BAB V
HASIL DAN PEMBAHASAN

5.1 Pembahasan Hasil dan Uraian dari Perancangan

Bab ini akan menjelaskan pembahasan dari hasil pengujian yang sudah peneliti lakukan. Hasil dari pengujian tersebut akan peneliti jelaskan secara detail sebagai berikut.

5.1.1 *Import Library*

Sebelum memulai sebuah *machine learning* maka dibutuhkannya sebuah *library* yang akan membantu peneliti. Maka dari itu diperlukan *import library machine learning* yang akan dipakai oleh peneliti. *Library* yang akan dipakai nantinya yaitu pandas, numpy (proses komputasi numerik), matplotlib (visualisasikan data kedalam sumbu x dan y), dan sklearn (membagi data *train* dan data *test*). Berikut gambar dari *import library* yang peneliti pakai.

```
[1]  #Import libraries
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
```

Gambar 5. 1. Import Library

5.1.2 *Import Dataset*

Tahapan selanjutnya yaitu melakukan *import dataset*. Dataset yang sudah di dapatkan kemudian di *import* atau di *load* ke dalam *program.* Dataset tersebut yang akan dipakai untuk melakukan prediksi harga *ethereum.* Berikut hasil dari *import dataset.*

	Date	High(d-5)	High(d-4)	High(d-3)	High(d-2)	High(d-1)	High(d)	High(d+1)	Low(d-5)	Low(d-4)	Low(d-3)	Low(d-2)	Low(d-1)	Low(d)	Low(d+1)
0	2017-11-14	329.451996	324.717987	319.453003	319.153015	328.415009	340.177002	340.911987	307.056000	294.541992	298.191986	298.513000	307.024994	316.763000	329.812988
1	2017-11-15	324.717987	319.453003	319.153015	328.415009	340.177002	340.911987	336.158997	294.541992	298.191986	298.513000	307.024994	316.763000	329.812988	323.605988
2	2017-11-16	319.453003	319.153015	328.415009	340.177002	340.911987	336.158997	334.963989	298.191986	298.513000	307.024994	316.763000	329.812988	323.605988	327.523010
3	2017-11-17	319.153015	328.415009	340.177002	340.911987	336.158997	334.963989	349.615997	298.513000	307.024994	316.763000	329.812988	323.605988	327.523010	327.687012
4	2017-11-18	328.415009	340.177002	340.911987	336.158997	334.963989	349.615997	371.290985	307.024994	316.763000	329.812988	323.605988	327.523010	327.687012	344.739990
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1536	2022-01-28	2542.144775	2537.208496	2498.507080	2705.784180	2510.451660	2548.778564	2627.609375	2381.515137	2172.301270	2359.384766	2417.683105	2328.452148	2367.833740	2526.989746
1537	2022-01-29	2537.208496	2498.507080	2705.784180	2510.451660	2548.778564	2627.609375	2631.408447	2172.301270	2359.384766	2417.683105	2328.452148	2367.833740	2526.989746	2550.460938
1538	2022-01-30	2498.507080	2705.784180	2510.451660	2548.778564	2627.609375	2631.408447	2697.735352	2359.384766	2417.683105	2328.452148	2367.833740	2526.989746	2550.460938	2489.072266
1539	2022-01-31	2705.784180	2510.451660	2548.778564	2627.609375	2631.408447	2697.735352	2802.315430	2417.683105	2328.452148	2367.833740	2526.989746	2550.460938	2489.072266	2682.621826
1540	2022-02-01	2510.451660	2548.778564	2627.609375	2631.408447	2697.735352	2802.315430	0.000000	2328.452148	2367.833740	2526.989746	2550.460938	2489.072266	2682.621826	0.000000

1541 rows × 15 columns

Gambar 5. 2. Import Dataset

Pada Gambar 5.2 data didapatkan dari website *yahoo.finance.com.* Kemudian data tersebut diolah kedalam data *time series.* Peneliti memfokuskan terhadap data *high* dan *low*. Kemudian data tersebut diolah kedalam data *time series* (d-5) samapai (d). Data tersebut merupakan harga 5 hari sebelumnya pada Ethereum.

5.1.3 *Pre-Processing Data*

Dataset yang sudah diimport ke dalam program. kemudian *dataset* tersebut akan dinormalisasikan terlebih dahulu sebelum dimasukkan ke dalam pemodelan *machine learning*. Peneliti menggunakan *MinMax scaler* untuk menormalisasikan data *history* Ethereum yang peneliti pakai. Berikut hasil data yang sudah di normalisasikan.

	High(d-5)	High(d-4)	High(d-3)	High(d-2)	High(d-1)	High(d)	High(d+1)	Low(d-5)	Low(d-4)	Low(d-3)	Low(d-2)	Low(d-1)	Low(d)	Low(d+1)
0	0.057075	0.055968	0.054737	0.054667	0.056832	0.059582	0.059754	0.060552	0.057173	0.058159	0.058245	0.060544	0.063174	0.066698
1	0.055968	0.054737	0.054667	0.056832	0.059582	0.059754	0.058643	0.057173	0.058159	0.058245	0.060544	0.063174	0.066698	0.065022
2	0.054737	0.054667	0.056832	0.059582	0.059754	0.058643	0.058364	0.058159	0.058245	0.060544	0.063174	0.066698	0.065022	0.066079
3	0.054667	0.056832	0.059582	0.059754	0.058643	0.058364	0.061789	0.058245	0.060544	0.063174	0.066698	0.065022	0.066079	0.066124
4	0.056832	0.059582	0.059754	0.058643	0.058364	0.061789	0.066857	0.060544	0.063174	0.066698	0.065022	0.066079	0.066124	0.070729

Gambar 5. 3. Data Normalisasi

5.1.4 Split Data

Pemisahan data adalah pendekatan yang umum digunakan untuk validasi model, dimana membagi set data yang diberikan menjadi dua set terpisah: pelatihan dan pengujian. Rasio yang umum digunakan adalah 80:20, yang berarti 80% data untuk pelatihan dan 20% untuk pengujian. Rasio lain seperti 70:30, 60:40, dan bahkan 50:50 juga digunakan dalam praktik(Joseph, 2022). Dataset yang sudah didapatkan akan diuji ke dalam pemodelan *K-Nearest Neighbors* dan *Support*

vector Machine. Data tersebut akan di bagi menjadi 3 rasio yang berbeda 90:10, 80:20, dan 70:30. Data yang digunakan sebanyak 1541 data yang diambil dari rentang waktu 5 tahun 2017 sampai 2022. Kemudian data dibagi menjadi dua variable yaitu variable X dan y yang dimana variabel(Joseph, 2022). X adalah feature yang akan digunakan dan variabel y adalah data prediksi. Variabel pada X merupakan data *High*(d-5) sampai *High*(d) serta *Low*(d-5) sampai *Low*(d). kemudian variabel y adalah data predictor yang dimana variable tersebut adalah data *High*(d+1) dan *Low*(d+1). Berikut hasil dari perbandingan pembagian data yang diuji kedalam pemodelan *K-Nearest Neighbors* dan *Support vector Machine.*

Tabel 5. 1 Pembagian Data dengan Target High d+1 pada KNN

Rasio	K-Optimal	RMSE	R^2
70:30	3	0.016	0.992
80:20	5	0.016	0.991
90:10	4	0.012	0.993

Sumber: Data diolah peneliti

Tabel 5. 2 Pembagian Data dengan Target Low d+1 pada KNN

Rasio	K-Optimal	RMSE	R^2
70:30	3	0.019	0.989
80:20	6	0.018	0.989
90:10	5	0.015	0.991

Sumber: Data diolah peneliti

Tabel 5. 3 Pembagian Data dengan Target High d+1 pada SVM

Rasio	C	RMSE	R^2
70:30	10	0.067	0.862
80:20	10	0.066	0.853
90:10	10	0.059	0.852

Sumber: Data diolah peneliti

Tabel 5. 4 Pembagian Data dengan Target Low d+1 pada SVM

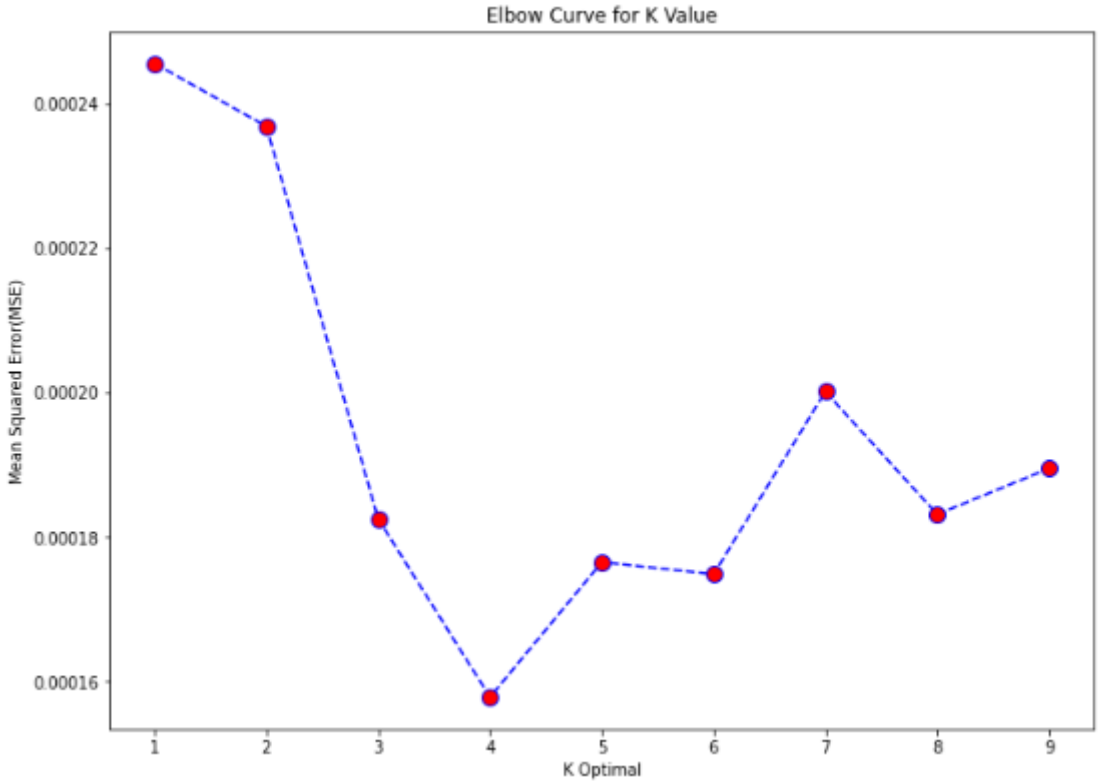
Rasio	C	RMSE	R^2
70:30	10	0,019	0.989
80:20	10	0,019	0.988
90:10	10	0,021	0.982

Sumber: Data diolah peneliti

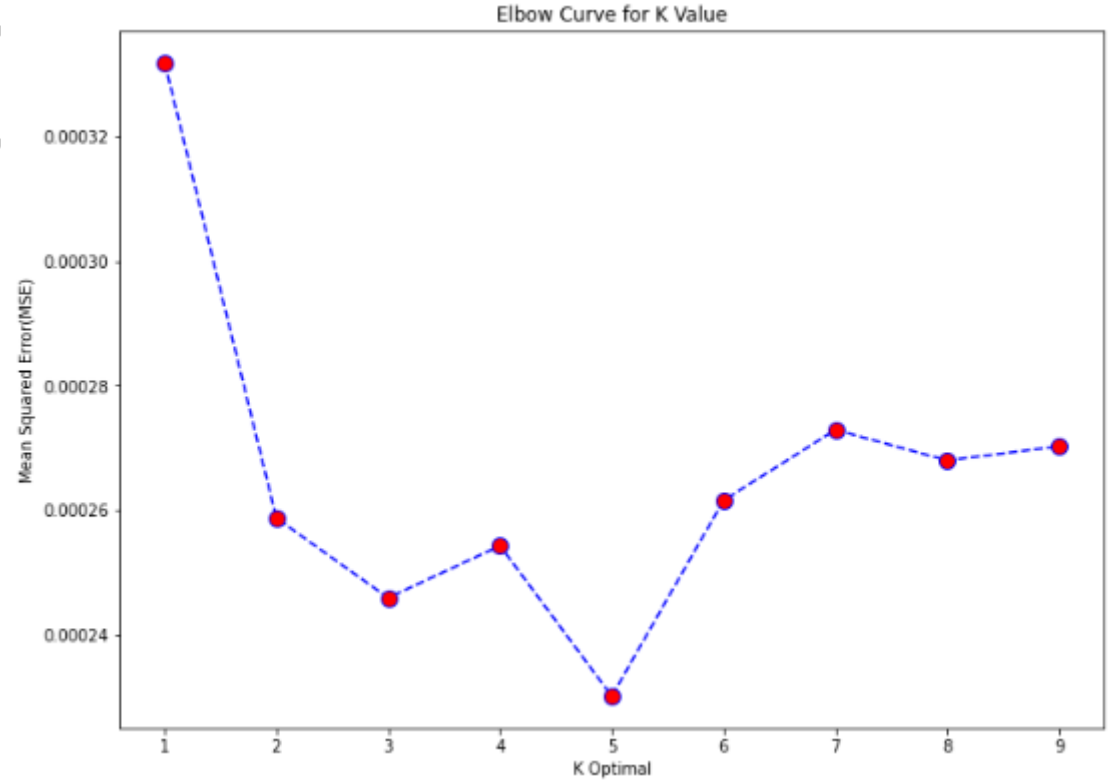
Berdasarkan tabel di atas maka data yang akan dipakai yaitu data yang memiliki RMSE terendah dan R^2 yang tertinggi. RMSE terendah merupakan hasil yang terbaik disebabkan jumlah data yang eror juga termasuk sedikit. R^2 tertinggi merupakan hasil yang baik sebab variabel independen dan dependen memiliki keterkaitan. Maka dapat disimpulkan pembagian data yang terbaik yaitu dengan rasio 90:10.

5.1.5 Pemodelan dengan K-Nearest Neighbors

Setelah pembagian data telah dilakukan kemudian melakukan pemodelan prediksi harga Ethereum menggunakan *K-Nearest Neighbors.* Untuk mendapatkan nilai K optimal maka harus menghitung nilai euclidean atau metode elbow. Kegunaan mencari nilai euclidean yaitu mendapatkan angka jarak antar tetangga. Berikut hasil dari K optimal yang di dapatkan dengan data split 90:10.

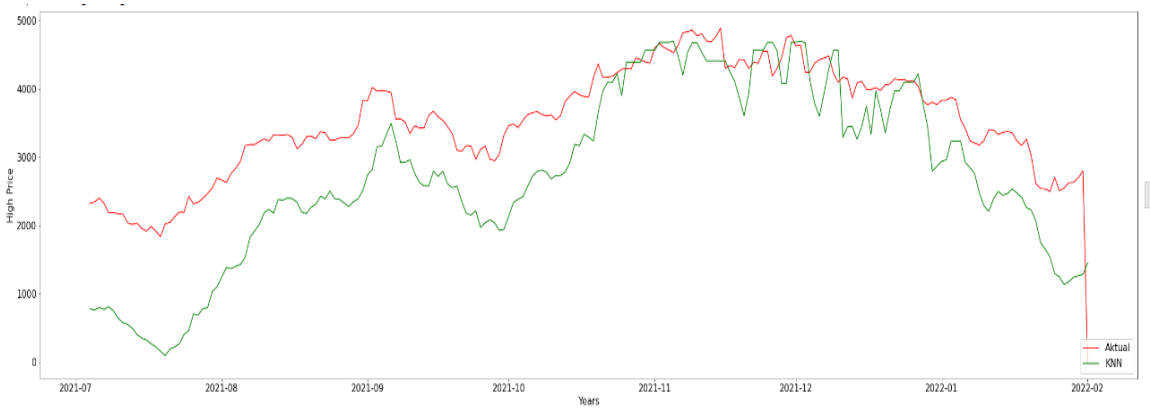


Gambar 5. 4. Nilai K Optimal dengan target high d+1.



Gambar 5. 5. Nilai K Optimal dengan target low d+1

Berdasarkan gambar 5.4 merupakan nilai k optimal dengan target *high* d+1 sebesar 7. Sedangkan nilai k optimal dengan target low d+1 sebesar 6. Nilai k optimal tersebut menggunakan rasio yang sudah dipakai yaitu 90:10 maka k optimal tersebut memiliki nilai MSE terkecil. Berikut tersaji diagram perbandingan antara hasil prediksi *K-Nearest Neighbors* dengan data uji yang dipakai.



Gambar 5. 6. Grafik perbandingan k-NN dengan data uji High(d+1)(time-series, k =4, rasio data latih 90:10, data uji = 155 data)

Gambar 5. 7. Grafik perbandingan k-NN dengan data uji High(d+1)(time-series, k =6, rasio data latih 80:20, data uji = 309 data)

Gambar 5.8. Grafik perbandingan k-NN dengan data uji High(d+1)(time-series, k =3, rasio data latih 70:30, data uji = 463 data)

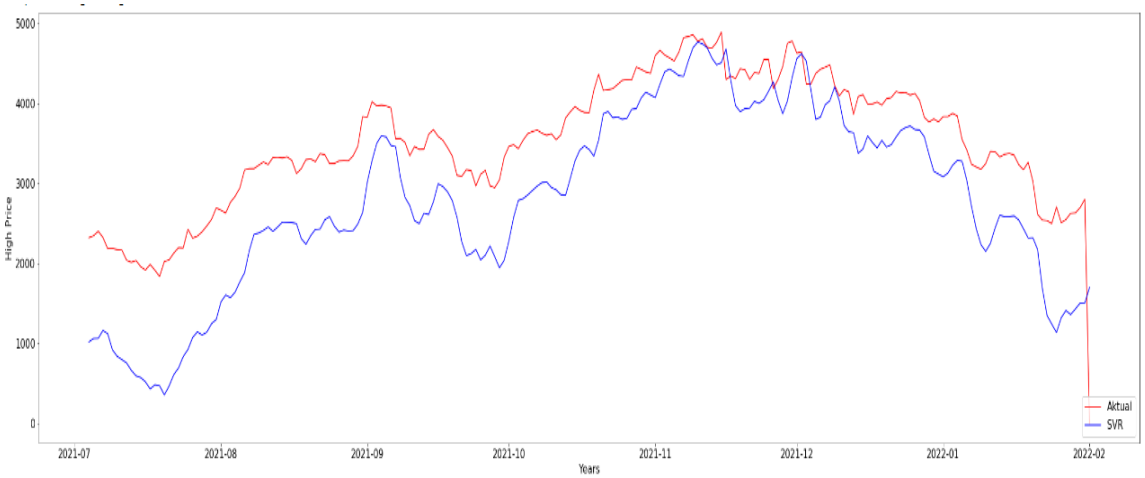
Gambar 5. 9. Grafik perbandingan k-NN dengan data uji Low(d+1)(time-series, k =5, rasio data latih 90:10, data uji = 155 data)

Gambar 5. 10.Grafik perbandingan k-NN dengan data uji Low(d+1)(time-series, k =6, rasio data latih 80:20, data uji = 309 data)

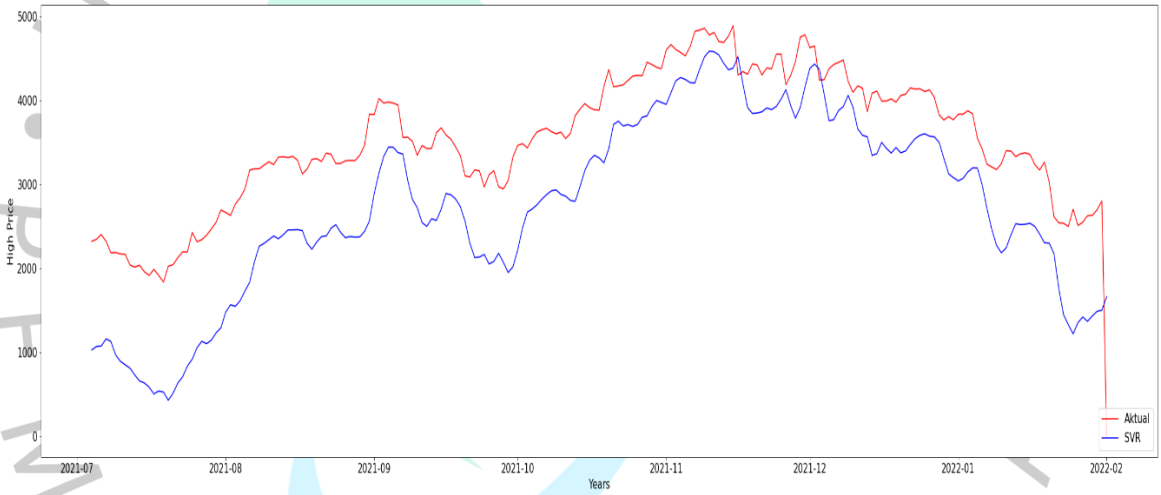
Gambar 5. 11.Grafik perbandingan k-NN dengan data uji Low(d+1)(time-series, k =3, rasio data latih 70:30, data uji = 463 data)

5.1.6 Pemodelan dengan Support Vector Regression

Untuk melakukan pemodelan *Support Vector Regression* dibutuhkan sebuah kernel. Kernel tersebut bertujuan untuk mengubah data yang awalnya non linier akan diubah menjadi data yang memiliki dimensi tinggi. Sehingga data yang memiliki dimensi tinggi dapat dibuatnya sebuah *hyperplane*. Penelitian ini menggunakan kernel *linear* sebagai pembantu mengubah data menjadi dimensi yg tinggi. Parameter yang dipakai sebesar 10 disebabkan parameter sebesar 10 akan mendapatkan hasil RMSE lebih kecil dan hasil R^2 yang lebih besar. Berikut merupakan hasil pemodelan *Support Vector Regression* yang dibandingkan dengan data aktual dalam bentuk grafik.



Gambar 5. 12.Grafik perbandingan SVR High(d+1) dengan data uji (time-series, rasio data latih 90:10 data uji = 155 data)



Gambar 5. 13.Grafik perbandingan SVR High(d+1) dengan data uji (time-series, rasio data latih 80:20 data uji = 309 data)

Gambar 5. 14.Grafik perbandingan SVR High(d+1) dengan data uji (time-series, rasio data latih 70:30 data uji = 463 data)

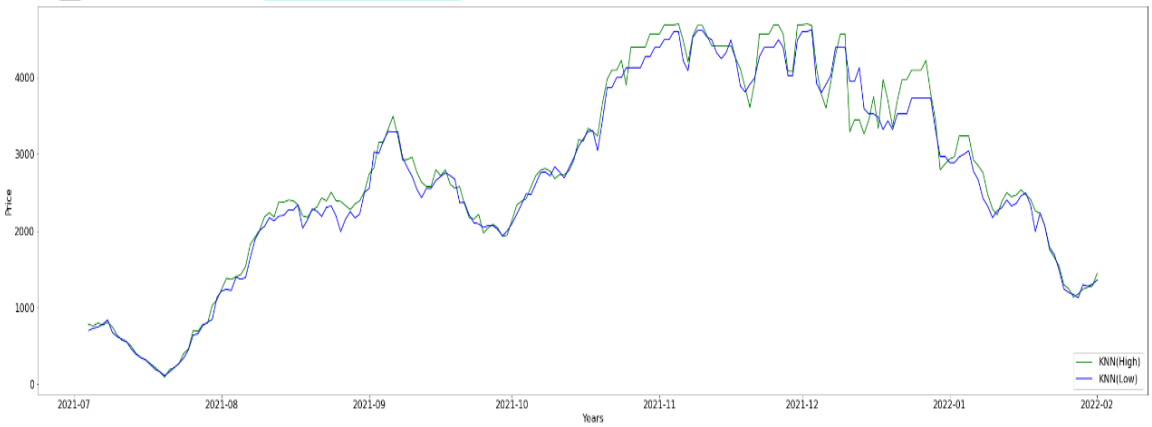
Gambar 5. 15.Grafik perbandingan SVR Low(d+1) dengan data uji (time-series, rasio data latih 90:10 data uji = 155 data)

Gambar 5.16. Grafik perbandingan SVR Low(d+1) dengan data uji (time-series, rasio data latih 80:20 data uji = 309 data)

Gambar 5. 16.Grafik perbandingan SVR Low(d+1) dengan data uji (time-series, rasio data latih 70:30 data uji = 463 data)

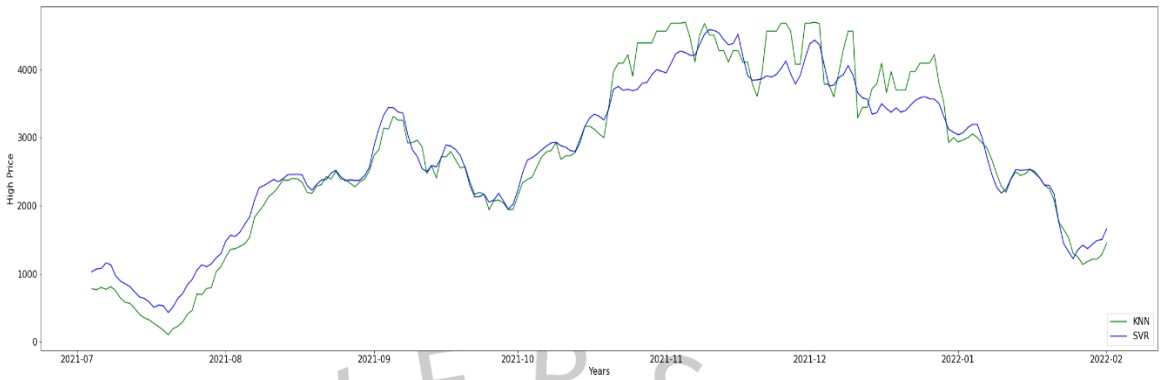
5.1.7 Perbandingan Hasil Algoritma

Pemodelan prediksi harga Ethereum yang diperoleh menggunakan algoritma K-Nearest Neighbors dan Support Vector Machine dibandingkan pada suatu grafik. Perbandingan hasil ini dapat memperlihatkan perbedaan model yang dihasilkan pada setiap algoritma. Perbedaan hasil di setiap algoritma merupakan suatu kewajaran dalam pemrosesan data dengan machine learning. Grafik yang akan dibandingkan yaitu membandingkan data uji dengan hasil prediksi algoritma K-Nearest Neighbors dan Support Vector Machine pada High(d+1) dan Low(d+1). Berikut merupakan hasil perbandingan dalam bentuk grafik.



Gambar 5. 17.Grafik perbandingan KNN High(d+1) dengan KNN Low(d+1) (time-series, rasio data latih 90:10, data uji = 155 data)

Gambar 5. 18.Grafik perbandingan SVR High(d+1) dengan SVR Low(d+1) (time-series, rasio data latih 90:10, data uji = 155 data)



Gambar 5. 19.Grafik perbandingan KNN High(d+1), SVR High(d+1) dan Data Uji (time-series, k =5, rasio data latih 80:20, data uji = 309)

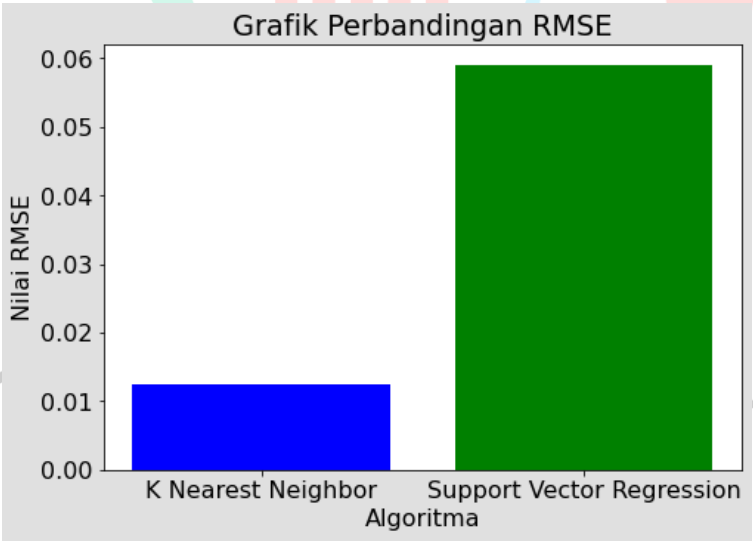
Gambar 5. 20.Grafik perbandingan KNN Low(d+1), SVR Low(d+1) dan Data Uji (time-series, k =6, rasio data latih 80:20, data uji = 309)

Gambar 5. 21Grafik perbandingan KNN High(d+1), SVR High(d+1) dan Data Uji (time-series, k =3, rasio data latih 70:30, data uji = 463 data)

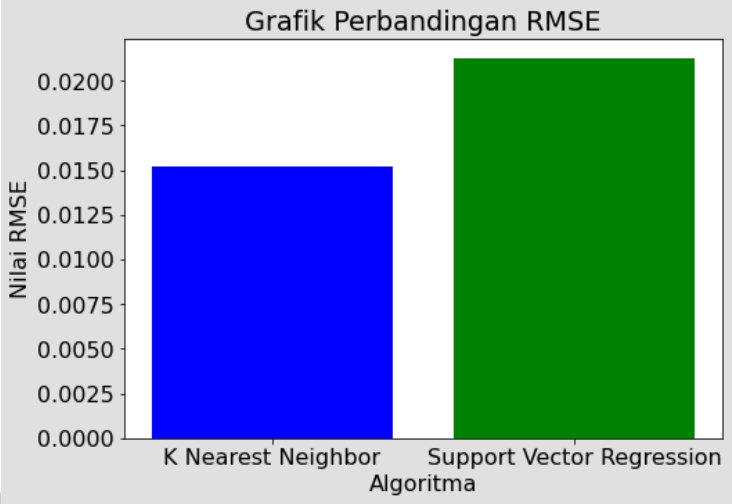
Gambar 5. 22Grafik perbandingan KNN Low(d+1), SVR Low(d+1) dan Data Uji (time-series, k =3, rasio data latih 70:30, data uji = 463 data)

5.2.8. Perbandingan Hasil Akurasi

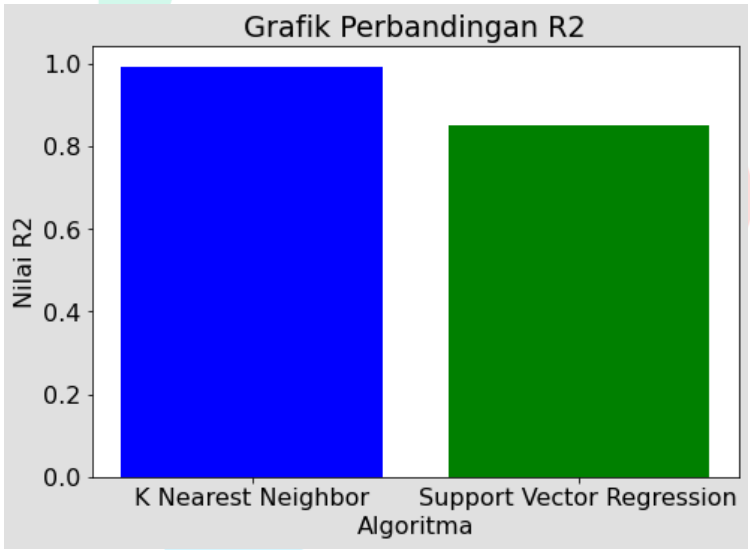
Pemodelan prediksi harga Ethereum yang diperoleh menggunakan algoritma K-Nearest Neighbors dan Support Vector Machine akan dihitung akurasinya. Akurasi yang akan dipakai yaitu RMSE dan R2. Setiap akurasi tersebut akan dibandingkan kedalam bentuk diagram batang. Berikut hasil pengujian dari akurasi kedua algoritma.



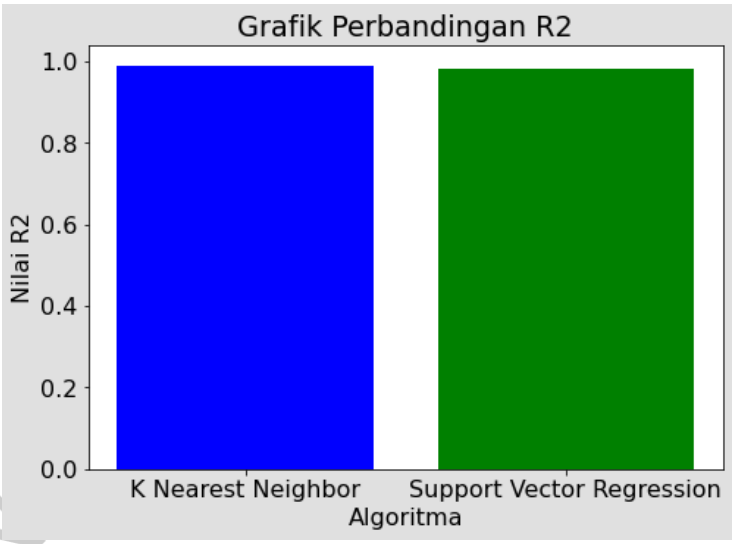
Gambar 5. 23.Grafik perbandingan akurasi RMSE KNN High(d+1) dan SVR High(d+1) (time-series, k =4, rasio data latih 90:10, data uji = 155)



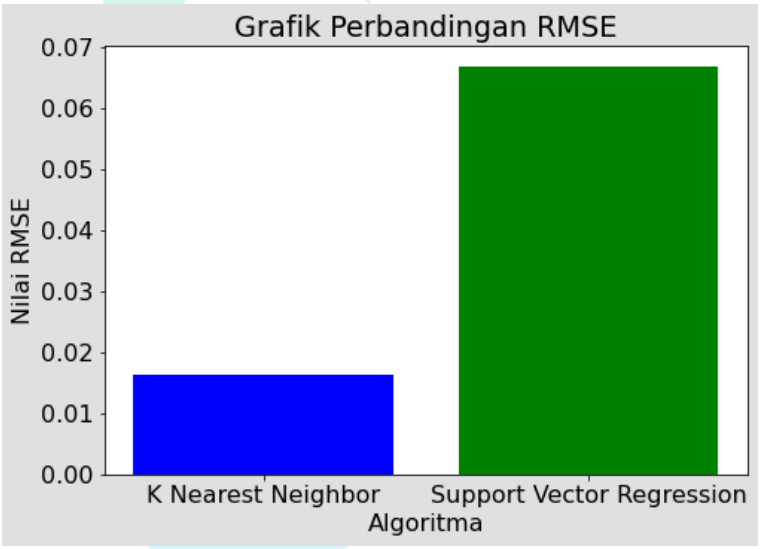
Gambar 5. 24.Grafik perbandingan akurasi RMSE KNN Low(d+1) dan SVR Low(d+1) (time-series, rasio data latih 90:10, data uji = 155)



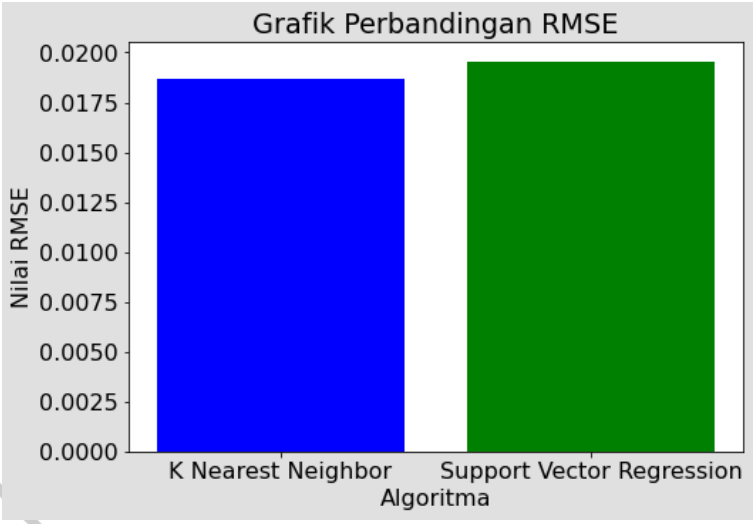
Gambar 5. 25.Grafik perbandingan akurasi R2 KNN High(d+1) dan SVR High(d+1) (time-series, k =5, rasio data latih 90:10, data uji = 155)



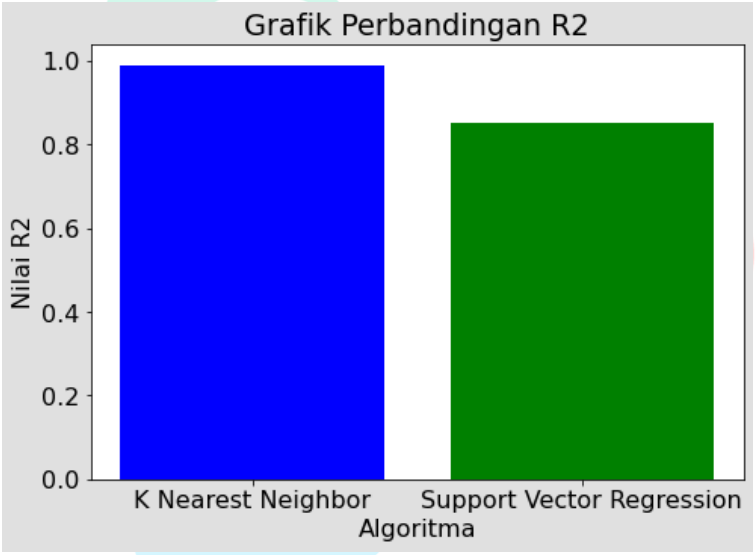
Gambar 5. 26.Grafik perbandingan akurasi R2 KNN Low(d+1) dan SVR Low(d+1) (time-series, rasio data latih 90:10, data uji = 155)



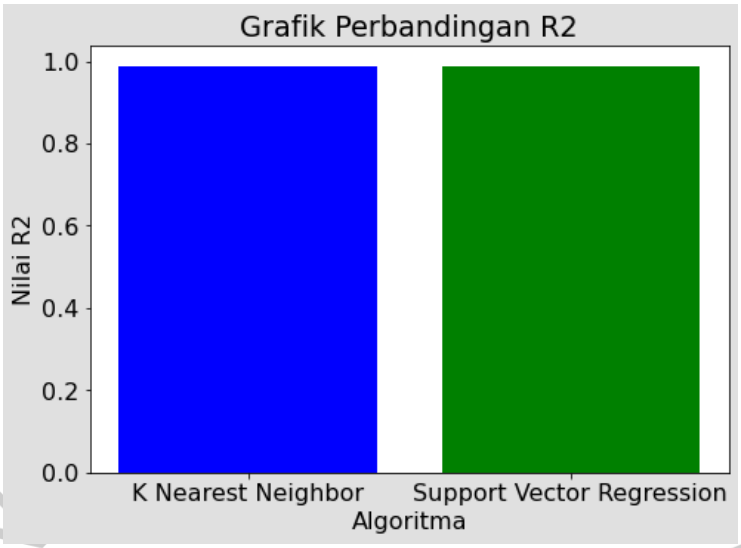
Gambar 5. 27.Grafik perbandingan akurasi RMSE KNN High(d+1) dan SVR High(d+1) (time-series, k =5, rasio data latih 80:20, data uji = 309)



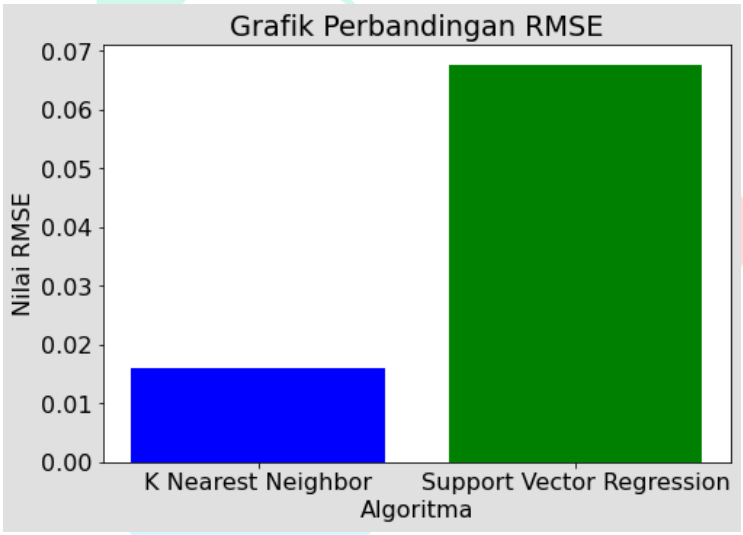
Gambar 5. 28.Grafik perbandingan akurasi RMSE KNN Low(d+1) dan SVR Low(d+1) (time-series, k =6, rasio data latih 80:20, data uji = 309)



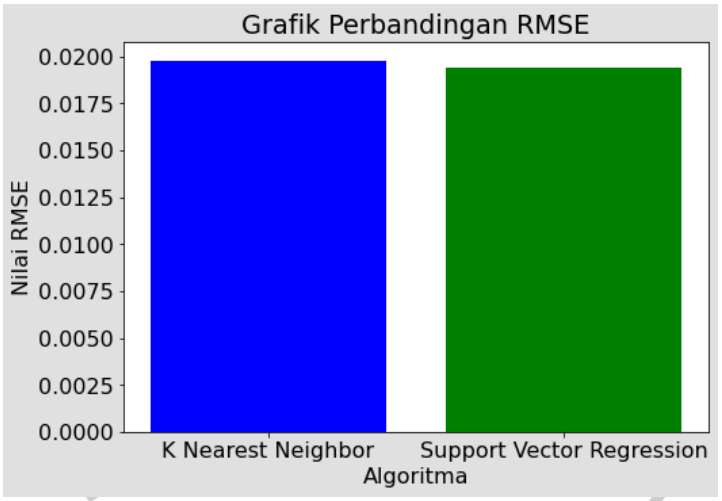
Gambar 5. 29.Grafik perbandingan akurasi R2 KNN High(d+1) dan SVR High(d+1) (time-series, k =5, rasio data latih 80:20, data uji = 309)



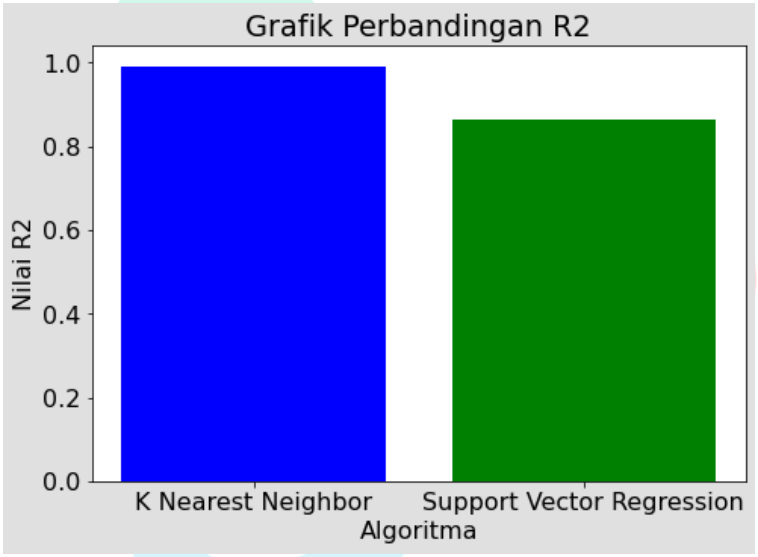
Gambar 5. 30.Grafik perbandingan akurasi R2 KNN Low(d+1) dan SVR Low(d+1) (time-series, k =6, rasio data latih 80:20, data uji = 309)



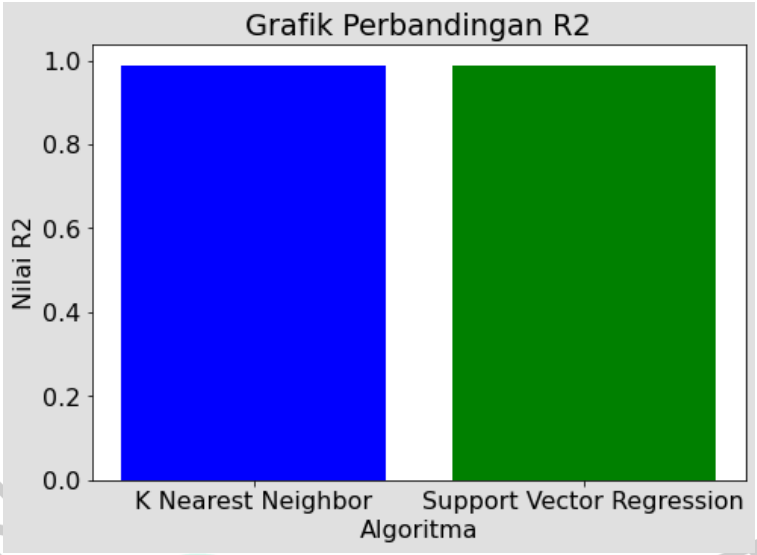
Gambar 5. 31.Grafik perbandingan akurasi RMSE KNN High(d+1) dan SVR High(d+1) (time-series, k =4, rasio data latih 70:30, data uji = 463)



Gambar 5. 32.Grafik perbandingan akurasi RMSE KNN Low(d+1) dan SVR Low(d+1) (time-series, k =3, rasio data latih 70:30, data uji = 463)



Gambar 5. 33.Grafik perbandingan akurasi R2 KNN High(d+1) dan SVR High(d+1) (time-series, k =4, rasio data latih 70:30, data uji = 463)



Gambar 5. 34.Grafik perbandingan akurasi R2 KNN Low(d+1) dan SVR Low(d+1) (time-series, k =3, rasio data latih 70:30, data uji = 463)

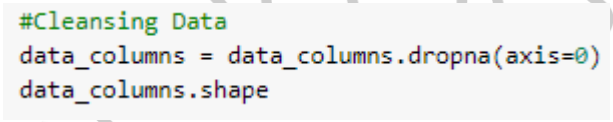
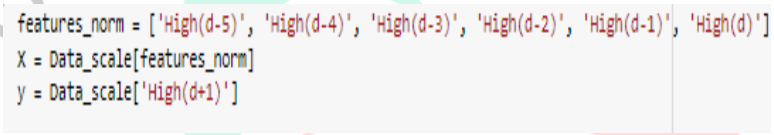
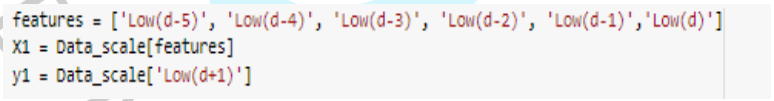
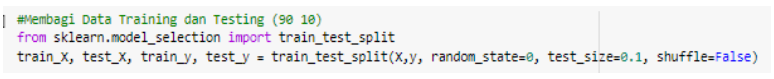
5.2 Hasil Pembahasan Pengujian Kedua Algoritma

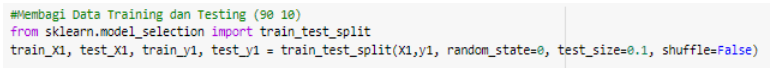
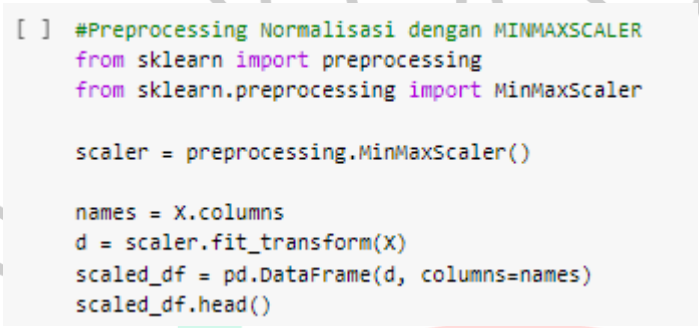
Dilakukannya pengujian terhadap kedua algoritma. Pengujian ini bertujuan untuk melihat apakah kode program yang dimasukkan sudah menghasilkan apa yang peneliti harapkan. Berikut hasil pengujian program terhadap kedua algoritma.

Tabel 5. 5 Pengujian program K-Nearest Neighbors

No.	Kode Program	Hasil yang Diharapkan	Hasil Pengujian

1.	`#Import libraries`
	`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`
		Melakukan import library yang akan digunakan .	Valid

2.		Dataset yang sudah ada diimport ke dalam program menjadi sebuah variabel.	Valid
3.		Data yang tidak lengkap atau bernilai nol akan dibersihkan.	Valid
4.		Menentukan fitur pertama yang akan digunakan. Serta menetapkan target High(d+1) yang akan dilakukan untuk prediksi.	Valid
5.		Menentukan fitur kedua yang akan digunakan. Serta target kedua yang akan diprediksi yaitu Low(d+1).	Valid
6.		Pada features high d+1 membagi data training dan	Valid

		data test sebesar 90:10.	
7.		Pada features low d+1 membagi data train dan data test sebesar 90:10.	Valid
8.		Normalisasi data menggunakan *min max scaler*	Valid
9.		Mencari nilai K optimal menggunakan metode elbow	Valid
10.		Melakukan pemodelan KNN dan mencari akurasi RMSE dan R^2	Valid

```
[26] #define model KNN
     model_knn= KNN_Reg(n_neighbors = 2)

     # training the model KNN
     model_knn.fit(train_X, train_y)

     KNeighborsRegressor(n_neighbors=2)
```

```
[27] # test for prediction
     test_predict_knn = model_knn.predict(test_X)
```

```
[28] #RMSE
     rmse_knn = sqrt(mse(test_y,test_predict_knn))

     #R2
     r2_knn =r2_score(test_y,test_predict_knn)
```

```
[29] print('RMSE K-NN: ', rmse_knn)
     print('Skor R2 K-NN: ', r2_knn)
```

11.

```
#Prediksi 1 Hari Selanjutnya
harga_ethereum =np.array([[2510, 2548, 2627, 2631, 2697,2802]])

#Pemodelan Prediksi
prediction_svr = SVR_model.predict(harga_ethereum)
prediction_knn = model_knn.predict(harga_ethereum)

# 01 Februari 2022 :2,802.32

#Hasil Prediksi
print ('Prediksi High Pada 02 Februari 2022')
print ('SVR : ', prediction_svr)
print ('KNN : ', prediction_knn)
```

Melakukan prediksi high d+1 pada harga Ethereum di tanggal 02 Februari 2022 — Valid

12.

```
#Prediksi 1 Hari Selanjutnya
harga_ethereum1 =np.array([[2328, 2367, 2526, 2550, 2489, 2682]])

#Pemodelan Prediksi
prediction_svr1 = SVR_model1.predict(harga_ethereum1)
prediction_knn1 = model_knn1.predict(harga_ethereum1)

# 01 Februari 2022 :2,802.32

#Hasil Prediksi
print ('Prediksi High Pada 02 Februari 2022')
print ('SVR : ', prediction_svr1)
print ('KNN : ', prediction_knn1)
```

Melakukan prediksi low d+1 pada harga Ethereum di tanggal 02 Februari 2022 — Valid

Sumber: Data diolah peneliti

Tabel 5. 6 Pengujian program Support Vector Machine

No.	Kode Program	Hasil yang Diharapkan	Hasil Pengujian

1.	`#Import libraries`
	`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`
		Melakukan import library yang akan digunakan .	Valid

2.	`#Load dataset`
	`path = "/content/drive/MyDrive/Dataset/HISTORY_ETH-USD.csv"`
	`data_columns = pd.read_csv(path,index_col=0,parse_dates=True, dtype={'path':int})`
	`data_columns.describe()`
		Dataset yang sudah ada diimport ke dalam program menjadi sebuah variabel.	Valid

3.	`#Cleansing Data`
	`data_columns = data_columns.dropna(axis=0)`
	`data_columns.shape`
		Data yang tidak lengkap atau bernilai nol akan dibersihkan.	Valid

4.	`features_norm = ['High(d-5)', 'High(d-4)', 'High(d-3)', 'High(d-2)', 'High(d-1)', 'High(d)']`
	`X = Data_scale[features_norm]`
	`y = Data_scale['High(d+1)']`
		Menentukan fitur pertama yang akan digunakan. Serta menetapkan target High(d+1) yang akan dilakukan untuk prediksi.	Valid

5.	`features = ['Low(d-5)', 'Low(d-4)', 'Low(d-3)', 'Low(d-2)', 'Low(d-1)','Low(d)']`
	`X1 = Data_scale[features]`
	`y1 = Data_scale['Low(d+1)']`
		Menentukan fitur kedua yang akan digunakan. Serta target kedua yang	Valid

			akan diprediksi yaitu Low(d+1).

6.	`#Membagi Data Training dan Testing (90 10)`
	`from sklearn.model_selection import train_test_split`
	`train_X, test_X, train_y, test_y = train_test_split(X,y, random_state=0, test_size=0.1, shuffle=False)`
			Pada features high membagi data training dan data test sebesar 90:10.

7.	`#Membagi Data Training dan Testing (90 10)`
	`from sklearn.model_selection import train_test_split`
	`train_X1, test_X1, train_y1, test_y1 = train_test_split(X1,y1, random_state=0, test_size=0.1, shuffle=False)`
		Valid	Pada features low d+1 membagi data train dan data test sebesar 90:10.

8.	`[ ] #Preprocessing Normalisasi dengan MINMAXSCALER`
	`from sklearn import preprocessing`
	`from sklearn.preprocessing import MinMaxScaler`
	
	`scaler = preprocessing.MinMaxScaler()`
	
	`names = X.columns`
	`d = scaler.fit_transform(X)`
	`scaled_df = pd.DataFrame(d, columns=names)`
	`scaled_df.head()`
		Valid	Normalisasi data menggunakan *minmax scaler*

9.	`[30] # Model SVR`
	`from sklearn.svm import SVR`
	`SVR_model = SVR(kernel = 'linear', C=10)`
	`#SVR_model = SVR(kernel = 'rbf', C=5000)`
	
	`# Fit Model`
	`SVR_model.fit(train_X, train_y)`
	
	`# test for prediction`
	`test_predict_SVR = SVR_model.predict(test_X)`
	
	`#RMSE`
	`rmse_SVR = sqrt(mse(test_y,test_predict_SVR))`
	
	`#R2`
	`r2_SVR =r2_score(test_y,test_predict_SVR)`
	
	`print('RMSE SVR: ', rmse_SVR)`
	`print('Skor R2 SVR: ', r2_SVR)`
		Valid	Melakukan pemodelan SVR dan mencari akurasi RMSE dan R^2

10.	`#Prediksi 1 Hari Selanjutnya`
	`harga_ethereum =np.array([[2510, 2548, 2627, 2631, 2697,2802]])`
	
	`#Pemodelan Prediksi`
	`prediction_svr = SVR_model.predict(harga_ethereum)`
	`prediction_knn = model_knn.predict(harga_ethereum)`
	
	`# 01 Februari 2022 :2,802.32`
	
	`#Hasil Prediksi`
	`print ('Prediksi High Pada 02 Februari 2022')`
	`print ('SVR : ', prediction_svr)`
	`print ('KNN : ', prediction_knn)`
		Melakukan prediksi high d+1 pada harga Ethereum di tanggal 02 Februari 2022	Valid

11.	`#Prediksi 1 Hari Selanjutnya`
	`harga_ethereum1 =np.array([[2328, 2367, 2526, 2550, 2489, 2682]])`
	
	`#Pemodelan Prediksi`
	`prediction_svr1 = SVR_model1.predict(harga_ethereum1)`
	`prediction_knn1 = model_knn1.predict(harga_ethereum1)`
	
	`# 01 Februari 2022 :2,802.32`
	
	`#Hasil Prediksi`
	`print ('Prediksi High Pada 02 Februari 2022')`
	`print ('SVR : ', prediction_svr1)`
	`print ('KNN : ', prediction_knn1)`
		Melakukan prediksi low d+1 pada harga Ethereum di tanggal 02 Februari 2022	Valid

Sumber: Data diolah penelit