pp. 207–217, 2015.

[2] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," no. October, 2017.

[3] S. Winiarti, "JURNAL INFORMATIKA Vol 2, No. 2, Juli 2008," *Pemanfaat. Teorema Bayes Dalam Penentuan Penyakit THT*, vol. 2, no. 2, pp. 209–219, 2008.

[4] A. Rohman, "Model Algoritma K-Nearest Neighbor (K-Nn) Untuk Prediksi Kelulusan Mahasiswa," *Neo Tek.*, vol. 1, no. 1, 2015, doi: 10.37760/neoteknika.v1i1.350.

[5] A. Hijmans and S. Bhulai, "Dutch football prediction using machine learning classifiers," pp. 1–24, 2016.

[6] N. Razali, A. Mustapha, F. A. Yatim, and R. Ab Aziz, "Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL)," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, 2017, doi: 10.1088/1757-899X/226/1/012099.

[7] Y. F. Alfredo and S. M. Isa, "Football Match Prediction with Tree Based Model Classification," *Int. J. Intell. Syst. Appl.*, vol. 11, no. 7, pp. 20–28, 2019, doi: 10.5815/ijisa.2019.07.03.

[8] pandit football, "Mengapa Liga Inggris Lebih Kompetitif?," *www.panditfootball.com*, 2014. https://panditfootball.com/mengapa-liga-inggris-lebih-kompetitif/ (accessed Jan. 29, 2020).

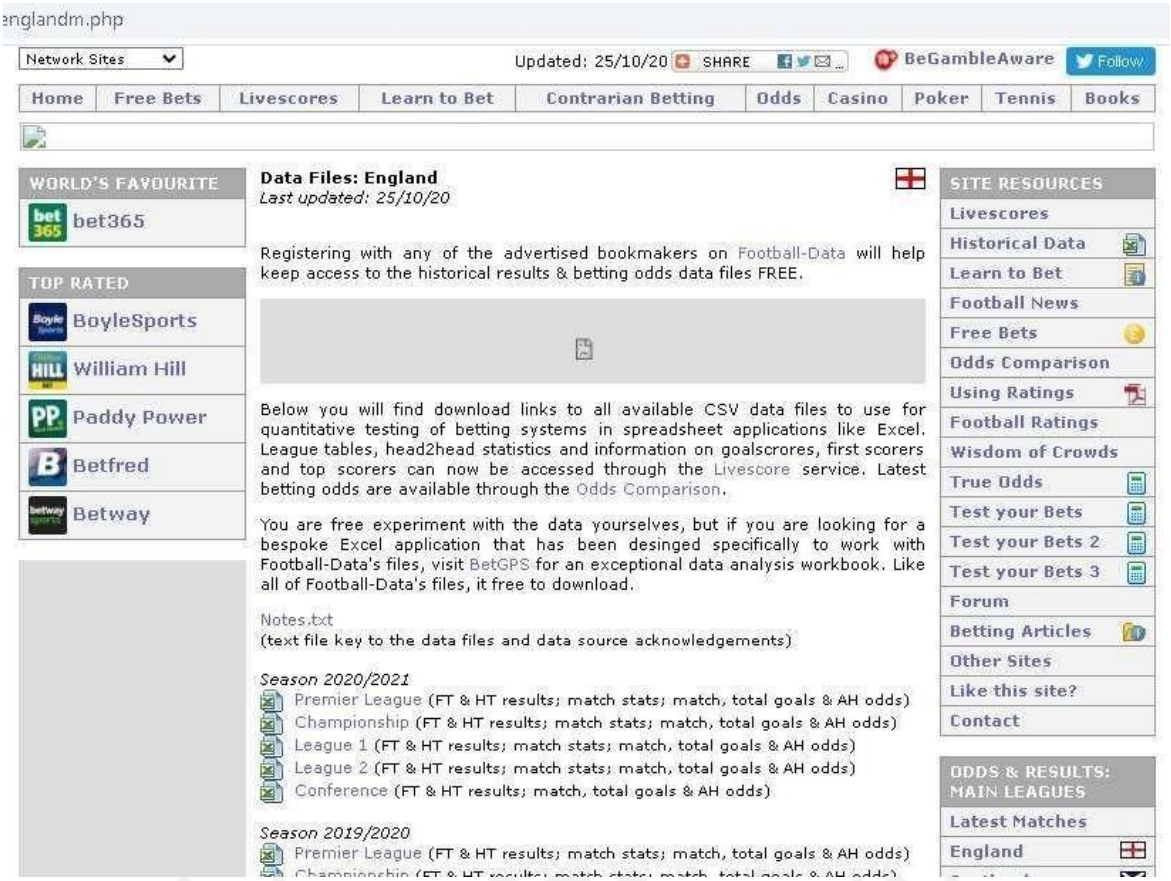
[9] J. W. G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," vol. 4, pp. 1–235, 2019.

[10] I. G. A. Socrates, A. L. Akbar, M. S. Akbar, A. Z. Arifin, and D. Herumurti, "Optimasi Naive Bayes Dengan Pemilihan Fitur Dan Pembobotan Gain Ratio," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 7, no. 1, p. 22, 2016, doi: 10.24843/lkjiti.2016.v07.i01.p03.

LAMPIRAN

Lampiran 1 – Database online Football data

Alamat web pengambilan data : www.football-data.co.uk



Lampiran 2 – Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	HomeTeam	AwayTeam	FTHG	FTAG	HTHG	HTAG	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	FTR
2	1	2	2	2	1	1	1	14	4	6	2	13	19	9	3	2	2	0	1
3	3	4	2	2	2	1	2	11	13	3	3	16	10	3	6	1	1	0	0
4	5	6	1	2	2	0	1	14	5	5	4	14	20	4	0	2	4	0	2
5	7	8	0	1	1	0	0	19	11	6	4	10	10	8	9	1	2	0	2
6	9	10	0	1	1	0	0	12	7	2	2	14	9	2	8	0	3	0	2
7	11	12	2	2	2	1	1	10	7	5	2	18	9	6	3	3	1	0	0
8	13	14	0	1	1	0	0	18	10	4	4	12	10	8	5	1	0	1	2
9	15	16	2	1	1	1	0	12	12	5	6	8	11	2	6	1	2	0	1
10	17	18	0	2	2	0	1	12	13	0	5	8	11	3	3	1	5	0	2
11	19	20	1	3	3	1	3	9	11	2	3	6	7	4	3	1	1	0	2
12	10	17	0	0	0	0	0	9	11	0	4	11	8	7	7	4	0	1	0
13	20	3	2	0	0	0	0	27	6	10	4	11	11	8	7	0	1	0	1
14	2	13	1	3	0	2	2	9	17	4	8	14	11	6	9	1	3	0	2
15	4	1	2	2	2	0	0	8	13	2	3	10	18	3	3	1	4	0	0
16	16	11	0	0	0	0	0	8	8	2	2	14	15	1	5	1	2	0	0
17	6	19	1	0	1	0	0	10	12	5	1	14	13	2	3	2	1	0	1
18	8	9	1	1	1	0	0	9	19	2	3	10	10	4	5	1	3	1	0
19	12	5	1	1	1	1	1	11	10	3	3	10	15	4	4	0	2	0	0
20	14	7	4	0	3	0	0	18	9	5	1	12	6	7	6	0	1	0	1
21	18	15	3	1	1	0	0	9	11	4	3	13	7	6	7	1	1	0	1
22	19	5	0	0	0	0	0	9	7	3	2	10	14	3	6	2	2	0	0
23	4	20	3	6	1	2	2	17	12	7	8	7	16	8	2	1	3	0	2
24	18	9	0	1	0	0	0	16	7	2	2	14	5	11	4	2	1	0	2
25	17	2	3	3	1	1	1	18	12	4	7	12	19	11	4	1	3	0	0
26	7	12	1	0	1	0	0	18	16	5	6	8	9	4	6	1	2	0	1
27	6	11	3	0	2	0	0	15	16	8	2	10	8	0	7	1	2	0	1
28	13	16	1	3	1	1	1	4	18	2	8	11	13	2	9	2	1	0	2

Lampiran 3 – Program Coding Naïve Bayes

```
# Naive Bayes

# Importing the libraries

import pandas as pd

# Importing the dataset
dataset = pd.read_csv('DATASKRIPSI.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.10, random_state = 10)

# Feature Scaling
#from sklearn.preprocessing import StandardScaler
#sc = StandardScaler()
#X_train = sc.fit_transform(X_train)
#X_test = sc.transform(X_test)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print (accuracies.mean()*100)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print (cm)

from sklearn.metrics import classification_report
print (classification_report(y_test, y_pred))

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print (accuracy)
```

Lampiran 4 – Program Coding K-Nearest Neighbors

```
# K-Nearest Neighbors (K-NN)

# Importing the libraries
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('DATASKRIPSI.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

dt=dataset.corr()

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 10)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
import numpy as np

grid = {'n_neighbors':np.arange(1,20),
        'p':np.arange(1,3),
        'weights':['uniform','distance']
       }

knn = KNeighborsClassifier(algorithm = "auto")
knn_cv = GridSearchCV(knn,grid,cv=10)
knn_cv.fit(X_train,y_train)
y_pred = knn_cv.predict(X_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

print("Hyperparameters:",knn_cv.best_params_)
print("Train Score:",knn_cv.best_score_)
print("Test Score:",knn_cv.score(X_test,y_test))

#Classification Report
from sklearn.metrics import classification_report
cr = classification_report(y_test, y_pred)
print(cr)
```