BAB V

HASIL DAN PEMBAHASAN

Hasil dan pembahasan dari penelitian ini terdiri dari hasil pengujian akurasi *machine learning* dengan algoritma KNN dan SVM yang sudah dilakukan. Berikut penjelasan terkait hasil penelitian yang sudah dilakukan.

5.1 Hasil

Sebelum melakukan perancangan *machine learning* pada penelitian ini telah dilakukan suatu rancangan pemodelan dalam bentuk *flowchart* untuk dijadikan gambaran alur dalam melakukan penelitian. Setelah melakukan penelitian dalam perancangan *machine learning* dengan algoritma KNN dan SVM, berikut penjelasan terkait hasil yang sudah dilakukan.

5.1.1 *Library* Pyhton

Library merupakan kumpulan *package* yang mempunyai fungsi dan tujuan untuk memudahkan dalam melakukan pembuatan kode program pada python. Dengan melakukan *import library*, program nantinya akan berjalan sesuai dengan kebutuhan dalam perancangan *machine learning*. Berikut beberapa *library* yang digunakan dalam penelitian.

a Pandas

Digunakan untuk memuat dan membaca sebuah *dataset* dalam bentuk *file* seperti csv dan excel dimana pada penelitian ini *dataset* diperlukan untuk kebutuhkan penelitian.

b Numpy

Digunakan untuk mengelola data *array* dan juga dapat melakukan operasi vektor dan matriks.

c Matplotlib

Digunakan untuk membuat visualisasi seperti diagram untuk menampilkan visualisasi data dengan baik agar terlihat lebih menarik.

d Sklearn

Digunakan untuk memberikan *package* untuk keperluan pengujian dalam algoritma.

5.1.2 *Dataset*

Awal dari melakukan persiapan data yaitu dengan mencari *dataset* menggunakan platform Yahoo Finance. Pada platform tersebut didapatkan *dataset* saham PT Bank Mandiri Tbk dengan nama BMRI.JK yang dimana data tersebut aktif berjalan memberikan informasi mengenai harga saham yang sedang terjadi dari hari – ke hari. Berikut *dataset* yang digunakan dalam penelitian.

Tabel 5. 1 Dataset Awal

No	Date	Open	High	Low	Close	Adj Close	Volume
0	2012-06-07	3450	3550	3425	3450	2499.356934	36661000
1	2012-06-08	3475	3500	3375	3475	2517.468506	22556000
2	2012-06-11	3500	3525	3475	3500	2535.579834	17709000
3	2012-06-12	3475	3500	3425	3475	2517.468506	10652000
4	2012-06-13	3500	3525	3475	3500	2535.579834	18399000
5	2012-06-14	3400	3500	3375	3400	2463.134521	29538000
6	2012-06-15	3500	3500	3400	3500	2535.579834	17207000

Dataset yang diambil berjumlah sekitar 2479 data selama 10 tahun dimana dimulai dari tanggal 7 Juni 2012 sampai 7 Juni 2022. Pada penelitian ini fitur yang digunakan hanya *close* untuk memprediksi harga saham penutupan PT Bank Mandiri Tbk.

Selanjutnya *dataset* tersebut diolah terlebih dahulu melalui Microsoft Excel untuk dilakukan pengubahan data ke dalam *time series*. Karena kegiatan dilakukan untuk bisa melakukan *time series* pemograman prediksi saham. Berikut tabel *dataset* yang sudah diolah ke dalam *time series.*

Tabel 5. 2 Dataset Time Series

Date	x(n-5)	x(n-4)	x(n-3)	x(n-2)	x(n-1)	x(n)	x(n+1)
2012-06-14	3450	3475	3500	3475	3500	3400	3500
2012-06-15	3475	3500	3475	3500	3400	3500	3550
2012-06-18	3500	3475	3500	3400	3500	3550	3575
2012-06-19	3475	3500	3400	3500	3550	3575	3625
2012-06-20	3500	3400	3500	3550	3575	3625	3625
2012-06-21	3400	3500	3550	3575	3625	3625	3600
2012-06-22	3500	3550	3575	3625	3625	3600	3500

Berdasarkan tabel 5.2 diatas merupakan data dengan satu indikator fitur *close* yang sudah diolah, terdapat kolom *date* yang berisi waktu tanggal harga saham, kolom x(n-5) sampai x(n-1) merupakan harga saham pada lima sampai satu hari sebelumnya. Untuk kolom x(n) merupakan harga saham pada hari tersebut dan untuk x(n+1) prediksi harga hari berikutnya. Lalu data displit menjadi dua variabel menjadi variabel X untuk data x(n-5) sampai (x(n) dan variabel y untuk x(n+1).

Pada pengolahan data dengan menggunakan dua indikator dengan menggunakan fitur *open* dan *close* dimana ditambahkan variabel pada dataset yaitu variabel o(n-5) sampai o(n) untuk mendeskripsikan harga *open* dalam lima hari sebelumnya sampai hari tersebut dan x(n-5) sampai x(n+1) mendeskripsikan harga *close* dalam lima hari sebelumnya sampai satu hari kedepan. Lalu data displit menjadi dua variabel menjadi variabel X untuk data o(n-5) sampai x(n) dan variabel y untuk x(n+1) dimana variabel y untuk mencari harga saham dihari berikutnya.

Pada pengolahan data dengan menggunakan tiga indikator dengan menggunakan fitur *open, high* dan *close* dimana ditambahkan variabel pada dataset yaitu variabel o(n-5) sampai o(n) untuk mendeskripsikan harga *open* dalam lima hari sebelumnya sampai hari tersebut, variabel h(n-5) sampai h(n) untuk mendeskripsikan harga *high* dalam lima hari sebelumnya sampai hari tersebut dan x(n-5) sampai x(n+1) mendeskripsikan harga *close* dalam lima hari sebelumnya sampai satu hari kedepan. Lalu data displit menjadi dua variabel menjadi variabel

X untuk data o(n-5) sampai x(n) dan variabel y untuk x(n+1) dimana variabel y untuk mencari harga saham dihari berikutnya.

Pada pengolahan data dengan menggunakan empat indikator dengan menggunakan fitur *open, high, low* dan *close* dimana ditambahkan variabel pada dataset yaitu variabel o(n-5) sampai o(n) untuk mendeskripsikan harga *open* dalam lima hari sebelumnya sampai hari tersebut, variabel h(n-5) sampai h(n) untuk mendeskripsikan harga *high* dalam lima hari sebelumnya sampai hari tersebut, variabel l(n-5) sampai l(n) untuk mendeskripsikan harga *low* dalam lima hari sebelumnya sampai hari tersebut dan x(n-5) sampai x(n+1) mendeskripsikan harga *close* dalam lima hari sebelumnya sampai satu hari kedepan. Lalu data displit menjadi dua variabel menjadi variabel X untuk data o(n-5) sampai x(n) dan variabel y untuk x(n+1) dimana variabel y untuk mencari harga saham dihari berikutnya.

5.1.3 *Cleansing* Data

Pada tahap perancangan *machine learning* yang dilakukan dalam penelitian ini terdapat suatu kesalahan yang menyebabkan terjadinya kendala pada *machine learning*. Kendala yang terjadi yaitu dengan adanya kesalahan dalam membaca *dataset* sehingga ada beberapa *dataset* yang tidak terbaca oleh *machine learning* sehingga data bernilai NaN. Maka dari itu perlu dilakukannya proses *cleansing* data untuk memperbaiki data yang tidak terbaca. Berikut hasil data setelah dilakukan *cleansing* data.

Tabel 5. 3 Cleansing Data

	x(n-5)	x(n-4)	x(n-3)	x(n-2)	x(n-1)	x(n)	x(n+1)
count	2465.000000	2465.000000	2465.000000	2465.000000	2465.000000	2465.000000	2465.000000
mean	5852.318458	5854.21501	5856.212982	5858.058824	5859.924949	5861.750507	5863.677485
std	1266.057191	1265.96464	1266.111380	1266.002401	1265.888557	1265.738260	1265.619092
min	3325.000000	3325.000000	3325.000000	3325.000000	3325.000000	3325.000000	3325.000000

25%	4825.000000	4825.00000	4825.00000	4825.00000	4825.00000	4825.00000	4825.00000
50%	5750.000000	5750.00000	5750.00000	5750.00000	5750.00000	5750.00000	5750.00000
75%	6975.000000	6975.00000	6975.00000	6975.00000	7000.00000	7000.00000	7000.00000
max	9050.000000	9050.00000	9050.00000	9050.00000	9050.00000	9050.00000	9050.00000

5.1.4 Normalisasi Data

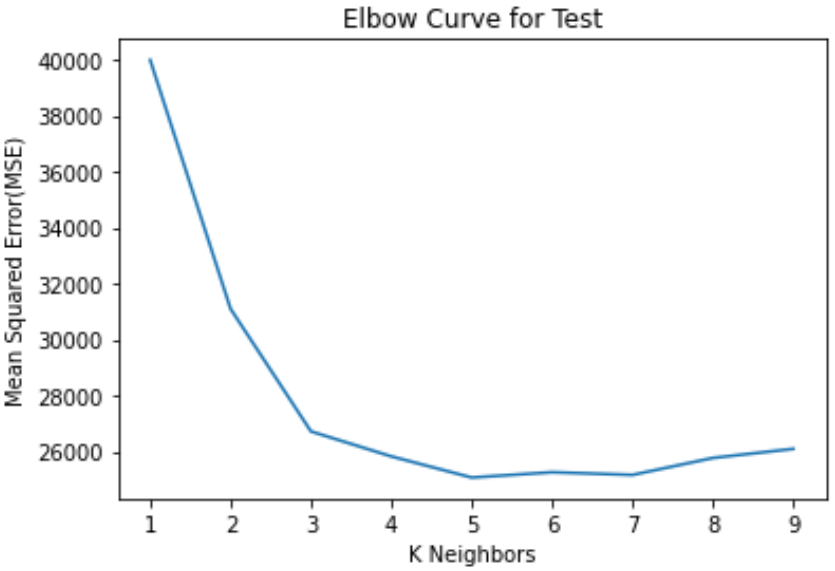
Pada perancangan *machine learning* dalam membangun model prediksi terdapat kendala yang terjadi pada *dataset* yang digunakan. Dimana *dataset* memiliki nilai yang kosong atau nilai null. Hal tersebut perlu dilakukannya normalisasi data untuk memperbaiki *dataset* yang memiliki nilai kosong atau null. Berikut hasil dari data yang sudah di normalisasi.

Tabel 5. 4 Normalisasi Data

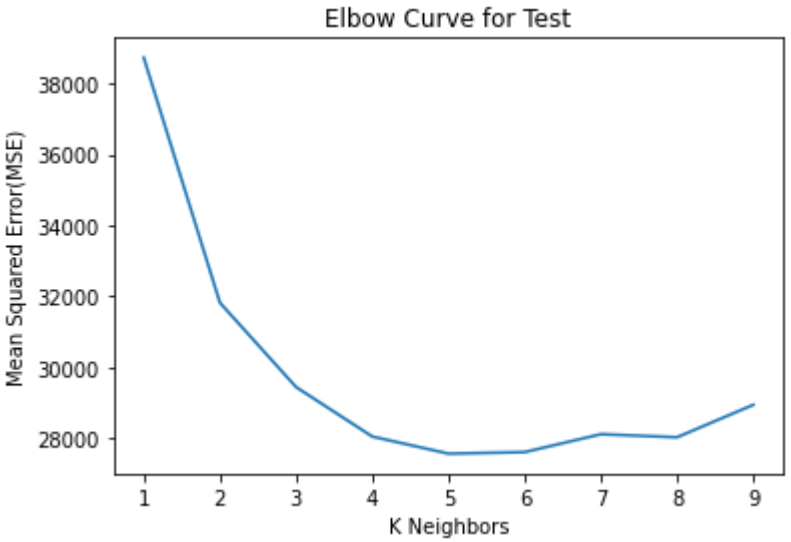
	x(n-5)	x(n-4)	x(n-3)	x(n-2)	x(n-1)	x(n)
0	0.039438	0.145947	0.027142	0.114036	0.055082	0.062715
1	1.417074	1.494469	1.608378	1.549379	1.572166	1.609851
2	-0.488168	-0.498998	-0.519458	-0.598754	-0.590902	-0.622726
3	-0.068038	-0.127666	0.007620	0.006629	-0.091733	0.043130
4	0.303240	0.302298	0.378528	0.260499	0.221472	0.180219

5.1.5 Pemodelan dengan Algoritma *K-Nearest Neighbors*

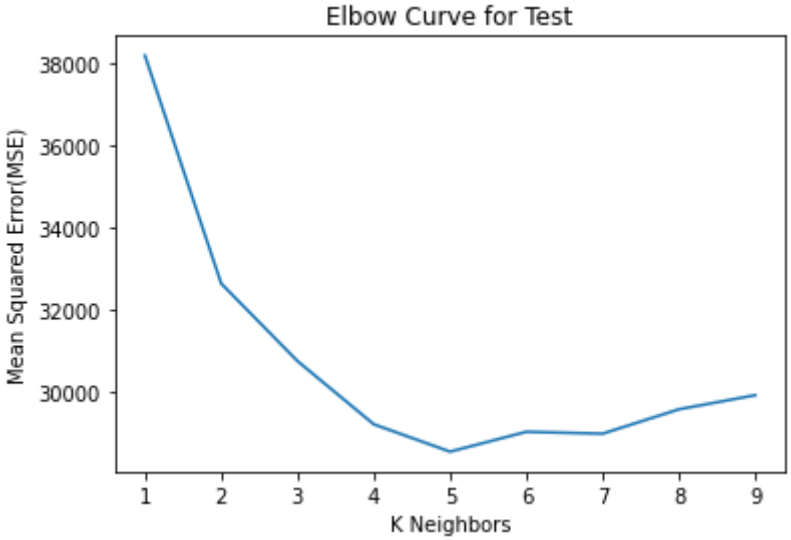
Langkah pertama yang dilakukan untuk pemodelan algoritma KNN yaitu dengan menentukan nilai K – optimal. Pencarian nilai K dilakukan dengan menghitung jarak antar Euclidean atau jarak antar tetangga menggunakan metode Elbow. Berikut hasil pencarian K – optimal dalam satu sampai empat indikator menggunaan metode Elbow.



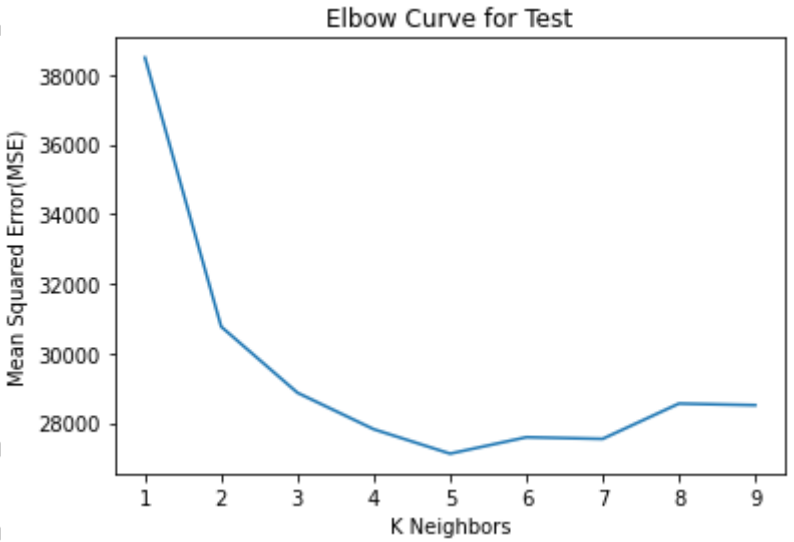
Gambar 5. 1 Metode Elbow (K - Optimal) satu indikator



Gambar 5. 2 Metode Elbow (K - Optimal) dua indikator



Gambar 5. 3 Metode Elbow (K - Optimal) tiga indikator



Gambar 5. 4 Metode Elbow (K - Optimal) empat indikator

Berdasarkan grafik Metode Elbow (K – Optimal), jumlah nilai K yang memilki nilai *mean squared erorr* (MSE) paling rendah merupakan nilai K yang paling optimal. Model menjadi lebih baik dengan adanya tingkat akurasi yang tinggi dan akurat apabila menggunakan nilai K – optimal. Berdasarkan pada empat grafik diatas, nilai K pada satu indikator, dua indikator, tiga indikator, dan empat indikator berada pada angka 5 semua untuk dijadikan K – optimal. Dimana dalam pemodelan menggunakan algoritma KNN dengan menggunakan satu sampai empat indikator memiliki nilai K – Optimal yang sama. Setelah mendapatkan nilai K – optimal, selanjutnya dilakukan pemodelan KNN untuk mencari nilai akurasi dan tingkat

erorr dengan menggunakan *root mean squared erorr* (RMSE) dengan hasil sebagai berikut.

Tabel 5. 5 Hasil Pemodelan KNN

	1 Indikator	2 Indikator	3 Indikator	4 Indikator
K – Optimal	5	5	5	5
RMSE	158	166	169	164
Akurasi	98.3%	98.1%	98.0%	98.1%

Berdasarkan pada tabel diatas, nilai K dengan jumlah indikator yang berbeda memiliki nilai yang sama yaitu berada pada angka 5 untuk dijadikan K – optimal. Tingkat RMSE yang dihasilkan dari ke empat dengan jumlah indikator yang berbeda ini menghasilkan nilai terendah dengan menggunakan satu indikator sebesar 158, dan nilai RMSE dengan dua, tiga dan empat indikator menghasilkan nilai masing – masing sebesar 166, 169, 164. Dimana nilai RMSE merupakan jarak selisih rata – rata *error* antara data asli dengan data prediksi dibuat dalam bentuk desimal agar mudah untuk dianalisa dalam melihat jarak *error* tersebut. Hasil dari akurasi KNN terbesar di dapatkan dengan menggunakan satu indikator sebesar 98.3%, dan hasil menggunakan dua, tiga, dan empat indikator masing – masing sebesar 98.1%, 98.0%, dan 98.1%. Dari hasil pemodelan menggunakan KNN berdasarkan tabel 5.5 diatas, prediksi dengan menggunakan satu indikator memiliki hasil yang paling baik.

5.1.6 Pemodelan dengan Algoritma *Support Vector Machine*

Dalam algoritma SVM dibutuhkan kernel dalam melakukam pemodelan SVM untuk mengubah data *non linier* menjadi data yang memiliki dimensi tinggi. Sehingga dengan adanya data tersebut dapat dibuatnya sebuah *hyperplane*. Pada penelitian ini menggunakan kernel *linear* untuk mengubah data. Parameter yang digunakan sebesar 10 agar mendapatkan nilai RMSE yang lebih kecil.

Tabel 5. 6 Hasil Pemodelan Algoritma SVM

	1 Indikator	2 Indikator	3 Indikator	4 Indikator
RMSE	152	153	152	149
Akurasi	98.4%	98.4%	98.4%	98.5%

Berdasarkan pada tabel diatas, pemodelan prediksi harga saham PT Bank Mandiri Tbk. menggunakan algoritma SVM dengan menggunakan empat indikator dengan jumlah indikator yang berbeda – beda mendapatkan hasil yang hampir sama. Nilai RMSE yang di dapatkan dengan menggunakan satu sampai tiga indikator yaitu sebesar 152 untuk satu indikator, 153 untuk dua indikator, dan 152 untuk tiga indikator dan 149 untuk empat indikator. Dimana nilai RMSE merupakan jarak selisih rata – rata *error* antara data asli dengan data prediksi dibuat dalam bentuk desimal agar mudah untuk dianalisa dalam melihat jarak *error* tersebut. Hasil akurasi yang didapatkan menggunakan satu sampai tiga indikator mendapatkan nilai akurasi yang sama yaitu sebesar 98.4% dan berbeda dengan menggunakan empat indikator yaitu sebesar 98.5%. Bedasarkan hasil tersebut, pada algoritma SVM di dapatkan hasil terbaik dengan menggunakan empat indikator dengan nilai RMSE 149 dan akurasi sebesar 98.5%.

5.1.7 Perbandingan Rasio

Dalam melakukan prediksi dengan membangun model *machine learning* ini dilakukan beberapa tahap dalam pengujian *dataset* dimana data dibagi ke dalam beberapa rasio untuk menguji tingkat *error* dengan hasil sebagai berikut.

Tabel 5. 7 Perbandingan RMSE Evaluasi Data pada KNN

Rasio	1 Indikator	2 Indikator	3 Indikator	4 Indikator
70 : 30	169	173	174	174
80 : 20	158	166	169	164
90 : 10	171	176	179	175

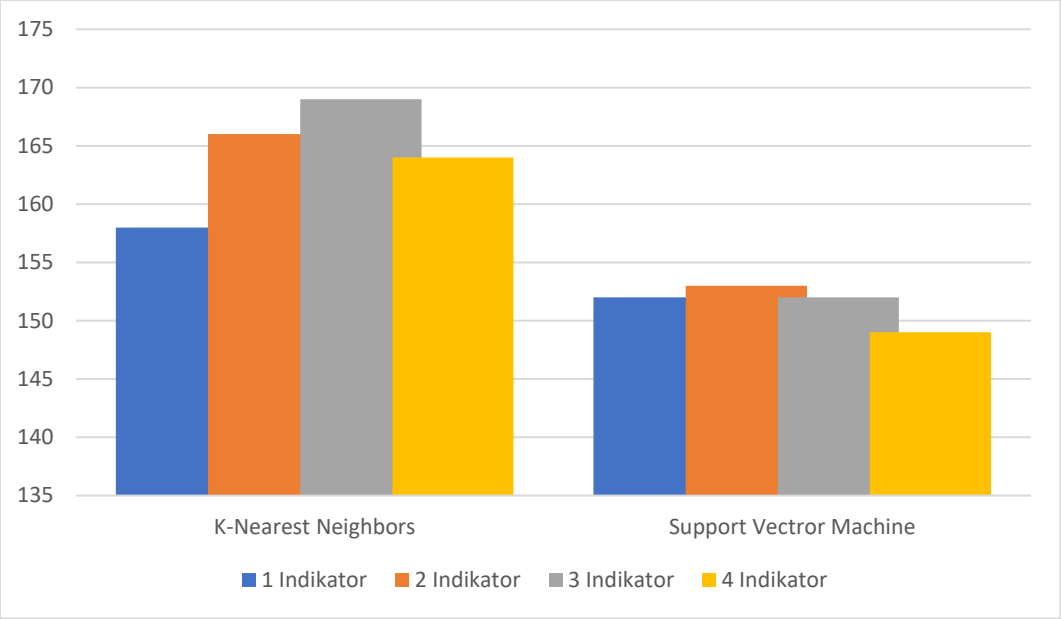
Tabel 5. 8 Perbandingan RMSE Evaluasi Data pada SVM

Rasio	1 Indikator	2 Indikator	3 Indikator	4 Indikator
70 : 30	161	161	158	157
80 : 20	152	153	152	149
90 : 10	179	168	170	165

Berdasarkan dua tabel diatas menjelaskan hasil dari perbandingan rasio data yang telah dibagi dengan tiga rasio yang berbeda pada setiap algoritma KNN dan SVM dengan jumlah indikator yang berbeda. Hasil yang ditunjukan pada tabel diatas kedua algoritma KNN dan SVM dengan seluruh jumlah indikator yang berbeda memiliki tingkat *error* RMSE dengan nilai paling rendah yaitu pada rasio perbandingan 80 : 20 dimana semakin kecil tingat *error* pada perancangan model prediksi akan menghasilkan hasil yang baik.

5.1.8 Hasil Prediksi

Setelah dilakukan pemodelan dalam memprediksi harga saham PT Bank Mandiri Tbk menggunakan kedua agoritma KNN dan SVM telah didapatkan hasil dari nilai RMSE dan akurasi dalam presentase. Berikut graifik hasil RMSE pada pemodelan algoritma KNN dan SVM.

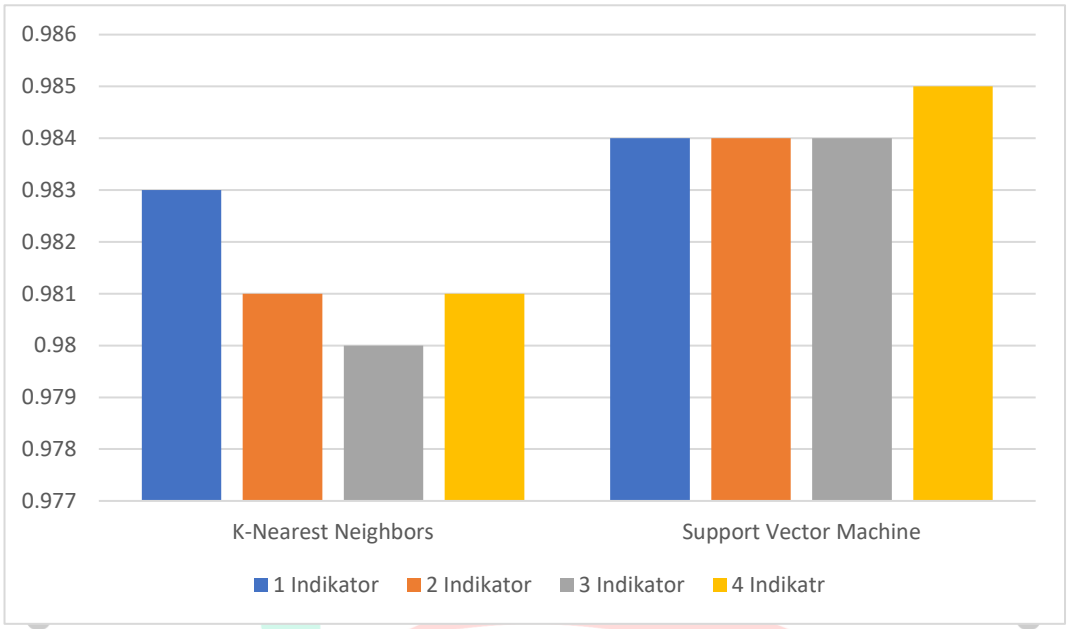


Gambar 5. 5 Perbandingan RMSE

Graifk diatas menunjukan hasil dari masing – masing pemodelan algoritma KNN dan SVM dengan empat jumlah indikator yang berbeda. Untuk algoritma KNN dengan satu indikator, dua indikator, tiga indikator, dan empat indikator memiliki nilai RMSE masing – masing sebesar 158, 166, 169, dan 164 dimana hasil dengan nilai RMSE *error* terkecil terdapat pada pemodelan dengan menggunakan satu indikator yaitu fitur *close*. Untuk algoritma SVM dengan satu indikator, dua indikator, tiga indikator, dan empat indikator memiliki nilai RMSE masing – masing sebesar 152, 153, 152 dan 149 dimana hasil dengan nilai RMSE *error* terkecil terdapat pada pemodelan dengan menggunakan empat indikator yaitu fitur *open*, *high*, *low*, dan *close*. Dimana, jika nilai RMSE semakin rendah maka pemodelan menghasilkan nilai akurasi yang semakin akurat dan tinggi.

Pemodelan pada algoritma KNN dengan menggunakan satu indikator, dua indikator, tiga indikator, dan empat indikator menghasilkan nilai akurasi masing – masing sebesar 98.3%, 98.1%, 98.3%, dan 98.1% dengan nilai akurasi terbesar menggunakan satu indikator yaitu fitur close. Sementara pemodelan algoritma SVM dengan menggunakan satu indikator, dua indikator, tiga indikator, dan empat indikator menghasilkan akurasi masing – masing sebesar 98.4%, 98.4%, 98.4% dan

98.5% dengan nilai akurasi terbesar menggunakan empat indikator yaitu fitur *open*, *high*, *low*, dan *close*. Berikut grafik perbandingan akurasi algoritma KNN dan SVM.



Gambar 5. 6 Perbandingan Akurasi

Selanjutnya pada penelitian ini dilakukan prediksi harga saham PT Bank Mandiri Tbk selama empat hari kedepan dengan menggunakan pemodelan *machine learning* dengan menggunakan algoritma KNN dan SVM. Berikut merupakan hasil yang telah dilakukan pada penelitian ini dalam memprediksi harga saham PT Bank Mandiri Tbk dalam lima hari kedepan.

Tabel 5. 9 Hasil Prediksi Harga Saham KNN

Tanggal	Prediksi KNN			
	1 Indikator	2 Indikator	3 Indikator	4 Indikator
8 Juni 2022	8245	8205	8205	8345
9 Juni 2022	8245	8205	8205	8205
10 Juni 2022	8245	8245	8245	8245
13 Juni 2022	8245	8245	8205	8245
14 Juni 2022	8245	8245	8205	8245

Tabel 5. 10 Hasil Prediksi Harga Saham SVM

Tanggal	Prediksi SVM			
	1 Indikator	2 Indikator	3 Indikator	4 Indikator
8 Juni 2022	8411	8401	8415	8404
9 Juni 2022	8511	8284	8438	8390
10 Juni 2022	8594	8439	8557	8507
13 Juni 2022	8681	8502	8541	8459
14 Juni 2022	8769	8539	8526	8463

Tabel diatas merupakan hasil prediksi harga saham yang dilakukan dalam lima hari kedepan. Predikisi dilakukan dengan menggunakan *dataset* harian dengan jumlah indikator yang berbeda – beda dimana pemodelan *machine learning* melakukan prediksi harga saham perhari. Prediksi harga saham selama lima hari ini dilakukan menggunakan data prediksi yang sudah dihasilkan pada satu hari untuk digunakan dalam memprediksi hari – hari berikutnya. Setiap prediksi harga pada masing – masing hari memiliki tingkat *erorr* selisih atau nilai RMSE berbeda – beda. Berdasarkan hasil prediksi dengan menggunakan jumlah indikator yang berbeda – beda mulai dari satu indikator, dua indikator, tiga indikator sampai empat indikator, hasil dari pemodelan prediksi harga saham PT Bank Mandiri Tbk memiliki nilai yang hampir sama dengan algortima KNN pada setiap jumlah indikator. Dimana dilihat dari pada tabel 5.9 hasil prediksi dengan KNN memiliki selisih nilai sebesar 40 – 100 pada setiap variasi indikator. Sedangkan hasil dari prediksi SVM memiliki nilai yang bervariatif pada setiap jumlah indikator yang berbeda. Dimana dilihat dari pada tabel 5.10 hasil prediksi dengan SVM memiliki selisih nilai sebesar 3 – 227 pada setiap variasi indikator.

5. 2 Pembahasan

Dalam pembahasan, dilakukan pemograman dalam membangun *machine learning* untuk pemodelan prediksi harga menggunakan algoritma KNN dan SVM. Kode program dibuat seusai dengan kebutuhan yang diperlukan dalam membangun pemodelan *machine learning* pada penelitian ini. Selanjutnya kode program diuji dengan melihat hasil yang dikeluarkan sudah sesuai dengan spesifikasi yang

dibutuhkan atau belum dengan menggunakan metode *Whitebox*. Berikut pengujian yang dilakukan.

Tabel 5. 11 Pengujian Kode Program

No	Kode Program	Hasil yang diharapkan	Hasil Pengujian
1	`from google.colab import drive` `drive.mount('/content/drive')`	Menghubungkan *machine learning* dengan Google Drive yang menyimpan data	Valid
2	`import pandas as pd` `import numpy as np` `import matplotlib.pyplot as plt` `import sklearn.metrics as metrics` `from sklearn.model_selection import train_test_split` `from sklearn.preprocessing import StandardScaler` `from sklearn.neighbors import KNeighborsRegressor as KNN_Reg` `from sklearn.metrics import mean_squared_error as mse` `from sklearn.metrics import mean_squared_error` `from sklearn.metrics import r2_score` `from sklearn.svm import SVR` `from math import sqrt`	Melakukan *import library* yang akan digunakan	Valid
3	`dataset = pd.read_excel('/content/drive/MyDrive/BMRIJK.xlsx', "Sheet1")`	Memanggil *dataset* dan memasukkannya	Valid
4	`#Cleansing Data` `dataset = dataset.dropna(axis=0)` `dataset.describe()`	Melakukan *Cleansing* data untuk menghapus data yang tidak terbaca	Valid
5	(a) `#split dataset` `X = dataset.iloc[:, 1:7]` `y = dataset.iloc[:, 7]` `X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, test_size=0.2)` (b)	Melakukan Split data menjadi dua variable x dan y pada 1	Valid

	`` #split dataset X = dataset.iloc[:, 1:13] y = dataset.iloc[:, 13] X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, test_size=0.2) ``` (c) ``` #split dataset X = dataset.iloc[:, 1:19] y = dataset.iloc[:, 19] X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, test_size=0.2) ``` (d) ``` #split dataset X = dataset.iloc[:, 1:25] y = dataset.iloc[:, 25] X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1, test_size=0.2) ```	indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	
6	``` #Normalisasi sc_X = StandardScaler() X_train = sc_X.fit_transform(X_train) X_test = sc_X.fit_transform(X_test) ```	Melakukan Normalisasi data untuk membersihkan data yang bernilai nol	Valid
7	``` def Elbow(K): test_mse =[] for i in K: model = KNN_Reg(n_neighbors=i) model.fit(X_train,y_train) tmp = model.predict(X_test) tmp = mse(tmp,y_test) test_mse.append(tmp) return test_mse K = range(1,10) test = Elbow(K) plt.plot(K,test) plt.xlabel('K Neighbors') plt.ylabel('Mean Squared Error(MSE)') plt.title('Elbow Curve for Test') ```	Menentukan nilai K – optimal pada KNN menggunakan metode Elbow	Valid

8	`#define model KNN` `model_knn= KNN_Reg(n_neighbors = 5)` `# training the model KNN` `model_knn.fit(X_train, y_train)` `#Akurasi` `acc1 = model_knn.score(X_test,y_test)` `# test for prediction` `test_predict_knn = model_knn.predict(X_test)` `#RMSE` `rmse_knn = sqrt(mean_squared_error(test_predict_knn,y_test))` `print('RMSE K-NN: ', rmse_knn)` `print('Accuracy KNN: ', acc1*100,'%')`	Melakukan pemodelan KNN untuk menetukan akurasi dan RMSE	Valid
9	`# Model SVR` `SVR_model = SVR(kernel = 'linear', C=10)` `# Fit Model` `SVR_model.fit(X_train, y_train)` `# test for prediction` `test_predict_SVR = SVR_model.predict(X_test)` `#Akurasi` `acc1 = SVR_model.score(X_test,y_test)` `#MSE` `mse_SVR = mse(test_predict_SVR, y_test)` `#RMSE` `rmse_SVR = sqrt(mean_squared_error(test_predict_SVR,y_test))` `print('RMSE SVR: ', rmse_SVR)` `print('Accuracy SVR: ', acc1*100,'%')`	Melakukan pemodelan SVM dengan kernel untuk menentukan akurasi dan RMSE	Valid
10	(a) `#Prediksi 1 Hari Selanjutnya (KNN)` `xaK =np.array([[8200, 8500, 8200, 8200, 8150, 8300]])` `xaK` `#Scaling Data` `XaK = sc_X.transform(xaK)` `#Pemodelan Prediksi` `ya_KNN = model_knn.predict(XaK)` `#Prediksi tanggal 8 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 8 Juni')` `print ('-')`	Melakukan pemodelan prediksi harga dengan KNN untuk hari pertama menggunakan 1 indikator (a), 2 indikator (b), 3 indikator	Valid

	(b) `#Prediksi 1 Hari Selanjutnya (KNN)` `xaK =np.array([[8200, 8275, 8325, 8275, 8200, 8200, 8200, 8500, 8200, 8200, 8150, 8300]])` `xaK` `#Scaling Data` `XaK = sc_X.transform(xaK)` `#Pemodelan Prediksi` `ya_KNN = model_knn.predict(XaK)` `#Prediksi tanggal 8 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 8 Juni')` `print ('-')` (c) `#Prediksi 1 Hari Selanjutnya` `xa =np.array([[8200, 8275, 8325, 8275, 8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8200, 8500, 8200, 8200, 8150, 8300]])` `xa` `#Scaling Data` `Xa = sc_X.transform(xa)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(Xa)` `ya_KNN = model_knn.predict(Xa)` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('SVR : ', ya_SVR)` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 8 Juni')` `print ('-')` (d) `#Prediksi 1 Hari Selanjutnya` `xa =np.array([[8200, 8275, 8325, 8275, 8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8000, 8225, 8050, 8200, 8025, 8150, 8200, 8500, 8200, 8200, 8150, 8300]])` `xa` `#Scaling Data` `Xa = sc_X.transform(xa)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(Xa)` `ya_KNN = model_knn.predict(Xa)` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('SVR : ', ya_SVR)` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 8 Juni')` `print ('-')`	(c), 4 indikator (d)	

11	(a) `#Prediksi 1 Hari Selanjutnya` `xaK1 =np.array([[8500, 8200, 8200, 8150, 8300, 8245]])` `xaK1` `#Scaling Data` `XaK1 = sc_X.transform(xaK1)` `#Pemodelan Prediksi` `ya_KNN1 = model_knn.predict(XaK1)` `#Prediksi tanggal 9 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 9 Juni')` `print ('KNN : ', ya_KNN1)` `print ('Harga Aktual 9 Juni')` `print ('-')` (b) `#Prediksi 1 Hari Selanjutnya` `xaK1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,` `8500, 8200, 8200, 8150, 8300, 8205]])` `xaK1` `#Scaling Data` `XaK1 = sc_X.transform(xaK1)` `#Pemodelan Prediksi` `ya_KNN1 = model_knn.predict(XaK1)` `#Prediksi tanggal 9 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 9 Juni')` `print ('KNN : ', ya_KNN1)` `print ('Harga Aktual 9 Juni')` `print ('-')` (c) `#Prediksi 1 Hari Selanjutnya` `xaK1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,` `8500, 8350, 8300, 8350, 8350, 8200,` `8500, 8200, 8200, 8150, 8300, 8205]])` `xaK1` `#Scaling Data` `XaK1 = sc_X.transform(xaK1)` `#Pemodelan Prediksi` `ya_KNN1 = model_knn.predict(XaK1)` `#Prediksi tanggal 9 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 9 Juni')` `print ('KNN : ', ya_KNN1)` `print ('Harga Aktual 9 Juni')` `print ('-')`	Melakukan pemodelan prediksi harga dengan KNN untuk hari kedua menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

	(d) `#Prediksi 1 Hari Selanjutnya` `xaK1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,` `8500, 8350, 8300, 8350, 8350, 8000,` `8225, 8050, 8200, 8025, 8150, 8200,` `8500, 8200, 8200, 8150, 8300, 8345]])` `xaK1` `#Scaling Data` `XaK1 = sc_X.transform(xaK1)` `#Pemodelan Prediksi` `ya_KNN1 = model_knn.predict(XaK1)` `#Prediksi tanggal 9 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 9 Juni')` `print ('KNN : ', ya_KNN1)` `print ('Harga Aktual 9 Juni')` `print ('-')`		
12	(a) `#Prediksi 1 Hari Selanjutnya` `xaK2 =np.array([[8200, 8200, 8150, 8300, 8245, 8245]])` `xaK2` `#Scaling Data` `XaK2 = sc_X.transform(xaK2)` `#Pemodelan Prediksi` `ya_KNN = model.predict(XaK2)` `#Prediksi tanggal 10 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 10 Juni')` `print ('-')` (b)	Melakukan pemodelan prediksi harga dengan KNN untuk hari ketiga menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

```
#Prediksi 1 Hari Selanjutnya
xaK2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,
                 8200, 8200, 8150, 8300, 8205, 8205]])
xaK2

#Scaling Data
XaK2 = sc_X.transform(xaK2)

#Pemodelan Prediksi
ya_KNN2 = model.predict(XaK2)

#Prediksi tanggal 10 Juni 2022

#Hasil Prediksi
print ('Prediksi 10 Juni')
print ('KNN : ', ya_KNN2)
print ('Harga Aktual 10 Juni')
print ('-')
```

(c)

```
#Prediksi 1 Hari Selanjutnya
xaK2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,
                 8350, 8300, 8350, 8350, 8200, 8500,
                 8200, 8200, 8150, 8300, 8205, 8205]])
xaK2

#Scaling Data
XaK2 = sc_X.transform(xaK2)

#Pemodelan Prediksi
ya_KNN2 = model.predict(XaK2)

#Prediksi tanggal 10 Juni 2022

#Hasil Prediksi
print ('Prediksi 10 Juni')
print ('KNN : ', ya_KNN2)
print ('Harga Aktual 10 Juni')
print ('-')
```

	(d) `#Prediksi 1 Hari Selanjutnya` `xaK2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,` `8350, 8300, 8350, 8350, 8000, 8225,` `8050, 8200, 8025, 8150, 8200, 8500,` `8200, 8200, 8150, 8300, 8345, 8205]])` `xaK2` `#Scaling Data` `XaK2 = sc_X.transform(xaK2)` `#Pemodelan Prediksi` `ya_KNN2 = model.predict(XaK2)` `#Prediksi tanggal 10 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('KNN : ', ya_KNN2)` `print ('Harga Aktual 10 Juni')` `print ('-')`		
13	(a) `#Prediksi 1 Hari Selanjutnya` `xaK3 =np.array([[8200, 8150, 8300, 8245, 8245, 8245]])` `xaK3` `#Scaling Data` `XaK3 = sc_X.transform(xaK3)` `#Pemodelan Prediksi` `ya_KNN = model.predict(XaK3)` `#Prediksi tanggal 11 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 11 Juni')` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 11 Juni')` `print ('-')` (b)	Melakukan pemodelan prediksi harga dengan KNN untuk hari keempat menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

```
#Prediksi 1 Hari Selanjutnya
xaK3 =np.array([[8275, 8200, 8200, 8200, 8500, 8200,
                 8200, 8150, 8300, 8205, 8205, 8245]])
xaK3

#Scaling Data
XaK3 = sc_X.transform(xaK3)

#Pemodelan Prediksi
ya_KNN3 = model.predict(XaK3)

#Prediksi tanggal 11 Juni 2022

#Hasil Prediksi
print ('Prediksi 13 Juni')
print ('KNN : ', ya_KNN3)
print ('Harga Aktual 31 Juni')
print ('-')
```

(c)

```
#Prediksi 1 Hari Selanjutnya
xaK3 =np.array([[8275, 8200, 8200, 8200, 8500, 8350,
                 8300, 8350, 8350, 8200, 8500, 8200,
                 8200, 8150, 8300, 8205, 8205, 8245]])
xaK3

#Scaling Data
XaK3 = sc_X.transform(xaK3)

#Pemodelan Prediksi
ya_KNN3 = model.predict(XaK3)

#Prediksi tanggal 11 Juni 2022

#Hasil Prediksi
print ('Prediksi 13 Juni')
print ('KNN : ', ya_KNN3)
print ('Harga Aktual 31 Juni')
print ('-')
```

	(d) `#Prediksi 1 Hari Selanjutnya` `xaK3 =np.array([[8275, 8200, 8200, 8200, 8500, 8350,` `8300, 8350, 8350, 8000, 8225, 8050,` `8200, 8025, 8150, 8200, 8500, 8200,` `8200, 8150, 8300, 8345, 8205, 8245]])` `xaK3` `#Scaling Data` `XaK3 = sc_X.transform(xaK3)` `#Pemodelan Prediksi` `ya_KNN3 = model.predict(XaK3)` `#Prediksi tanggal 11 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 13 Juni')` `print ('KNN : ', ya_KNN3)` `print ('Harga Aktual 31 Juni')` `print ('-')`		
14	(a) `#Prediksi 1 Hari Selanjutnya` `xaK4 =np.array([[8150, 8300, 8245, 8245, 8245, 8245]])` `xaK4` `#Scaling Data` `XaK4 = sc_X.transform(xaK4)` `#Pemodelan Prediksi` `ya_KNN = model.predict(XaK4)` `#Prediksi tanggal 12 Juni 2022` `#Hasil Prediksi` `print ('Prediksi 12 Juni')` `print ('KNN : ', ya_KNN)` `print ('Harga Aktual 12 Juni')` `print ('-')` (b)	Melakukan pemodelan prediksi harga dengan KNN untuk hari kelima menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

```
#Prediksi 1 Hari Selanjutnya
xaK4 =np.array([[8200, 8200, 8200, 8500, 8200, 8200,
                 8150, 8300, 8205, 8205, 8245, 8245]])
xaK4

#Scaling Data
XaK4 = sc_X.transform(xaK4)

#Pemodelan Prediksi
ya_KNN4 = model.predict(XaK4)

#Prediksi tanggal 12 Juni 2022

#Hasil Prediksi
print ('Prediksi 14 Juni')
print ('KNN : ', ya_KNN4)
print ('Harga Aktual 14 Juni')
print ('-')
```

(c)

```
#Prediksi 1 Hari Selanjutnya
xaK4 =np.array([[8200, 8200, 8200, 8500, 8350, 8300,
                 8350, 8350, 8200, 8500, 8200, 8200,
                 8150, 8300, 8205, 8205, 8245, 8205]])
xaK4

#Scaling Data
XaK4 = sc_X.transform(xaK4)

#Pemodelan Prediksi
ya_KNN4 = model.predict(XaK4)

#Prediksi tanggal 14 Juni 2022

#Hasil Prediksi
print ('Prediksi 14 Juni')
print ('KNN : ', ya_KNN4)
print ('Harga Aktual 14 Juni')
print ('-')
```

(d)

```
#Prediksi 1 Hari Selanjutnya
xaK4 =np.array([[8200, 8200, 8200, 8500, 8350, 8300,
                 8350, 8350, 8000, 8225, 8050, 8200,
                 8025, 8150, 8200, 8500, 8200, 8200,
                 8150, 8300, 8345, 8205, 8245, 8245]])
xaK4

#Scaling Data
XaK4 = sc_X.transform(xaK4)

#Pemodelan Prediksi
ya_KNN4 = model.predict(XaK4)

#Prediksi tanggal 14 Juni 2022

#Hasil Prediksi
print ('Prediksi 14 Juni')
print ('KNN : ', ya_KNN4)
print ('Harga Aktual 14 Juni')
print ('-')
```

15	(a) (b)	Melakukan pemodelan prediksi harga dengan SVM untuk hari pertama menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

(a)

```
#Prediksi 1 Hari Selanjutnya
xaS =np.array([[8200, 8500, 8200, 8200, 8150, 8300]])
xaS

#Scaling Data
XaS = sc_X.transform(xaS)

#Pemodelan Prediksi
ya_SVR = SVR_model.predict(XaS)


#Hasil Prediksi
print ('Prediksi 8 Juni')
print ('SVR : ', ya_SVR)
print ('Harga Aktual 8 Juni')
print ('-')
```

(b)

```
#Prediksi 1 Hari Selanjutnya
xaS =np.array([[8200, 8275, 8325, 8275, 8200, 8200,
                8200, 8500, 8200, 8200, 8150, 8300]])
xaS

#Scaling Data
XaS = sc_X.transform(xaS)

#Pemodelan Prediksi
ya_SVR = SVR_model.predict(XaS)


#Hasil Prediksi
print ('Prediksi 8 Juni')
print ('SVR : ', ya_SVR)
print ('Harga Aktual 8 Juni')
print ('-')
```

	(c) `#Prediksi 1 Hari Selanjutnya` `xaS =np.array([[8200, 8275, 8325, 8275, 8200, 8200,` `8200, 8500, 8350, 8300, 8350, 8350,` `8200, 8500, 8200, 8200, 8150, 8300]])` `xaS` `#Scaling Data` `XaS = sc_X.transform(xaS)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS)` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 8 Juni')` `print ('-')` (d) `#Prediksi 1 Hari Selanjutnya` `xaS =np.array([[8200, 8275, 8325, 8275, 8200, 8200,` `8200, 8500, 8350, 8300, 8350, 8350,` `8000, 8225, 8050, 8200, 8025, 8150,` `8200, 8500, 8200, 8200, 8150, 8300]])` `xaS` `#Scaling Data` `XaS = sc_X.transform(xaS)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS)` `#Hasil Prediksi` `print ('Prediksi 8 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 8 Juni')` `print ('-')`		
16	(a) `#Prediksi 1 Hari Selanjutnya` `xaS1 =np.array([[8500, 8200, 8200, 8150, 8300, 8411]])` `xaS1` `#Scaling Data` `XaS1 = sc_X.transform(xaS1)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS1)` `#Hasil Prediksi` `print ('Prediksi 9 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 9 Juni')` `print ('-')`	Melakukan pemodelan prediksi harga dengan SVM untuk hari kedua menggunakan 1 indikator (a), 2 indikator	Valid

(b)

```
#Prediksi 1 Hari Selanjutnya
xaS1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,
                8500, 8200, 8200, 8150, 8300, 8041]])
xaS1

#Scaling Data
XaS1 = sc_X.transform(xaS1)

#Pemodelan Prediksi
ya_SVR = SVR_model.predict(XaS1)


#Hasil Prediksi
print ('Prediksi 9 Juni')
print ('SVR : ', ya_SVR)
print ('Harga Aktual 9 Juni')
print ('-')
```

(c)

```
#Prediksi 1 Hari Selanjutnya
xaS1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,
                8500, 8350, 8300, 8350, 8350, 8200,
                8500, 8200, 8200, 8150, 8300, 8415]])
xaS1

#Scaling Data
XaS1 = sc_X.transform(xaS1)

#Pemodelan Prediksi
ya_SVR = SVR_model.predict(XaS1)

#Hasil Prediksi
print ('Prediksi 9 Juni')
print ('SVR : ', ya_SVR)
print ('Harga Aktual 9 Juni')
print ('-')
```

(d)

```
#Prediksi 1 Hari Selanjutnya
xaS1 =np.array([[8275, 8325, 8275, 8200, 8200, 8200,
                8500, 8350, 8300, 8350, 8350, 8000,
                8225, 8050, 8200, 8025, 8150, 8200,
                8500, 8200, 8200, 8150, 8300, 8404]])
xaS1

#Scaling Data
XaS1 = sc_X.transform(xaS1)

#Pemodelan Prediksi
ya_SVR = SVR_model.predict(XaS1)


#Hasil Prediksi
print ('Prediksi 9 Juni')
print ('SVR : ', ya_SVR)
print ('Harga Aktual 9 Juni')
print ('-')
```

(b), 3 indikator (c), 4 indikator (d)

17	(a) `#Prediksi 1 Hari Selanjutnya` `xaS2 =np.array([[8200, 8200, 8150, 8300, 8411, 8511]])` `xaS2` `#Scaling Data` `XaS2 = sc_X.transform(xaS2)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS2)` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 10 Juni')` `print ('-')` (b) `#Prediksi 1 Hari Selanjutnya` `xaS2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,` `8200, 8200, 8150, 8300, 8041, 8284]])` `xaS2` `#Scaling Data` `XaS2 = sc_X.transform(xaS2)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS2)` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 10 Juni')` `print ('-')` (c) `#Prediksi 1 Hari Selanjutnya` `xaS2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,` `8350, 8300, 8350, 8350, 8200, 8500,` `8200, 8200, 8150, 8300, 8415, 8438]])` `xaS2` `#Scaling Data` `XaS2 = sc_X.transform(xaS2)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS2)` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 10 Juni')` `print ('-')`	Melakukan pemodelan prediksi harga dengan SVM untuk hari ketiga menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

	(d) `#Prediksi 1 Hari Selanjutnya` `xaS2 =np.array([[8325, 8275, 8200, 8200, 8200, 8500,` `8350, 8300, 8350, 8350, 8000, 8225,` `8050, 8200, 8025, 8150, 8200, 8500,` `8200, 8200, 8150, 8300, 8404, 8390]])` `xaS2` `#Scaling Data` `XaS2 = sc_X.transform(xaS2)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS2)` `#Hasil Prediksi` `print ('Prediksi 10 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 10 Juni')` `print ('-')`		
18	(a) `#Prediksi 1 Hari Selanjutnya` `xaS3 =np.array([[8200, 8150, 8300, 8411, 8511, 8594]])` `xaS3` `#Scaling Data` `XaS3 = sc_X.transform(xaS3)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS3)` `#Hasil Prediksi` `print ('Prediksi 13 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 13 Juni')` `print ('-')` (b) `#Prediksi 1 Hari Selanjutnya` `xaS3 =np.array([[8275, 8200, 8200, 8200, 8500, 8200,` `8200, 8150, 8300, 8041, 8284, 8439]])` `xaS3` `#Scaling Data` `XaS3 = sc_X.transform(xaS3)` `#Pemodelan Prediksi` `ya_SVR = SVR_model.predict(XaS3)` `#Hasil Prediksi` `print ('Prediksi 13 Juni')` `print ('SVR : ', ya_SVR)` `print ('Harga Aktual 13 Juni')` `print ('-')` (c)	Melakukan pemodelan prediksi harga dengan SVM untuk hari keempat menggunakan 1 indikator (a), 2 indikator (b), 3 indikator (c), 4 indikator (d)	Valid

	`` #Prediksi 1 Hari Selanjutnya xaS3 =np.array([[8275, 8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8200, 8500, 8200, 8200, 8150, 8300, 8415, 8438, 8557]]) xaS3 #Scaling Data XaS3 = sc_X.transform(xaS3) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS3) #Hasil Prediksi print ('Prediksi 13 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 13 Juni') print ('-') ``` (d) ``` #Prediksi 1 Hari Selanjutnya xaS3 =np.array([[8275, 8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8000, 8225, 8050, 8200, 8025, 8150, 8200, 8500, 8200, 8200, 8150, 8300, 8404, 8390, 8507]]) xaS3 #Scaling Data XaS3 = sc_X.transform(xaS3) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS3) #Hasil Prediksi print ('Prediksi 13 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 13 Juni') print ('-') ```		
19	(a) ``` #Prediksi 1 Hari Selanjutnya xaS4 =np.array([[8150, 8300, 8411, 8511, 8594, 8681]]) xaS4 #Scaling Data XaS4 = sc_X.transform(xaS4) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS4) #Hasil Prediksi print ('Prediksi 14 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 14 Juni') print ('-') ``` (b)	Melakukan pemodelan prediksi harga dengan SVM untuk hari kelima menggunakan 1 indikator (a), 2 indikator (b), 3 indikator	Valid

	`` #Prediksi 1 Hari Selanjutnya xaS4 =np.array([[8200, 8200, 8200, 8500, 8200, 8200, 8150, 8300, 8041, 8284, 8439, 8502]]) xaS4 #Scaling Data XaS4 = sc_X.transform(xaS4) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS4) #Hasil Prediksi print ('Prediksi 14 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 14 Juni') print ('-') ``` (c) ``` #Prediksi 1 Hari Selanjutnya xaS4 =np.array([[8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8200, 8500, 8200, 8200, 8150, 8300, 8415, 8438, 8557, 8541]]) xaS4 #Scaling Data XaS4 = sc_X.transform(xaS4) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS4) #Hasil Prediksi print ('Prediksi 14 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 14 Juni') print ('-') ``` (d) ``` #Prediksi 1 Hari Selanjutnya xaS4 =np.array([[8200, 8200, 8200, 8500, 8350, 8300, 8350, 8350, 8000, 8225, 8050, 8200, 8025, 8150, 8200, 8500, 8200, 8200, 8150, 8300, 8404, 8390, 8507, 8459]]) xaS4 #Scaling Data XaS4 = sc_X.transform(xaS4) #Pemodelan Prediksi ya_SVR = SVR_model.predict(XaS4) #Hasil Prediksi print ('Prediksi 14 Juni') print ('SVR : ', ya_SVR) print ('Harga Aktual 14 Juni') print ('-') ```	(c), 4 indikator (d)	