

## **BAB II**

### **TINJAUAN PUSTAKA**

Dalam penelitian ini, terdapat tinjauan pustaka dari pencapaian terdahulu dan tinjauan teoritis. Berikut penjabaran pada tinjauan pustaka yakni:

#### **2.1 Pencapaian Terdahulu**

Peneliti menyajikan beberapa penelitian relevan yang sudah ada sebelumnya untuk mendukung peneliti dalam melakukan penelitian ini. Adapun kegunaannya untuk mengetahui hasil yang telah dilakukan oleh peneliti terdahulu. Beberapa peneliti telah melakukan penelitian pada pengujian akurasi SVM (*Support Vector Machine*) dan KNN (*K-Nearest Neighbor*) dalam pemodelan harga altcoin *Crypto* yang mengacu pada penelitian terdahulu. Referensi penelitian terdahulu dengan algoritma yang sama dijadikan sebagai kerangka acuan. Situasi yang dianalisis beragam, demikian pula hasil akurasi pengukuran.

Pertama, jurnal "*Comparative Study on KNN and SVM Based Weather Classification Models for Day Ahead Short Term Solar PV Power Forecasting*". Penelitian ini membandingkan algoritma SVM dan KNN pada mengklasifikasikan peramalan cuaca untuk jangka pendek dan sehari kedepannya peramalan tenaga surya PV (Bianchi, 2020, p 162). Hasil perbandingan algoritma SVM dan K-Nearest Neighbor untuk peramalan cuaca tenaga surya PV yaitu, hasil dari simulasi algoritma SVM memiliki tingkat akurasi lebih baik skala sampel kecil sedangkan KNN lebih sensitif terhadap panjang *dataset* pelatihan dan dapat mencapai akurasi di atas SVM dengan sampel yang cukup (Fei Wang et al., 2017).

Kedua, jurnal "*Application of SVM-KNN using SVR as feature selection on stock analysis for Indonesia stock exchange*". Penelitian dilakukan untuk menganalisa perbandingan SVM dan KNN dalam menganalisa saham pada bursa efek Indonesia. Hasil perbandingan antara algoritma SVM dan KNN dalam menganalisa saham pada bursa efek Indonesia. yaitu, Algoritma SVM dan KNN dilakukan dengan metode evaluasi RMSE yang dimana hasil akurasi prediksi harga penutupan adalah 93,33% akurat (D.A.Puspitasari & Z.Rustam, 2018).

Kemudian ketiga, jurnal “*Predicting uncertainty of machine learning models for modelling nitrate pollution of groundwater using quantile regression and UNEEC methods*” (Omid Rahmati et al., 2019). Penelitian ini dilakukan untuk memprediksi ketidakpastian model pembelajaran mesin untuk memodelkan nitrat pencemaran air tanah menggunakan regresi kuantil dan metode UNEEC (Firmansyah et al., 2019). Pada penelitian ini menggunakan metode evaluasi RMSE. Penelitian ini membandingkan tiga algoritma yaitu SVM, *Random Forest*, dan *KNN*. Hasil dari penelitian ini algoritma *KNN* memiliki tingkat akurasi yang lebih tinggi. Berdasarkan Statistik PICP baik dalam metode QR (0,94) dan UNEEC (di semua cluster, 0,85-0,91), tetapi juga memiliki metode prediktif statistik kinerja (RMSE = 10,63,  $R^2 = 0,71$ ) yang relatif sama dengan RF (RMSE = 10,41,  $R^2 = 0,72$ ) dan lebih tinggi dari SVM (RMSE = 13,28,  $R^2 = 0,58$ ).

## 2.2 Tinjauan Teoritis

Tinjauan teoritis merupakan landasan pendekatan teori yang digunakan peneliti untuk menjelaskan persoalan penelitian. Dalam penelitian ini, peneliti menggali informasi dari jurnal ilmiah dan artikel ilmiah dalam rangka mendapatkan informasi yang ada sebelumnya tentang teori yang berkaitan dengan judul, serta digunakan untuk memperoleh landasan teori ilmiah.

### 2.2.1 *Machine Learning*

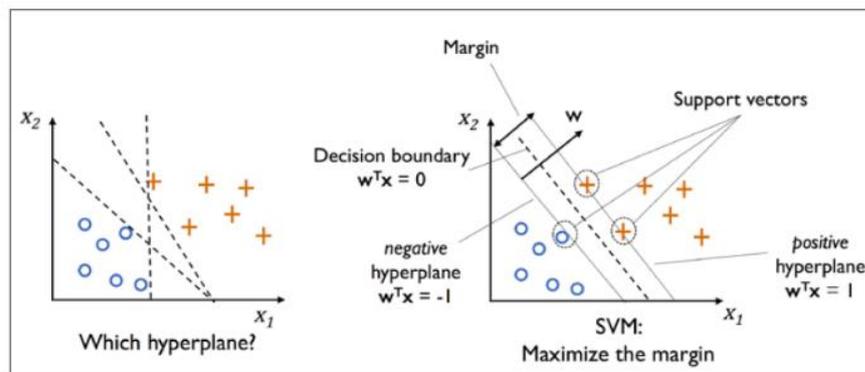
*Machine learning* pertama kali diteliti oleh ilmuwan matematika Thomas Bayes, Andrey Markov, dan Andrien Marie yang dikemukakan pada tahun 1920. *machine learning* merupakan mesin yang dikembangkan dengan tujuan bisa belajar tanpa adanya perintah dari penggunaannya dan mempunyai kemampuan pada hal membaca sebuah data yang sudah ada dan bisa memproses data tersebut. *machine learning* dapat melakukan tugas yang sangat beragam dengan tugas yang sudah dipelajari oleh *machine learning* (Dicoding Intern R & I, 2020)

### 2.2.2 *Support Vector Machine (SVM)*

SVM yaitu metode yang ada pada *machine learning*. SVM juga merupakan *machine learning* yang terbaik untuk *klasifikasi dan regresi*. SVM memiliki konsep yang berawal dari sebuah masalah regresi sehingga butuh pelatihan positif dan negatif dengan tujuan menciptakan *hyperplane* (pemisah) yang

terbaik yang digunakan dalam memisahkan ke dalam dua kelas dan menguatkan *margin* yang di antara dua kelas tersebut. Tidak semua masalah bisa klasifikasikan dengan metode SVM ini, maka dari itu adanya pengembangan pada fungsi kernel yang bertujuan untuk bisa mengklasifikasikan data yang berbentuk non linier (Pratama et al., 2018).

SVM difungsikan bertujuan untuk mendapatkan *hyperplane* terbaik dan memaksimalkan antar kelas. *Hyperplane* merupakan sebuah fungsi yang bisa membagi dua kelas. Klasifikasi fungsi untuk membagi kelas ada tiga yaitu. *Line* Whereas merupakan fungsi yang berguna untuk membagi antar kelas dalam 2-D. *Plane* Similarly merupakan fungsi untuk membagi antar kelas dalam 3-D. Terakhir fungsi tertinggi dalam membagi antar kelas yaitu *hyperplane* (Samsudiney, 2019).



Gambar 2. 1. *Hyperplane* yang memisahkan dua kelas

Gambar 2.1 merupakan ilustrasi dari *hyperplane* dalam SVM. Garis tengah yang berada tengah-tengah antara bulat dan positif merupakan jarak antara dua kelas atau bisa disebut *hyperplane*. Pada gambar tersebut ada sebuah objek yang bernama *support vectors*. *Support vector* itu merupakan objek yang paling sulit untuk diklasifikasikan disebabkan *support vector* adalah objek yang paling terdekat dengan garis *hyperplane*. Objek *support vectors* tersebut terbilang sulit diklasifikasikan disebabkan posisinya yang overlap pada kelas lainnya. Maka dari itu SVM merupakan *machine learning* yang memiliki tingkat keakuratan yang tinggi (Samsudiney, 2019).

### 2.2.3 *K-Nearest Neighbor*

Algoritma *K-Nearest Neighbor* (KNN) merupakan salah satu metode klasifikasi dalam data mining, dimana KNN mengklasifikasikan satu set data berdasarkan data pembelajaran yang telah diklasifikasikan atau diberi label. KNN termasuk dalam kelompok pembelajaran *supervised*, yaitu hasil instans kueri yang baru diklasifikasikan berdasarkan mayoritas kedekatan dengan kategori yang ada di KNN (A. Wanto, et al., 2020).

*K-Nearest Neighbor* memiliki konsep umum untuk regresi sama dengan seperti klasifikasi. Menentukan nilai k tetangga terdekat pada dataset. Selanjutnya melakukan prediksi dengan label dari k tetangga terdekat. Berikut rumus Euclidean Distance pada persamaan (1) sebagai berikut:

$$euc = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad (1)$$

Keterangan :

a = data latih

b = data uji

k = variable data

d = dimensi data

Nilai *Euclidean Distance* merupakan perhitungan untuk mengukur jarak dua titik. Euclidean Distance bertujuan untuk membantu klasifikasi data pada data mining.

### 2.2.4 *Trading*

Trading merupakan sebuah proses transaksi finansial jangka pendek atau istilahnya yaitu perdagangan dalam bentuk mata uang. *Trading* menjadi salah satu instrumen yang paling penting karena bisa mendapatkan profit atau keuntungan yang sangat besar. Dalam dunia trading ini seorang *trader* harus memiliki keahlian

dalam membaca pergerakan pasar dengan sangat baik. Seorang *trader* harus selalu mengamati harga disetiap waktu agar dapat menghasilkan profit yang besar. Prinsip dari *trading* yaitu *Buy and Sell* yang dimana seorang *trader* menggelontorkan uangnya untuk membeli mata uang atau *coin crypto* pada saat harganya berpotensi naik dalam waktu dekat. Jadi *trader* terfokus kepada perubahan harga (Pintek, 2021).

### **2.2.5 Cryptocurrency**

*Cryptocurrency* merupakan mata uang digital yang dibangun oleh sistem Blockchain. Mata uang digital ini tidak memerlukan orang ketiga atau perantara. Maka dari itu *Cryptocurrency* ini merupakan transaksi yang transparan karena dan tidak memiliki perantara. Karena memakai sistem *blockchain* maka setiap transaksi data akan saling terhubung dimana data tersebut mencakup lingkungan pengguna *Cryptocurrency* tersebut (Bhiantara, 2018). *Cryptocurrency* sudah memikat banyak sekali investor belakangan ini ini disebabkan beberapa faktor yaitu:

- 1) *Cryptocurrency* memiliki buku besar yang bisa mencatat semua transaksi.
- 2) *Cryptocurrency* merupakan mata uang terdesentralisasi jadi pemerintah dan bank tidak bisa mengendalikannya.
- 3) Kemudahan dalam melakukan pemakaian *Cryptocurrency*.
- 4) Identitas pengguna sangat terjaga dengan adanya buku besar.

Uraian tersebut merupakan faktor-faktor kenapa *Cryptocurrency* sangat diminati oleh para investor.

### **2.2.6 Blockchain**

*Blockchain* merupakan teknologi terbaru yang tugasnya pencatatan transaksi dan terhubung dengan satu sama lain. Cara kerja blockchain saat transaksi yang baru pemakaian atau pengeditan pada transaksi yang sebelumnya sudah ada dimasukkan ke sistem *blockchain* pada dasarnya adalah sebuah node di dalam sistem blockchain akan melakukan kerja pada algoritma yang gunanya untuk evaluasi dan verifikasi pada blok yang sudah ada *blockchain* individual yang diajukan. Blok baru akan diterima dan dikirim ke database sekaligus ditambah ke database jika transaksi tersebut sudah disahkan. Apabila dari sekian banyak orang

tidak setuju dengan perubahan atau penambahan maka entri pada database dibatalkan. Mekanisme konsensus terdistribusi inilah yang memungkinkan blockchain berfungsi sebagai database terdistribusi tanpa memerlukan banyak dari sebuah otoritas. Maka otoritas terpusat menentukan transaksi mana yang asli dan mana yang tidak (Bhiantara, 2018).

### **2.2.7 Ethereum**

Ethereum merupakan sebuah altcoin (*alternative coin*) yang jaringan rantai blok publik yang memfokuskan dalam melakukan pemrograman yang terdesentralisasi yang diciptakan oleh Vitalik Buterin pada tahun 2013. Ethereum merupakan sebuah *platform* yang terdesentralisasi berguna untuk smart contract. Smart contract merupakan aplikasi yang berjalan sendiri sesuai input pemrogramannya. smart contract ini memiliki kelebihan tidak adanya *downtime*, penipuan atau gangguan dari eksternal maupun internal, Dapat berjalan selama 24 jam.

### **2.2.8 Pemrograman Python**

*Python* dikemukakan oleh Guido Van Rossum pada tahun 1991. Pemrograman *Python* yaitu bahasa pemrograman yang dijumpai dan digunakan. Bahasa pemrograman sangat populer dan sering digunakan kalangan programmer. *Python* adalah bahasa pemrograman yang berorientasi dinamis. Dari keunggulannya dan kapabilitasnya, Python diakui sebagai bahasa pemrograman yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif (Manzilati, A, 2021) . Bahasa pemrograman *Python* populer digunakan dalam pengujian pemodelan menggunakan *machine learning*. Dalam penelitian ini, pengujian dilakukan dengan menerapkan bahasa *Python* untuk memprediksi harga Ethereum dan menerapkan *machine learning* pada algoritma SVM dan *Decision Tree*.

### **2.2.9 Root Mean Squared Error**

Tujuan dari evaluasi hasil adalah untuk menentukan kebenaran dari algoritma pembelajaran mesin yang digunakan. Root Mean Squared Error (RMSE) adalah perhitungan yang dapat digunakan untuk menghitung akurasi kinerja algoritma pembelajaran mesin (Fatin, N, 2017). Algoritma yang memiliki akurasi

yang paling baik yaitu <sup>2</sup> algoritma yang memiliki RMSE yang paling terkecil karena tingkat *error*nya yang paling rendah. Berikut rumus dari RMSE pada persamaan (2) sebagai berikut.

$$RMSE = \sqrt{\frac{1}{n} \left( \sum_{j=1}^n (y_j - \hat{y}_j)^2 \right)} \quad (2)$$

Y = original data test

$\hat{y}$  = prediksi data test

n = jumlah dari data test.

### 2.2.10 R-Squared

*R-square* merupakan metode evaluasi koefisien determinasi yang dimana cara kerja dari R-Squared ini menentukan proporsi varian dalam bentuk variabel dependen yang dapat dijelaskan dengan variabel independen. *R-square* adalah angka berkisar 0 sampai 1 yang mengindikasikan besarnya kombinasi variabel independen secara bersama – sama mempengaruhi nilai variabel dependen. Semakin mendekati angka satu, model yang dikeluarkan oleh regresi tersebut akan semakin baik (Rheem et al, 2019, p. 114).

$$R^2 = \frac{SSres}{SStotal} \quad (3)$$

SSres = Sum Squared Regression

SStotal = Sum Squared Total

Berikut merupakan rumus dari  $R^2$  pada persamaan (3) yaitu Sum Squared Regression(SSres) dibagi dengan Sum Squared Total (SStotal).SSres adalah kuadrat dari selisih nilai Y dengan nilai rata-rata. Sebagai berikut rumus untuk mendapatkan SSres.

$$SSres = \sum (Y_{Pred} - Y_{rata-rata})^2 \quad (4)$$

SSres = Sum Squared Regression

Ypred = nilai Y prediksi

Yrata-rata = nilai Y dengan nilai rata-rata

Pada persamaan (4) yaitu rumus untuk mendapatkan SSres. SStotal adalah kuadrat dari selisih nilai Y aktual dengan nilai rata-rata. Berikut rumus pada persamaan (5) untuk mendapatkan SStotal.

$$SStotal = \sum (Y_{aktual} - Y_{rata-rata})^2 \quad (5)$$

SStotal = Sum Squared Total

Yaktual = nilai Y data aktual

Yrata-rata = nilai Y dengan nilai rata-rata

### 2.2.11 Analisis *Time Series*

Analisis *time series* adalah menganalisis atau mengamati data yang berorientasi pada waktu atau kronologis pada variabel yang akan diamati. Analisis *time series* ini sangat berguna untuk menganalisa data yang dipengaruhi dengan waktu. Contohnya di bidang keuangan, ekonomi, trading dan masih banyak lagi. Maka dari itu analisis *time series* ini sangat cocok untuk mengolah data yang dipengaruhi waktu (Prasetya et al, 2020, p. 714).

### 2.2.12 *Scikit-learn*

*Scikit-learn* atau *sklearn* merupakan *library* machine learning open source yang ditujukan untuk pemrograman *python*. *scikit-learn* memiliki fungsi untuk mengekstrak fitur, memproses data, dan evaluasi model. *Scikit-learn* sangat diminati oleh para peneliti disebabkan *scikit-learn* memiliki fungsi yang mudah digunakan, dan dokumentasi yang jelas (Hackeling, G. 2017).

### 2.2.13 *NumPy*

*NumPy* adalah sebuah *library* pada pemrograman *python* yang memiliki fungsi *powerfull* dan efisien untuk melakukan *scientific*. *numpy* memiliki fitur spesial disebabkan dalam pemrosesan numerik melewati multi dimensi *ndarrays*. *library numpy* ini memiliki fungsi yang bagus dalam melakukan komputasi

scientific, terutama adalah multidimensi array, dan opsi matematika (sumber: numpy.org, 2022).

#### 2.2.14 *matplotlib*

*Matplotlib* adalah sebuah *library python* yang digunakan untuk membuat 2D *plotting*. *matplotlib* juga dapat mendukung pekerjaan *interaktif* ataupun *non interaktif plotting*. *Library* ini juga dapat menyimpan gambar dengan beberapa output format diantaranya adalah *PNG*, dan *PS*. *Matplotlib* dapat membuat banyak variasi dari tipe plot, seperti garis, bar, diagram pie, histogram, dan lainnya (sumber: matplotlib.org, 2021).

#### 2.2.15 *Pandas*

*Pandas* adalah sebuah *library python* yang bekerja untuk menyediakan data struktur yang cepat, mudah digunakan dan ekspresif. *Dataframe* merupakan objek utama pada *pandas*, sebuah data struktur berorientasi kolom dengan judul baris dan kolom, dan *series* (objek array berlabel satu dimensi). *Pandas* juga mempunyai fungsi yang digunakan untuk *reshape*, *slice* dan *dice*, melakukan agregasi, dan memilih subset data. (sumber: pandas.pydata.org, 2021).

#### 2.2.16 *Confusion Matrix*

*Confusion matrix* adalah ukuran yang sangat populer yang digunakan saat memecahkan masalah klasifikasi. Hal ini dapat diterapkan pada klasifikasi biner serta untuk masalah klasifikasi multi-kelas. Contoh *confusion matrix* untuk klasifikasi biner ditunjukkan pada Tabel 3.1.

Tabel 3.1. *Confusion Matrix for Binary Classification*

<i>Actual</i>	<i>Predict</i>	
	<i>Negative</i>	<i>Positive</i>
<i>Negative</i>	TN	FP

*Positive*

FN

TP

---

Sumber: Kulkarni, Ajay, 2021

*Confusion matrix* mewakili hitungan dari nilai yang diprediksi dan aktual. Output "TN" adalah singkatan dari *True Negative* yang menunjukkan jumlah contoh negatif yang diklasifikasikan secara akurat. Demikian pula, "TP" berdiri untuk *True Positive* yang menunjukkan jumlah contoh positif yang diklasifikasikan secara akurat. Istilah "FP" menunjukkan nilai Positif Palsu, yaitu jumlah contoh negatif aktual yang diklasifikasikan sebagai positif; dan "FN" berarti nilai Negatif Palsu yang merupakan jumlah contoh positif aktual yang diklasifikasikan sebagai negatif. Salah satu *matrix* yang paling umum digunakan saat melakukan klasifikasi adalah akurasi. Keakuratan model (melalui *confusion matrix*) dihitung menggunakan rumus pada persamaan (6) sebagai berikut.

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \quad (6)$$

TN = *True Negative*

TP = *True Positif*

FP = *False Positif*

FN = *False Negative*

Akurasi dapat menyesatkan jika digunakan dengan dataset yang tidak seimbang, dan oleh karena itu ada metrik lain berdasarkan matriks kebingungan yang dapat berguna untuk mengevaluasi kinerja. Dalam *Python*, matriks kebingungan dapat diperoleh menggunakan fungsi "*confusion matrix()*" yang merupakan bagian dari perpustakaan "*sklearn*". Fungsi ini dapat diimpor ke *Python* menggunakan "dari *sklearn.metrics import confusion\_matrix*". Untuk mendapatkan matriks kebingungan, pengguna perlu memberikan nilai aktual dan nilai yang diprediksi ke fungsi.