e- ISSN: 2828–299X

Perbandingan Metode Random Forest, K-Nearest Neighbor, dan SVM Dalam Prediksi Akurasi Pertandingan Liga Italia

Ahmad Assril Karim[1], Muhammad Ary Prasetyo[2], Muhammad Rohid Saputro[3]
[1,2,3]Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
E-mail: *[1]asrilae@gmail.com, [2]muhammadaryp18@gmail.com, [3]rohidjunior@gmail.com

Abstrak – *Prediksi (Forecasting) dilakukan hampir oleh semua kalangan, baik itu pengusaha, pemerintah, dan juga orang awam. Masalah yang diramalkan pun bermacam-macam, seperti prakiraan cuaca, jumlah penjualan, skor pertandingan, maupun tingkat inflasi ekonomi. Algoritma KNN, SVM, dan Random Forest merupakan metode machine learning yang dapat digunakan untuk mengatasi suatu permasalahan yang berhubungan dengan deret dan situasi peramalan. Perlunya mengetahui prediksi kemenangan tim pertandingan sepak bola liga italia selalu menjadi pembahasan yang tidak pernah dilewatkan oleh penggemar sepak bola, oleh karena itu peramalan sangat berguna untuk melihat gambaran-gambaran tentang masa mendatang sehingga para penggemar sepakbola dan pelatih tim sepak bola dapat mengantisipasi suatu kejadian yang mendatang. Misalnya, penggemar ataupun pelatih tim sepak bola liga italia dapat memperkirakan kemenangan tim dalam masa yang mendatang. Data yang digunakan pada penelitian ini yaitu 380 pertandingan Liga Italia pada musim 2020/2021. Data yang diperoleh diambil dari situs resmi http://www.football-data.co.uk. Metode Random forest menghasilkan akurasi sebesar 62%, metode SVM menghasilkan akurasi sebesar 64%, dan metode KNN menghasilkan akurasi sebesar 57%. Dengan hasil pengujian yang telah dilakukan, diketahui bahwa metode SVM merupakan metode yang lebih baik dibandingkan dengan metode Random Forest dan KNN dalam memprediksi akurasi pertandingan Liga Italia musim 2020/2021.*

Kata Kunci *— Data Mining, K-Nearest Neighbor, Random Forest, Support Vector Machine*

1. PENDAHULUAN

Peramalan (*Forecasting*) memiliki definisi sebagai alat atau teknik yang digunakan untuk melakukan prediksi suatu nilai pada masa mendatang dengan memperhatikan data atau faktor-faktor yang relevan, baik berdasarkan data yang sudah lampau maupun data yang ada pada saat ini. Prediksi biasanya digunakan dalam ramalan penjualan, namun juga bisa digunakan untuk memprediksi peluang kemenangan pada dunia olahraga salah satunya adalah sepak bola. Definisi ramalan atau prediksi peluang kemenangan dalam sepak bola berarti menentukan perkiraan besarnya peluang kemenangan pada masa yang mendatang. Prakiraan atau peramalan merupakan ilmu dan juga seni dalam memprediksi kejadian yang mungkin akan terjadi dimasa mendatang. Metode peramalan dapat digunakan untuk melakukan analisis suatu kemungkinan kemenangan, untuk memprediksi kemungkinan kekalahan, dan juga dapat memprediksi nilai pertandingan pada masa yang akan datang[1].

Liga sepak bola merupakan sebuah kompetisi pertandingan sepak bola yang diadakan oleh suatu negara. Hal tersebut diadakan untuk mengembangkan olahraga sepak bola supaya sebuah tim ataupun pemain dapat berkembang menuju ranah lebih tinggi. Salah satu liga sepak bola yang ada saat ini adalah Liga Italia. Liga Italia merupakan sebuah ajang kompetisi pertandingan sepak bola yang diselenggarakan di negara Italia, dimana dari hasil rutin liga tersebut didapatkan sebuah statistik yang dapat digunakan untuk mengembangkan bakat suatu tim atau pemain.

Selama ini dalam melakukan prediksi terhadap kemenangan tim sepak bola dalam liga italia masih diprediksi secara manual menggunakan statistik yang ada. Sehingga sering terjadi ketidakakuratan dalam meramal masa depan dan juga masih belum terdapat sistem tertentu yang bisa digunakan dalam membantu memprediksi kemenangan tim tertentu pada masa yang mendatang. Sehingga untuk melakukan peramalan kedepannya perlu dibuat sistem prediksi kemenangan tim liga italia supaya dapat membantu dalam hal peramalan.

Dalam membuat sistem prediksi pada penelitian ini digunakan metode algoritma *machine learning* yaitu metode SVM (*Support Vector Machine*), KNN (*K-Nearest Neighbor*), dan *Random Forest*. Algoritma SVM, KNN, dan *Random Forest* merupakan algoritma *machine learning* yang dapat digunakan untuk klasifikasi data secara *supervised learning,* karena itu label akan ditentukan oleh atribut-atribut yang sudah terklasifikasi secara terawasi. Sehingga tiga metode tersebut dapat digunakan untuk melakukan analisis suatu statistik liga sepak bola dan memprediksi kemenangan tim sepak bola liga italia pada masa mendatang, dimana semua metode tersebut dipilih juga sebagai perbandingan keakuratan masing-masing metode.

e- ISSN: 2828–299X

2. METODE PENELITIAN

2.1 Encoding

Preprocessing adalah proses menyiapkan data sebelum diolah oleh suatu model. Encoding merupakan bagian dari *preprocessing* yang bertujuan untuk mengubah tipe data supaya dapat dibaca oleh model. Data yang bertipe kategorikal harus diubah menjadi data numerik supaya dapat dibaca oleh model, karena model tidak dapat memproses data yang bertipe kategorikal. Ada 2 metode *encoding* yang dapat digunakan untuk mengubah tipe data yaitu *Label Encoding* dan *One Hot Encoding*. *Label Encoding* adalah teknik yang mengubah setiap nilai kolom secara berurutan, sedangkan *One Hot Encoding* adalah teknik yang mengubah setiap nilai dalam kolom menjadi kolom baru dan mengisinya dengan nilai biner yaitu 0 dan 1[2].

2.2 Random Forest

Random Forest merupakan bagian dari *Decision Tree*. *Random Forest* digunakan untuk membangun pohon keputusan yang memiliki *root node*, *inner node*, dan *leaf node* yang masing-masing digunakan untuk mengumpulkan data, berisi pertanyaan tentang data, dan membuat keputusan[3]. Tahapan untuk menyelesaikan masalah pada *random forest* adalah sebagai berikut[4] :

a. Menentukan jumlah *k* (*tree*) yang dipilih dari fitur *m*, dimana $k < m$.
b. Diambil sampel acak sebanyak *N* untuk setiap *k* dari dataset.
c. Setiap *k* dilakukan pengambilan subset prediktor (*p*) secara acak, dimana $m<p$.
d. Langkah kedua dan ketiga diulangi sebanyak *k*.
e. Hasil prediksi diperoleh dari *vote* terbanyak dari hasil klasifikasi sebanyak *k*.

2.3 K-Nearest Neighbor

K-Nearest Neighbor atau disebut juga *lazy learner* (pembelajar yang malas) mengklasifikasikan data berdasarkan kemiripan atau kedekatan terhadap data lainnya (*neighbor*). Metode ini digunakan untuk analisis klasifikasi dan juga prediksi. Langkah-langkah perhitungan pada KNN adalah sebagai berikut[4] :

a. Menentukan jumlah *k* (tetangga) yang akan digunakan.
b. Menghitung jarak kedekatan data menggunakan *Euclidean Distance* dengan persamaan (1) berikut:

$$d(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \text{.....(1)}$$

c. Hasil perhitungan jarak diurutkan dari tertinggi sampai terendah.
d. Menghitung jumlah setiap kelas berdasarkan *k*.
   Tentukan kelas baru bagi data uji dengan kelas mayoritas

2.4 Support Vector Machine

Support Vector Machine merupakan metode yang bekerja dengan prinsip *Structural Risk Minimization* (SRM) yang bertujuan menemukan *hyperplane* untuk memisahkan dua buah *class* pada input space[5]. *Hyperplane* ditentukan dengan cara mencari titik maksimum dan menghitung *margin hyperplane*. Margin adalah jarak *hyperplane* dengan data terdekat dari setiap kelas, sedangkan data yang paling dekat dengan *hyperplane* disebut *support vector*.

2.5 Confusion Matrix

Confusion Matrix merupakan salah satu metode untuk mengukur kinerja model yang telah dibuat. Nilai *confusion matrix* dapat dilihat pada tabel 1.

Tabel 1. *Confusion Matrix*

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Berikut rumus untuk mengukur nilai *accuracy, precision, recall* pada persamaan (2)(3)(4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \text{.....(2)}$$

$$Precision = \frac{TP}{TP+FP} \text{.....(3)}$$

$$Recall = \frac{TP}{TP+FN} \text{.....(4)}$$

2.6 Dataset

Dataset yang digunakan pada penelitian ini yaitu dataset pertandingan Liga Italia pada musim 2020/2021. Data tersebut didapat dari www.football-data.co.uk . Liga Italia diisi oleh 20 tim dan setiap tim akan bermain 38 kali, 19 kali bermain di kandang, dan 19 kali bermain tandang. Atribut yang dipakai pada dataset dapat dilihat pada tabel 2.

e- ISSN: 2828–299X

Tabel 2. Data Input dan Output

Atribut	Keterangan	Deskripsi
HomeTeam	*Input*	Nama dari tim kandang yang bermain.
AwayTeam	*Input*	Nama dari tim tandang yang bermain.
HS	*Input*	*Home Team Shots*. Jumlah tendangan kearah gawang oleh tim kandang.
AS	*Input*	*Away Team Shots*. Jumlah tendangan kearah gawang oleh tim tandang.
HST	*Input*	*Home Team Shots on Target*. Jumlah tendangan kearah gawang yang tepat sasaran oleh tim kandang.
AST	*Input*	*Away Team Shots on Target*. Jumlah tendangan kearah gawang yang tepat sasaran oleh tim tandang.
HC	*Input*	*Home Team Corners*. Jumlah tendangan pojok oleh tim kandang.
AC	*Input*	*Away Team Corners*. Jumlah tendangan pojok oleh tim tandang.
HF	*Input*	*Home Team Fouls Commited*. Jumlah pelanggaran yang dilakukan oleh tim kandang.
AF	*Input*	*Away Team Fouls Commited*. Jumlah pelanggaran yang dilakukan oleh tim tandang.
HY	*Input*	*Home Team Yellow Cards*. Jumlah kartu kuning yang didapat oleh tim kandang.
AY	*Input*	*Away Team Yellow Cards*. Jumlah kartu kuning yang didapat oleh tim tandang.
HR	*Input*	*Home Team Red Cards*. Jumlah kartu merah yang didapat oleh tim kandang.
AR	*Input*	*Away Team Red Cards*. Jumlah kartu merah yang didapat oleh tim tandang.
HTHG	*Input*	*Half Time Home Team Goals*. Skor tim kandang di babak pertama pada pertandingan.
HTAG	*Input*	*Half Time Away Team Goals*. Skor tim tandang di babak pertama pada pertandingan.
FTR	*Output*	*Full Time Result*. Hasil akhir pertandingan, memiliki 3 label yaitu A (*Away*) yang berarti tim tandang menang, D (*Draw*) yang berarti kedua tim seri, dan H (*Home*) yang berarti tim kandang menang.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Data Set Penelitian

Dataset penelitian dengan nama “I1.csv” terlebih dahulu dilakukan proses input di google colab. Dataset tersebut di import dan ditampilkan di sistem seperti pada gambar 1.

```
[ ] dataset = data[['HomeTeam','AwayTeam','HS','AS','HST','AST','HC','AC','HF','AF',
                    'HY','AY','HR','AR','HTHG','HTAG','FTR']]
    dataset
```

	HomeTeam	AwayTeam	HS	AS	HST	AST	HC	AC	HF	AF	HY	AY	HR	AR	HTHG	HTAG	FTR
0	Fiorentina	Torino	17	6	8	3	4	1	17	20	3	1	0	0	0	0	H
1	Verona	Roma	9	15	7	7	0	6	19	20	2	0	0	0	0	0	D
2	Parma	Napoli	4	12	2	7	4	2	16	12	4	1	0	0	0	0	A
3	Genoa	Crotone	10	11	8	5	6	4	13	14	4	2	0	0	3	1	H
4	Sassuolo	Cagliari	21	8	10	4	7	3	9	18	1	3	0	0	0	0	D

Gambar 1. Dataset

e- ISSN: 2828–299X

Untuk Label pada penelitian ini terdapat pada kolom *Full Time Result* (FTR) yaitu *away* (A) merupakan hasil kemenangan untuk tim *Away* (tandang) , *draw* (D) merupakan hasil seri , dan *home* (H) merupakan hasil kemenangan untuk tim *Home* (kandang).

3.2. Hasil Preprocessing

a. Encoding data

Pada kolom *HomeTeam* dan *AwayTeam* yang bertipe data string diubah menjadi integer, pada gambar 2. Proses *encoding* yang digunakan yaitu *Label Encoding* supaya nilai pada kolom berurutan.

```
[ ] from sklearn.preprocessing import LabelEncoder
    le=LabelEncoder()
    dataset['HomeTeam']=le.fit_transform(dataset['HomeTeam'])
    dataset['AwayTeam']=le.fit_transform(dataset['AwayTeam'])
```

Gambar 2. *Encoding*

b. Pembagian data latih dan data uji (Split data)

Dataset dibagi menjadi data training (data latih) sebesar 80% dan data testing (data uji) sebesar 20% dari 380 data yang digunakan, pembagian data dilakukan dengan *Train Test Split* seperti pada gambar 3.

```
[ ] X = dataset[dataset.columns[0:16]]
    Y = dataset['FTR']
```

```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score

    # membagi dataset menjadi training dan testing
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=1)
```

Gambar 3. *Split Data*

Data terlebih dahulu dipisah menjadi atribut dan label, Atribut ditandai dengan huruf X, label ditandai dengan huruf Y. Kemudian dilakukan pembagian data menjadi data latih dan data uji dengan fungsi *train_test_split()*. Atribut yang digunakan yaitu *HomeTeam*, *AwayTeam*, HS, AS, HST, AST, HC, AC, HF, AF, HY, AY, HR, AR, HTHG, HTAG. Sedangkan label yang digunakan yaitu FTR.

3.3. Prediksi dengan Algoritma yang digunakan

a. Algoritma Random Forest

Hasil prediksi yang dilakukan menggunakan algoritma Random Forest mendapatkan tingkat akurasi sebesar 62%. Presentase tingkat akurasi dihasilkan dari menghitung data latih penelitian pada gambar 4.

```
[ ] from sklearn.ensemble import RandomForestClassifier
    ranfor = RandomForestClassifier(n_estimators = 100, random_state = 0)
    ranfor.fit(X_train, y_train)

    RandomForestClassifier(random_state=0)

[ ] from sklearn import metrics
    ranfor_pred = ranfor.predict(X_test)
    print(ranfor_pred)  #hasil prediksi
    print(y_test)  #jawaban yang sebenarnya
    print(metrics.accuracy_score(y_test, ranfor_pred))

    ['A' 'H' 'H' 'H' 'D' 'A' 'H' 'D' 'H' 'A' 'A' 'H' 'A' 'H' 'A' 'H' 'D' 'A'
     'H' 'H' 'A' 'A' 'A' 'H' 'H' 'A' 'H' 'A' 'H' 'H' 'H' 'H' 'H' 'D' 'H' 'H'
     'A' 'A' 'D' 'H' 'H' 'A' 'A' 'H' 'D' 'H' 'H' 'A' 'D' 'H' 'A' 'D' 'A' 'A'
     'A' 'H' 'A' 'H' 'H' 'A' 'A' 'H' 'A' 'H' 'A' 'H' 'H' 'H' 'A' 'D' 'H' 'H'
     'H' 'H' 'H' 'A']
    244    D
    180    H
    277    D
    185    H
    233    A
          ..
    323    D
    324    H
    165    H
    192    H
    191    A
    Name: FTR, Length: 76, dtype: object
    0.618421052631579
```

Gambar 4. *Random Forest*

Pada pengujian menggunakan *Random Forest*, terlebih dahulu dilakukan pembuatan model kemudian ditampung pada sebuah variabel, model *Random Forest* yang dibuat ditambahkan sebuah parameter yaitu *n_estimator* sebesar 100. Setelah itu dilakukan uji akurasi prediksi dari model yang sudah dilatih terhadap data uji.

b. Algoritma KNN

Hasil prediksi yang dilakukan menggunakan algoritma KNN mendapatkan tingkat akurasi sebesar 57%. Presentase tingkat akurasi dihasilkan dari menghitung data latih penelitian pada gambar 5.

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(X_train, y_train)

    KNeighborsClassifier()

[ ] from sklearn import metrics
    knn_pred = knn.predict(X_test)
    print(knn_pred)  #hasil prediksi
    print(y_test)  #jawaban yang sebenarnya
    print(metrics.accuracy_score(y_test, knn_pred))

['A' 'H' 'H' 'H' 'A' 'A' 'H' 'H' 'H' 'A' 'A' 'A' 'D' 'D' 'H' 'A' 'D' 'H'
 'H' 'H' 'A' 'A' 'A' 'D' 'H' 'H' 'H' 'A' 'H' 'H' 'D' 'D' 'A' 'A' 'H' 'H'
 'D' 'A' 'H' 'H' 'H' 'A' 'A' 'A' 'A' 'H' 'H' 'H' 'D' 'H' 'D' 'D' 'A' 'A'
 'D' 'H' 'A' 'D' 'H' 'D' 'A' 'D' 'H' 'A' 'A' 'H' 'H' 'H' 'H' 'D' 'H' 'H'
 'H' 'H' 'H' 'A']
244    D
180    H
277    D
185    H
233    A
      ..
323    D
324    H
165    H
192    H
191    A
Name: FTR, Length: 76, dtype: object
0.5657894736842105
```

Gambar 5. *K-Nearest Neighbor*

Pada pengujian menggunakan KNN, nilai *k* yang digunakan yaitu 5. Pada coding Gambar 5 *k* disebutkan dengan nama *n_neighbors*.

c. Algoritma SVM

Hasil prediksi yang dilakukan menggunakan algoritma SVM mendapatkan tingkat akurasi sebesar 64%. Presentase tingkat akurasi dihasilkan dari menghitung data latih penelitian pada gambar 6.

```
[ ] from sklearn.svm import SVC
    # membuat objek SVC
    svm = SVC(kernel='poly')
    svm.fit(X_train, y_train)

    SVC(kernel='poly')

[ ] from sklearn import metrics
    svm_pred = svm.predict(X_test)
    print(svm_pred)  #hasil prediksi
    print(y_test)  #jawaban yang sebenarnya
    print(metrics.accuracy_score(y_test, svm_pred))

['A' 'H' 'D' 'H' 'A' 'A' 'H' 'A' 'D' 'A' 'A' 'D' 'A' 'H' 'A' 'H' 'D' 'A'
 'H' 'H' 'H' 'A' 'A' 'H' 'H' 'A' 'H' 'A' 'H' 'H' 'H' 'H' 'A' 'A' 'H' 'A'
 'H' 'A' 'D' 'H' 'H' 'A' 'A' 'H' 'D' 'H' 'D' 'A' 'D' 'H' 'A' 'D' 'A' 'A'
 'A' 'H' 'A' 'H' 'A' 'A' 'H' 'H' 'A' 'H' 'A' 'H' 'H' 'H' 'A' 'A' 'H' 'H'
 'H' 'H' 'H' 'A']
244    D
180    H
277    D
185    H
233    A
      ..
323    D
324    H
165    H
192    H
191    A
Name: FTR, Length: 76, dtype: object
0.6447368421052632
```

Gambar 6. *Support Vector Machine*

Pada pengujian menggunakan SVM, ditambahkan sebuah parameter yaitu kernel, kernel yang digunakan adalah *poly*.

4. SIMPULAN

Pada penelitian ini melakukan Perbandingan Metode *Random Forest*, *K-Nearest Neighbor*, dan SVM Dalam Prediksi Akurasi Pertandingan Liga Italia. Data yang digunakan yaitu data pertandingan Liga Italia pada musim 2020/2021. Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa hasil

e- ISSN: 2828–299X

pengujian menggunakan metode *Random Forest* memperoleh tingkat akurasi sebesar 62%, metode *K-Nearest Neighbor* memperoleh tingkat akurasi sebesar 57%, dan metode SVM memperoleh tingkat akurasi sebesar 64%. Dari hasil pengujian yang telah dilakukan pada dataset Liga Italia musim 2020/2021, menunjukkan bahwa metode SVM memperoleh hasil yang lebih baik jika dibandingkan dengan metode *Random Forest* dan *K-Nearest Neighbor*.

5. SARAN

Saran yang dapat diberikan untuk mengembangkan penelitian ini lebih lanjut yaitu :

1. Penambahan dataset dari beberapa musim sebelum atau setelah musim 2020/2021 agar akurasi yang didapat lebih tinggi.
2. Gunakan metode lain untuk memprediksi pertandingan Liga Italia, seperti *XGBoost*.

DAFTAR PUSTAKA

[1] Sekarningrum, A. 2022. Apa itu forecasting? Pahami pentingnya forecasting untuk perkembangan bisnis. https://www.ekrut.com/media/forecasting-adalah . Diakses pada tanggal 15 Desember 2022.

[2] Hamami, Faqih, Iqbal Ahmad Dahlan. 2022. Klasifikasi Cuaca Provinsi Dki Jakarta Menggunakan Algoritma Random Forest Dengan Teknik Oversampling. Jurnal TEKNOINFO, Vol. 16, No. 1, 87-92.

[3] Siburian, Vanissa Wanika, Ika Elvina Mulyana. 2018. Prediksi Harga Ponsel Menggunakan Metode Random Forest. Prosiding Annual Research Seminar 2018 Computer Science and ICT, Vol.4 No.1, 144-147.

[4] Erdiansyah, Urmi, dkk. 2022. Komparasi Metode K-Nearest Neighbor dan Random Forest Dalam Prediksi Akurasi Klasifikasi Pengobatan Penyakit Kutil. Jurnal Media Informatika Budidarma, Vol 6, Nomor 1, Page 208-214.

[5] Widaningsih, Sri. 2019. Perbandingan Metode Data Mining Untuk Prediksi Nilai Dan Waktu Kelulusan Mahasiswa Prodi Teknik Informatika Dengan Algoritma C4.5, Naïve Bayes, Knn, Dan Svm. Jurnal Tekno Insentif, Vol. 13, No. 1, Hal 16-25.

[6] Lumbanraja, Favorisen.R. 2020. Prediksi Jumlah Penderita Penyakit Tuberkulosis Di Kota Bandar Lampung Menggunakan Metode Svm (Support Vector Machine). Kumpulan jurnaL Ilmu Komputer (KLIK), Vol 07, No. 3, 320-330.

[7] Sitepu, Ade Clinton. 2021. Analisis Kinerja Support Vector Machine dalam Mengidentifikasi Komentar Perundungan pada Jejaring Sosial. Jurnal Media Informatika Budidarma, Volume 5, Nomor 2, Page 475-484.