

ANALISIS INVESTASI DALAM MEMPREDIKSI PERGERAKAN HARGA BITCOIN DENGAN MENGGUNAKAN RECURRENT NEURAL NETWORK PADA PLATFORM INDODAX

Julianto

Institute Agama Islam Negeri (IAIN) Pontianak
E-mail : julianto@iainptk.ac.id

Abstract

Kemajuan teknologi yang semakin pesat membuat banyak bidang mengalami perubahan termasuk didalamnya bidang investasi asset digital terutama crypto. Ada banyak cara yang dilakukan oleh para trader maupun investor dalam melakukan perdagangan Bitcoin yang merupakan salah satu asset digital di dunia crypto. Indodax merupakan salah satu platform buatan local Indonesia yang melayani transaksi perdagangan asset digital. Analisis teknikal dan fundamental dilakukan untuk memprediksi pergerakan harga bitcoin, namun volatilitas yang tinggi menyebabkan pergerakan bitcoin sulit untuk diprediksi. Penggunaan Recurrent Neural Network yang merupakan sub bidang ilmu dari Machine Learning merupakan salah satu cara untuk dapat melakukan prediksi terhadap bitcoin.

Kata Kunci : *RNN, LSTM, Bitcoin, Indodax, Training, Testing*

Abstract

Rapid technological advances have made many fields change, including the field of digital asset investment, especially crypto. There are many ways for traders and investors to trade Bitcoin which is one of the digital assets in the crypto world. Indodax is a platform made locally in Indonesia that serves digital asset trading transactions. Technical and fundamental analysis is carried out to predict bitcoin price movements, but high volatility makes bitcoin movements difficult to predict. The use of the Recurrent Neural Network which is part of Machine Learning is one way to be able to make predictions on bitcoin.

Key Word : *RNN, LSTM, Bitcoin, Indodax, Training, Testing*

A. PENDAHULUAN

Kemajuan teknologi akan mendorong perubahan, perubahan tersebut memberikan dampak yang signifikan pada berbagai bidang, salah satunya adalah bidang investasi. Jenis dan wujud dari investasi sendiri sangat beragam mulai dari tabungan bank, deposito, Forex, Emas, Tanah, Bangunan, Asuransi, Reksadana dan *Cryptocurrency* atau mata uang digital salah satunya adalah *Bitcoin*.

Untuk membantu investasi di bidang crypto secara optimal, diperlukan sebuah *tools* yang dapat membantu investor maupun *trader* dalam memprediksi pergerakan harga *Bitcoin*. *Tools* tersebut menggunakan arsitektur *Deep Learning* yakni *Recurrent Neural Network* yang merupakan *algoritma Deep Learning* yang merupakan tipe *Neural Network* yang sangat baik dalam menemukan pola temporal dalam melakukan prediksi.

Prediksi merupakan suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013 : 8). Prediksi memungkinkan hasil yang dicapai bisa mendekati nilai kebenaran yang dapat dijadikan acuan dalam pengambilan keputusan.

Menurut Larose (2015:4), Data mining adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar Larose (2015:5) juga mengungkapkan bahwa berdasarkan riset yang dilakukan oleh Marketsand Market, pasar data global diperkirakan akan tumbuh sebesar 26% untuk tahun 2013 hingga 2018, sehingga hampir seluruh perusahaan di dunia berlomba-lomba untuk mempelajari data mining dengan tujuan dapat menemukan pola dan tren yang berguna sebagai kesempatan untuk meningkatkan laba.

Larose (2005:6-8) mengungkapkan bahwa proyek *Data Mining* mempunyai sebuah siklus hidup yang terdiri dari enam tahapan. Keenam tahapan tersebut dikenal dengan istilah *Cross-Industry Standard Process for Data Mining*, jika digambarkan akan seperti gambar 1.



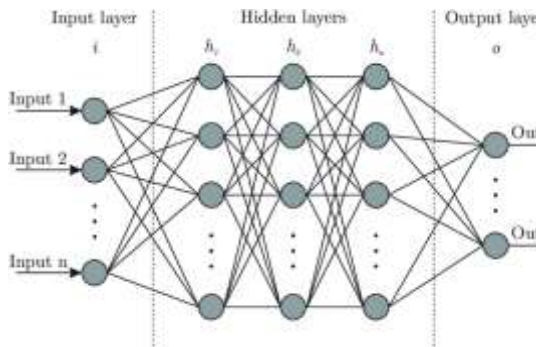
Gambar 1. CRISP-DM (Larose, 2015:4)

Artificial Intelligence (AI) merupakan sebuah konsep umum tentang bagaimana membuat mesin-mesin yang mampu berpikir secara cerdas (*smart machine*) seperti layaknya manusia, atau mesin yang mampu belajar dari berbagai input yang diberikan padanya. Sehingga menghasilkan outcome yang akurat (Primatha, 2018).

Machine Learning adalah pemrograman komputer untuk mencapai kriteria atau performa tertentu dengan menggunakan dengan menggunakan sekumpulan data *training* atau pengalaman di masa lalu (Primatha, 2018). *Machine Learning* memerlukan sebuah model yang didefinisikan berdasarkan parameter-parameter tertentu. Proses *learning* adalah eksekusi program komputer untuk mengoptimasi parameter-parameter dari model tersebut, dengan memanfaatkan *Data Mining*.

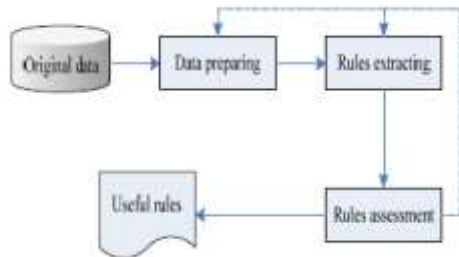
Neural Network adalah teknik komputasi yang memproses informasi, berkaitan input variabel dengan *output* yang diinginkan. JST mendasarkan perhitungan mereka pada interkoneksi unit sederhana yang disebut *neuron* buatan. Model-model ini terinspirasi oleh fungsinya *neuron* sel dan bagaimana ia menghasilkan dan menyebarkan *impuls* listrik (Loyo, 2017).

Neural Network secara esensial terdiri dari tiga bagian: arsitektur atau model, *algoritma learning*, dan fungsi aktivasi. Pada gambar 2.2 diperlihatkan arsitektur *Artificial Neural Network*.



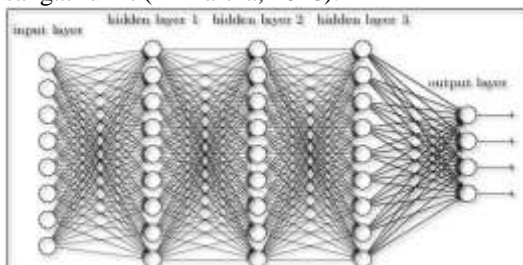
Gambar 2. Arsitektur Artificial Neural Network (Bre, 2017)

Data Mining berdasarkan Neural Network disusun dengan menyiapkan data, rule extracting, rules assessment sehingga menghasilkan useful rules yang berguna dalam menghasilkan informasi yang dibutuhkan. Dimana alur proses Data Mining yang berdasarkan Neural Network dapat dilihat pada gambar 2.3..



Gambar 3 Data Mining Process based on Neural Network (Ni, 2018)

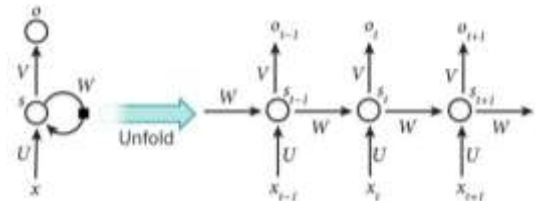
Deep Learning merupakan metode learning yang memanfaatkan Artificial Neural Network yang berlapis lapis (multi layer). Artificial Neural Network dibuat mirip dengan otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit (Primartha, 2018).



Gambar 4 Diagram Deep Learning (Primartha, 2018)

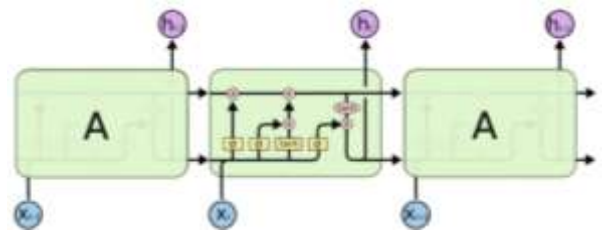
Recurrent Neural Network (RNN) merupakan bagian yang tak terpisahkan dari Deep

Learning. Deep Learning dikenal sebagai pembelajaran terstruktur dan pembelajaran hierarkis adalah bagian dari keluarga yang lebih luas dari Machine Learning berdasarkan Jaringan Saraf Tiruan. Pembelajaran dapat berupa Supervised, Semi-Supervised, dan Unsupervised (Bengio, Y.; Courville, A.; Vincent, P., 2013). Adapun proses RNN ditunjukkan pada gambar 2.4.



Gambar 5 Proses RNN saat menghitung proses didepannya (Britz, 2015)

LSTM (Long Short Term Memory) adalah Arsitektur Recurrent Neural Network (RNN) yang digunakan dalam bidang Deep Learning. LSTM memiliki koneksi feedback yang tidak hanya data memproses titik data tunggal (seperti gambar), tetapi seluruh urutan data (seperti speech atau video). LSTM pertama kali diajukan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, LSTM menjadi salah satu model yang banyak digunakan pada Deep Learning untuk NLP (Ashrovy, 2017).

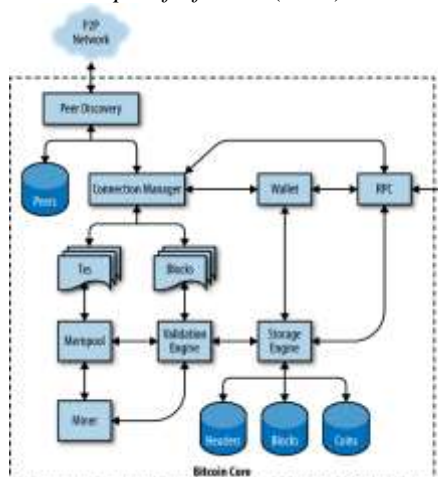


Gambar 6 Arsitektur Algoritma LSTM, (Olah, Christopher, 2017)

Cryptocurrency adalah asset digital yang dirancang yang berfungsi sebagai media pertukaran menggunakan kriptografi yang kuat untuk mengamankan transaksi keuangan, mengontrol penciptaan unit tambahan, dan memverifikasi transfer asset. (Schueffel, 2017). Cryptocurrency adalah nama yang diberikan untuk sebuah sistem yang merupakan kriptografi untuk melakukan proses pengiriman data secara aman untuk melakukan proses pertukaran token digital secara tersebar (Dourado & Brito, 2014).

Bitcoin adalah mata uang digital yang berbentuk uang elektronik yang terdesentralisasi tanpa bank sentral atau administrator tunggal yang dapat dikirim dari pengguna ke pengguna di jaringan *bitcoin peer-to-peer* tanpa perlu perantara (Calvery, 2013). *Bitcoin* sebagai media pertukaran yang terdesentralisasi memiliki lapisan tinggi dalam sistem keamanan dan penyimpan nilai yang ada di luar keuangan tradisional sistem dimana jaringan sepenuhnya *P2P* dan tidak ada perantara atau kontrol terpusat.

Bitcoin terdiri dari dua jenis struktur data yaitu transaksi dan blok. Transaksi dikelompokkan bersama dalam blok. Blok dirantai bersama melalui *hash* dari pendahulunya, sehingga membentuk struktur data yang diautentikasi, yaitu *blockchain*. Transaksi dan blok disebarluaskan di antara semua *Node* yang berpartisipasi menggunakan protokol melalui jaringan *peer-to-peer (P2P)*. Blok baru ditambahkan ke *blockchain* jika simpul jaringan dapat memberikan valid *proof-of-work (PoW)*.



Gambar 7 Arsitektur Inti *Bitcoin* (Aung, 2019)

Investasi adalah suatu transaksi atau peristiwa penjualan dan pembelian suatu sekuritas (tidak termasuk kas), bangunan maupun peralatan. Sedangkan asset lain tidak untuk diperjualbelikan (Smith & Skousen dalam Irham, 2014:8).

Python merupakan bahasa pemrograman general purpose tingkat tinggi. Diciptakan oleh *Guido van Rossum* dan dirilis pertama kali pada tahun 1991, filosofi desain *python* menekankan keterbacaan kode dengan penggunaan spasi yang signifikan. Konstruksi bahasanya dan pendekatannya berorientasi objek bertujuan untuk membantu

programmer menulis kode yang jelas dan logis untuk proyek skala kecil dan besar (Kuhlman, 2013).

Anaconda merupakan distribusi dari bahasa pemrograman *Python* dan *R* yang bersifat gratis dan *open source* untuk kebutuhan *scientific computing (data science)*, aplikasi *Machine Learning*, pemrosesan data berskala besar, analisis prediksi, dll.

Anaconda Navigator adalah antarmuka pengguna grafis desktop (*GUI*) yang termasuk dalam distribusi *Anaconda* yang memungkinkan pengguna untuk meluncurkan aplikasi dan mengelola paket *conda*, lingkungan dalam saluran tanpa menggunakan perintah *command-line*.

Jupyter Notebook (sebelumnya *Ipython Notebooks*) adalah lingkungan komputasi interaktif berbasis *web* untuk membuat dokumen *Notebook Jupyter*. Dokumen *Jupyter Notebook* adalah dokumen *JSON*, mengikuti skema versi, dan berisi daftar sel *input/output* yang diurutkan yang dapat berisi kode, teks (menggunakan *markdown*), matematika, plot, dan *rich media*, dengan akhiran ".ipynb".

Tensor Flow adalah *software library open source* yang powerful dan sangat baik untuk perhitungan numerik dan disesuaikan untuk pembelajaran *Machine Learning* berskala besar (Geron, 2017). Prinsip dasar *TensorFlow* ketika pertama kali mendefinisikan pada *Python* sebuah grafik perhitungan untuk ditampilkan, kemudian *TensorFlow* mengambil grafik tersebut dan menjalankannya secara efisien menggunakan kode *C++* yang telah dioptimalkan. *TensorFlow* merupakan *backend Deep Learning* sistem dalam membangun model prediksi maupun deskripsi yang berisi aliran data dan pemrograman terdiferensiasi di berbagai tugas, yang mana *library* ini berupa simbolis matematika yang digunakan untuk aplikasi pembelajaran seperti *Neural Network*.

Keras adalah *library Neural Network open source* yang ditulis dengan *python*. *Keras* mampu berjalan di atas *TensorFlow, Microsoft Cognitive toolking, R, Theano*, atau *PlaidML*. Dirancang untuk memungkinkan eksperimen cepat dengan *Deep Neural Network*, berfokus pada penggunaan yang ramah terhadap pengguna, modular, dan dapat dikembangkan sebagai bagian dari upaya penelitian proyek *ONEIROS (Open-ended Neuro-Electronic Intelligent robot Operating System)*.

Pandas adalah *software library* yang ditulis untuk bahasa pemrograman *Python* untuk manipulasi dan analisis data. Secara khusus menawarkan struktur data dan operasi untuk memanipulasi tabel numerik dan deret waktu. Ini adalah perangkat lunak gratis yang dirilis di bawah lisensi BSD. *Pandas* berasal dari istilah “data panel”, istilah ekonometrik untuk kumpulan data yang mencakup pengamatan selama beberapa periode waktu untuk individu yang sama (McKinney, 2018).

Numpy adalah bahasa pemrograman *python* berfungsi menambahkan dukungan untuk *array*, matriks besar, dan matriks, bersama dengan beberapa fungsi matematika tingkat tinggi untuk beroperasi pada *array*. Pendahulu *Numpy*, *Numeric*, pada awalnya diciptakan oleh Jim Hugunin dengan kontribusi dari beberapa pengembang lainnya. Pada tahun 2005, Travis Oliphant menciptakan *Numpy* dengan memasukan fitur-fitur dari *Numarray* yang bersaing ke *Numeric* dengan modifikasi ekstensif.

Matplotlib adalah librari untuk bahasa pemrograman *Python* dan ekstensi numerik matematika *Numpy*. *Matplotlib* menyediakan API berorientasi objek untuk menanamkan plot ke aplikasi menggunakan toolkit *GUI* untuk tujuan umum seperti *Tkinter*, *wxPython*, *Qt*, atau *TK +*. *Matplotlib* awalnya ditulis oleh John D. Hunter, memiliki komunitas pengembangan yang aktif, dan didistribusikan dibawah lisensi gaya BSD. *Pyplot* adalah modul *Matplotlib* yang mendediakan antarmuka seperti *MATLAB*. *Matplotlib* dirancang agar dapat digunakan seperti *MATLAB*, dengan kemampuan untuk menggunakan *Python*, dan keuntungan karena bebas dan opensource.

Scikit-Learn (sebelumnya *scikits.learn* dikenal juga sebagai *sklearn*) adalah *library Machine Learning* untuk bahasa pemrograman *Python*. Jenis fiturnya terdiri dari klasifikasi, regresi dan *algoritma clustering* termasuk *Vector Machine*, *Random Forest*, *Gradient Boosting*, *K-Means*, dan *DBSCAN*, dan dirancang untuk beroperasi dengan librari numerik dan *Science Python Numpy* dan *Scipy*.

Penggunaan model *LSTM* pada arsitektur *Recurrent Neural Network* bertujuan untuk mempelajari pola data *time series* yang panjang dan kompleks yang diharapkan mampu menyelesaikan permasalahan *vanishing gradient* maupun permasalahan lainnya ketika *RNN* biasa tidak mampu menangani data panjang dan

kompleks tersebut. Adapun perbedaan antara topik penelitian yang diambil penulis dengan beberapa penelitian dari peneliti sebelumnya diatas terletak pada teknik ataupun model, periode data *time series* dan *platform* yang digunakan. Penulis menggunakan model *LSTM* dalam membantu arsitektur *RNN* dalam memprediksi harga *bitcoin* walaupun beberapa peneliti juga menggunakan teknik *LSTM* yang sama. Sedangkan periode yang digunakan yakni periode 9 Juni 2017 – 9 September 2019 dan *platform* yang digunakan adalah *platform indodax.com*.

B. Metode Penelitian

Desain dari penelitian ini mengikuti keenam tahapan yang terdapat pada *Cross Industry Standards Process For Data Mining (CRISP-DM)* sebagaimana yang telah dijelaskan pada sub bab sebelumnya. Sedangkan metode penelitian yang digunakan yakni eksperimen. Data yang digunakan merupakan data *time series* pada periode 9 Juni 2017 – 9 September 2019. Adapun langkah dalam melakukan metode *eksperiment* ini adalah :

1. Preprocessing Data
 - a. Data Gathering/Obtaining
 - b. Data Cleansing
 - c. Data Normalization
2. Processing Data
3. Post Processing Data



Gambar 8 Alur tahapan penelitian dalam membangun sistem

C. HASIL DAN PEMBAHASAN

C.1 Fase Pemahaman Bisnis / *Business Understanding Phase*

Dalam menghasilkan profit *trader* harus mampu melakukan trading dengan mengikuti *trend* dimana ketika kondisi *uptrend* (tren naik), maka *trader* sebaiknya membuka posisi *buy*. Sebaliknya, saat

downtrend (tren turun), sebaiknya membuka posisi *sell*. Secara garis besar *trader* dapat membuka *buy* dengan kondisi pada saat harga *bitcoin* berada di berada di level terendah (*level support*), dengan ekspektasi setelah mencapai level terendah kemudian harga akan berbalik naik. Untuk posisi *sell*, sebaiknya *sell* dibuka saat harga berada di puncak (*level resistance*) dimana harga akan berbalik dari naik ke turun.

Dari hasil pengamatan dan analisis variabel data yang berpengaruh besar terhadap pergerakan harga *bitcoin* yakni *Date, Open, High, Low, Close, Volume, dan change*.

C.2 Fase Pemahaman Data / Data Understanding Phase

Fase kedua pada penelitian ini, yakni fase pemahaman data dimana pada fase ini dilakukan proses identifikasi dan pendefinisian terhadap data yang dikumpulkan. Tahapan pendefinisian dan identifikasi terhadap atribut data tersebut dilakukan dengan mengumpulkan indikator data yang terdapat pada *platform indodax.com*. Adapun data yang dikumpulkan terangkum pada tabel 4.1.

Tabel 1 Jenis dan tipe Atribut data *Bitcoin*

No	Nama Variabel Data	Jenis Atribut	Tipe Atribut
1	Date	Numeric	Interval
2	Open	Numeric	Rasio
3	High	Numeric	Rasio
4	Low	Numeric	Rasio
5	Close	Numeric	Rasio
6	Volume	Numeric	Rasio
7	Change	Numeric	Rasio

Setelah data yang dikumpulkan telah definisikan, tahapan berikutnya adalah melakukan proses *filter* terhadap variabel data dengan cara melakukan seleksi terhadap variabel data yang dirasa berpengaruh dan memiliki hubungan yang kuat antar beberapa variabel yang menjadi penentu pergerakan harga *bitcoin*. Adapun tahapan seleksi data terangkum pada table 4.2.

Tabel 2 Tahapan seleksi data indikator

No	Tahapan	
	Collecting Data	Pemilihan data
1	Mengumpulkan daftar <i>Bitcoin</i> dan <i>Altcoin</i> yang	Memilih <i>Bitcoin</i> sebagai objek penelitian dalam

	terdaftar di <i>Indodax.com</i>	memprediksi pergerakan harga <i>bitcoin</i>
2	Mengumpulkan indikator pergerakan harga <i>bitcoin</i> , seperti <i>Market, Date, Price, Volume, Change, Open, Low, High, Close</i>	Indikator yang diambil yakni <i>Date, Open, Low, High, Close, dan Volume</i>
3	Merecord data pergerakan harga <i>bitcoin</i> yakni pada tanggal 3 November 2014 – 9 September 2019	<i>Data record</i> yang diambil yakni pada tanggal 1 Januari 2017 – 9 September 2019

Setelah pengumpulan dan pemilihan data indikator selesai dilakukan langkah berikutnya adalah melakukan pengentrian data yang telah dikumpulkan berdasarkan variabel yang ditentukan sebelumnya, pengentrian data dilakukan dengan bantuan *text editor Sublime Text*. Data yang dientri tersebut berdasarkan pada periode 1 Januari 2017 – 9 September 2019 kedalam format *CSV file*. Adapun data yang telah dikolektifkan dapat dilihat pada gambar 4.1.



Gambar 9 Tampilan *file CSV Bitcoin Price* pada *Sublime Text*

C.3 Fase Penyiapan Data / Data Preparation Phase

Fase penyiapan data dimulai dengan melakukan *preprocessing* data. Data yang diolah bersumber dari record data yang berformat *CSV file*. Data tersebut diolah dengan menggunakan *Anaconda* sebagai *tools* dalam mempermudah analisis dan *eksperiment* terhadap data yang telah dikumpulkan yang digunakan dalam memprediksi harga *bitcoin*.



Gambar 10 Tampilan Anaconda Distribution

Setelah Anaconda telah terinstall, untuk dapat melakukan pengolahan data dengan mudah diperlukan beberapa library khusus seperti Tensor Flow yang merupakan Backend Deep Learning Sistem, Keras yang merupakan Front End dalam membangun Neural Network, Pandas yang berfungsi untuk mengolah data, Numpy yang berfungsi untuk operasi vektor dan

matriks, Scikit-Learn yang memiliki fitur untuk data science dan analisis data, dan Plotly yang merupakan library python untuk membuat dan menampilkan grafik. Library tersebut dapat diaktifkan pada tab menu environment yang terdapat pada Anaconda Navigator.



Gambar 11 Tampilan Anaconda Navigator

Sebelum mulai melakukan pengolahan data diperlukan adanya sebuah teks editor yang digunakan untuk menulis dan mengedit script code. Editor yang digunakan untuk pengolahan data pada studi kasus ini adalah Jupyter Notebook. Jupyter Notebook dapat diaktifkan pada Anaconda Navigator. Adapun tampilan Jupyter Notebook dapat dilihat pada gambar 12.



Gambar 12 Tampilan Editor Jupyter Notebook

Setelah beberapa requirement terpenuhi, langkah pertama pada tahapan data preparation phase yakni dengan melakukan import terhadap library python yang akan digunakan dalam membantu pengolahan data. Library tersebut antara lain Numpy, Pandas, Scipy, Pandas, Keras, Matplotlib, plotly, dll.

```
In [1]: #Import library
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from IPython.display import display
```

Gambar 13 proses import library python pada Jupyter Notebook

Langkah kedua yakni dengan melakukan import file data dengan format .CSV. Dimana file CSV tersebut bersumber dari record data sebelumnya yang ditampilkan seperti pada gambar 4.9.



Gambar 14 Tampilan file data sampel bitcoin Indodax

Selanjutnya pada tahapan ini diperlukan proses seleksi data sebagai modal untuk melakukan tahapan training dan testing dalam melakukan modeling. Dalam melakukan proses seleksi data dari beberapa indikator seperti Date, Open, Low, High, Close, dan Volume akan diseleksi indikator mana yang akan digunakan untuk proses modeling. Sebagaimana yang diketahui untuk dapat melakukan proses modeling perlu adanya proses feature scalling pada dataset. Adapun rumus feature scalling tersebut seperti terlihat pada tabel 4.3.

Tabel 3 Rumus Feature scalling

Standarisasi	Normalisasi
$X_{stand} = \frac{x - \text{mean}}{\text{standard dev}}$	$X_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$

Penggunaan rumus feature scalling menggunakan python memudahkan developer untuk tidak melakukan penginputan secara manual karena sudah tersedia pada platform Jupyter Notebook. Adapun proses feature scalling dapat dilihat pada gambar 15.

```
In [4]: #Penggunaan Feature Scalling
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

Gambar 4.15 Penggunaan *Feature scalling* pada data

4.4 Fase Pemodelan / *Modelling Phase*

Fase pemodelan dimulai dengan menginisialisasi dan mendefinisikan variabel `x_train` dan `y_train` yang akan digunakan untuk proses *training* terhadap *dataset* dimana proses *training* terhadap *dataset* merupakan sebuah proses dalam membangun model. Setelah struktur data selesai dibuat, maka tahapan berikutnya adalah melakukan proses *reshaping*. Proses diperlukan untuk memodifikasi dimensi dari *matriks* yang dihasilkan secara original ke dimensi yang diinginkan. Agar dimensi bisa *multiply-able* maka harus memiliki dimensi yang sama. Proses pembuatan struktur data dan *reshaping* pada *Jupyter Notebook* dapat dilihat pada gambar 16

```
In [3]: #membuat data struktur dengan 40 time steps dan 1 output
x_train = []
y_train = []
for i in range(40, 90):
    x_train.append(training_set_scaled[i-40:i, 0])
    y_train.append(training_set_scaled[i, 1])
x_train, y_train = np.array(x_train), np.array(y_train)

#reshaping
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
```

Gambar 16 Proses pembuatan Struktur data dan *Reshaping*

Tahapan berikutnya adalah dengan melakukan *import* terhadap beberapa *library* dan *package* yakni *Sequential*, *Dense*, *LSTM*, dan *Dropout* dimana *package* ini akan bekerja di bagian backend *TensorFlow* dalam melakukan pengolahan data. Proses *import* keempat librari *Keras* dan *Package* pada *Jupyter Notebook* dapat dilihat pada gambar 17

```
In [6]: #Import librari keras dan Package
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout

Using TensorFlow backend.
```

Gambar 17 Proses *import* model dan *layer Keras*

Tahapan berikutnya yakni dengan melakukan *import* terhadap `train_test_split` dan *Inisialisasi* terhadap *RNN* pada model dengan menggunakan fungsi `sequential`. Proses *Inisialisasi RNN* dapat dilihat pada gambar 18

```
In [7]: from sklearn.preprocessing import train_test_split

In [8]: model = Sequential()
model.add(LSTM(100, input_shape=(x_train.shape[1], x_train.shape[2])))
model.compile(optimizer='adam', loss='mse')
```

Gambar 18 Proses *Inisialisasi RNN*

Kemudian dengan menambahkan beberapa *LSTM layer* dan beberapa *Regulasi Dropout* diharapkan tahapan untuk menuju proses *training* nantinya dapat berjalan dengan baik. Adapun jumlah penambahan *Layer LSTM* berjumlah empat. Proses penambahan keempat *LSTM layer* dan beberapa *Regulasi Dropout* dapat dilihat pada gambar 19

```
In [9]: #Menambahkan layer LSTM dengan 100 neuron Regularisasi Dropout
model.add(LSTM(100, input_shape=x_train.shape[1:], recurrent_dropout=0.2, dropout=0.2))
model.add(LSTM(100, recurrent_dropout=0.2, dropout=0.2))
model.add(LSTM(100, recurrent_dropout=0.2, dropout=0.2))
model.add(LSTM(100, recurrent_dropout=0.2, dropout=0.2))

In [10]: #Menambahkan layer LSTM dengan 50 neuron Regularisasi Dropout
model.add(LSTM(50, recurrent_dropout=0.2, dropout=0.2))

In [11]: #Menambahkan layer LSTM dengan 25 neuron Regularisasi Dropout
model.add(LSTM(25, recurrent_dropout=0.2, dropout=0.2))

In [12]: #Menambahkan layer LSTM dengan 10 neuron Regularisasi Dropout
model.add(LSTM(10, recurrent_dropout=0.2, dropout=0.2))
```

Gambar 20 Proses penambahan *LSTM Layer* dan *Dropout*

Kemudian tahapan berikutnya yakni dengan menambahkan *output layer* menggunakan *model*, dimana *model* digunakan untuk memprediksi variabel target yakni dengan menambahkan *Dense* dengan unit bernilai 1. Adapun proses penambahan *output layer* dapat dilihat pada gambar 21

```
In [13]: #Menambahkan Layer Output
model.add(Dense(units = 1))
```

Gambar 21 Proses penambahan *Output Layer* pada *Jupyter Notebook*

Dari hasil tahapan tersebut kemudian dilakukan proses *compiling* terhadap *RNN* dengan menggunakan *optimizer adam*. Adapun proses dalam mengcompiling *RNN* dapat dilihat pada gambar 11

```
In [14]: model.compile(optimizer='adam', loss='mse')
```

Gambar 11 Proses *Compiling RNN*

Tahapan berikutnya yakni dengan melakukan *fitting* terhadap *RNN* ke *training set* pada variabel `x_train` dan `y_train` dengan nilai *epoch* yang diset bernilai 100, *validation_split* bernilai 0.2 yang artinya 20% untuk proses validasi sampel dan 80% residual untuk proses training sampel, serta *batch size* bernilai 32. Adapun proses *fitting RNN* kedalam *training set* dapat dilihat pada gambar 23


```
In [16]: #Membuat data training dataset
dataset_train = df[['Open', 'High', 'Low', 'Close', 'Volume']]
dataset_train = dataset_train.dropna()
dataset_train = dataset_train.reset_index(drop=True)
dataset_train = dataset_train.to_numpy()

#Membuat data validasi
dataset_val = df[['Open', 'High', 'Low', 'Close', 'Volume']]
dataset_val = dataset_val.dropna()
dataset_val = dataset_val.reset_index(drop=True)
dataset_val = dataset_val.to_numpy()

#Membuat data testing
dataset_test = df[['Open', 'High', 'Low', 'Close', 'Volume']]
dataset_test = dataset_test.dropna()
dataset_test = dataset_test.reset_index(drop=True)
dataset_test = dataset_test.to_numpy()
```

Gambar 23 Proses *Fitting RNN* kedalam *training set*



Gambar 27 Proses dalam menampilkan visualisasi hasil prediksi

4.5 Fase Evaluasi /Evaluation Phase

Tahapan pertama pada fase ini yakni dengan melakukan proses evaluasi terhadap model untuk mengetahui seberapa tinggi tingkat akurasi model dalam memprediksi pergerakan harga bitcoin. Dalam proses ini didapatkan akurasi model training memiliki tingkat akurasi sebesar 0.11% yang artinya model tersebut bernilai cukup baik.

```
In [18]: #Evaluasi akurasi model training
scores = model.evaluate(x_train, y_train)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

922/922 [*****] - 6s 6ms/step
acc: 0.11%
```

Gambar 24 Proses Evaluasi model *RNN*

Tahapan selanjutnya yakni tahapan prediksi dan visualisasi terhadap data lama dengan menggunakan hasil *training dataset*. Adapun proses membuat prediksi dapat dilihat pada gambar 25

```
In [17]: #Membuat prediksi menggunakan data harga bitcoin
dataset_test = df[['Open', 'High', 'Low', 'Close', 'Volume']]
dataset_test = dataset_test.dropna()
dataset_test = dataset_test.reset_index(drop=True)
dataset_test = dataset_test.to_numpy()
```

Gambar 25 Proses membuat prediksi dan hasil visualisasi

Tahapan berikutnya yakni mendapatkan hasil prediksi pergerakan harga *bitcoin*. Adapun proses dalam mendapatkan hasil prediksi dapat dilihat pada gambar 26

```
In [18]: #Membuat hasil prediksi Bitcoin
dataset_total = pd.concat([df[['Open']], df[['Open']], axis = 0)
inputs = dataset_total[:(len(dataset_total) - len(df) - 60)].values
inputs = np.reshape(inputs, (-1,))
inputs = np.transpose(inputs)
x_test = []
for i in range(60, 1000):
    x_test.append(inputs[i-60:i, 0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1]))
predicted_stock_price = model.predict(x_test)
predicted_stock_price = ac.inverse_transform(predicted_stock_price)
```

Bitcoin

Tahapan terakhir adalah dengan memvisualisasikan hasil. Adapun proses dalam memvisualisasikan hasil prediksi dapat dilihat pada gambar 27

D. KESIMPULAN

Berdasarkan hasil eksperimen dan analisis, kesimpulan dari penelitian ini dapat disimpulkan menjadi beberapa point. Pertama, dalam melakukan pengkorelasi antar variabel, data yang tercatat pada variabel *Date*, *Open*, *High*, *Low*, *Close*, dan *Volume* dilakukan proses pengimporan kedalam *Jupyter Notebook*. Tahapan berikutnya dengan melakukan *feature scalling* pada data. Data kemudian diinisialisasi dan didefinisikan ke dalam variabel *x_train* dan *y_train* yang digunakan untuk proses *training* terhadap *dataset* dan digunakan membuat data struktur dengan 60 *timestep* dan 1 *output* untuk menyimpan data dalam bentuk yang efisien yang kemudian digunakan dalam proses *modeling*. Selanjutnya penggunaan *reshaping* dalam memodifikasi dimensi dari *matriks* yang dihasilkan ke dimensi yang diinginkan. Kemudian dilakukan proses pengimporan terhadap *library Keras* dan *packagenya* yakni *Sequential*, *Dense*, *LSTM*, dan *Dropout*. Tahapan berikutnya adalah dengan melakukan proses *Inisialisasi RNN* dan penambahan empat *layer LSTM* dan beberapa regulasi *Dropout*. Tahapan berikutnya yakni menambahkan *layer output* dan melakukan *compile* terhadap *RNN* dengan menggunakan *optimizer big data* yakni *adam*. Kemudian dilakukan *fitting* terhadap *RNN* ke *training set* pada variabel *x_train* dan *y_train* dengan nilai *epoch* = 100 dan *batch size* = 32. Dari tahapan *fitting* kemudian membuat prediksi dengan mendapatkan data harga *bitcoin* pada *file CSV* sebelumnya dan hasil yang diperoleh dalam prediksi *bitcoin* kemudian divisualisasikan. Kedua, dalam membangun model prediksi pergerakan harga bitcoin menggunakan algoritma dan Arsitektur *RNN* dan varian *RNN* yakni *LSTM* dalam membangun model untuk memprediksi

Bitcoin. Ketiga tingkat akurasi model prediksi yang diperoleh dari tahapan evaluasi yakni 0.11%

diakses tanggal 12 Agustus 2019, dari <https://medium.com/@cloudycalvin/overview-of-bitcoin-architecture-cb3c88a1b20a>

DAFTAR PUSTAKA

AGULLar, Loyo J.S. (2017). *Forecasting volatility using Artificial Neural Networks and parametric methods*. Dikutip 7 Agustus 2018, dari <http://www.scripstonesonline.uba.uva.nl/document/654916>

Ashrovy, Ron. (17 Oktober 2017). *Recurrent Neural Network Part Four (END)*. Dikutip tanggal 22 Agustus 2019 dari <https://medium.com/@ashrovy/recurrent-neural-network-part-4-d371474b8fa9>

Aung, Sithu. (2019). *Bitcoin Architecture Core*. dikutip tanggal 23 Agustus 2019, dari <https://id.pinterest.com/pin/551128073142441517/>

Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (8): 1798–1828.

Berry, M. and Linoff, G., (1999). *Mastering Data Mining : The art and science of customer relationship management*. John Wiley & Sons, Inc..

Berry, M.J. and Linoff, G.S., (2004). *Data Mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). *Bitcoin: Economics, technology, and governance*. *Journal of Economic Perspectives*, 29(2), 213-38.

Böhme, Rainer, et al. "Bitcoin: Economics, technology, and governance." *Journal of Economic Perspectives* 29.2 (2015): 213-38.

Cheng, Calvin (9 November 2017), "Overview of Bitcoin Architecture",

Bre, Facundo (2017, November). *Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks*. Dikutip tanggal 7 Agustus 2018, dari https://www.researchgate.net/publication/321259051_Prediction_of_wind_pressure_coefficients_on_building_surfaces_using_Artificial_Neural_Networks

Britz, Denny. (17 September 2015), "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs", diakses tanggal 7 Agustus 2018, dari <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-RNNs/>

Calvery, Jennifer Shasky (statement), *Director Financial Crime Enforcement Network United States Department of the Treasury Before the United States Senate Committee on Banking, Housing, and Urban Affairs Subcommittee on National Security and International Trade and Finance Subcommittee on Economic Policy* (PDF). [fincen.gov](https://www.fincen.gov). *Financial Crimes Enforcement Network*. (19 November 2013).

Ciresan, Dan; Meier, U.; Schmidhuber, J. (June 2012). "Multi-column Deep Neural Networks for image classification". 2012 *IEEE Conference on Computer Vision and Pattern Recognition*: 3642–3649

Crosby, M., Pattanayak, P., Verma, S. and Kalyanaraman, V., (2016). *Blockchain technology: Beyond Bitcoin*. *Applied Innovation*, 2(6-10), p.71.

Dourado, Eli dan Jerry Brito. (2014). *Cryptocurrency*. The New Palgrave.

Elman, Jeffrey L. (1990). "Finding Structure in Time". *Cognitive Science*. **14** (2): 179–211. doi:10.1016/0364-0213(90)90002-E.

- Han, J., Pei, J. and Kamber, M., (2011). *Data Mining : concepts and techniques*. Elsevier.
- Herdianto. 2013. *Prediksi Kerusakan Motor Induksi Menggunakan Metode Jaringan Saraf Tiruan Backpropagation*. Tesis. Universitas Sumatera Utara : Medan
- Hileman, G. and Rauchs, M., (2017). *Global Cryptocurrency benchmarking study*. Cambridge Centre for Alternative Finance, 33.
- Iansiti, Marco; Lakhani, Karim R. (January 2017). "The Truth About Blockchain". Harvard Business Review. Harvard University
- Jacobs, E., (2011). *Bitcoin: a bit too far?*. *Journal of Internet Banking and Commerce*, 16(2), p.1.
- Kuhlman, Dave(2013). "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1..
- Kurihara, Yutaka, and Akio Fukushima. "The market efficiency of Bitcoin: A weekly anomaly perspective." *Journal of Applied Finance and Banking* 7.3 (2017): 57.
- Larose D, T., (2005). *Discovering knowledge in data : an introduction to Data Mining*, Jhon Wiley & Sons Inc.
- Larose, Daniel T., and Larose, Chantal D. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining Second Edition*. New Jersey: John Wiley & Sons Inc.
- Lustig, Caitlin, and Bonnie Nardi. "Algorithmic authority: The case of Bitcoin." *2015 48th Hawaii International Conference on System Sciences*. IEEE, (2015).
- Wes McKinney (2011). "*Pandas: a Foundational Python Library for Data Analysis and Statistics*". Dikutip tanggal 18 November 2019 pada ebook scribd.com
- Narayanan, A., Bonneau, J., Felten, E., Miller, A. and Goldfeder, S., (2016). *Bitcoin and Cryptocurrency technologies: A comprehensive introduction*. Princeton University Press.
- Ni, Xianjun. "Research of Data Mining based on Neural Networks." *World Academy of Science, Engineering and Technology* 39 (2008): 381-384.
- Nian, L.P. and Chuen, D.L.K., (2015). *Introduction to Bitcoin*. In *Handbook of Digital Currency* (pp. 5-30). Academic Press.
- Olah, Christoper (27 Agustus 2015), "Understanding LSTM Network". Dikutip tanggal 5 Desember 2019, dari <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Olson, D.L. and Delen, D., (2008). *Advanced Data Mining techniques*. Springer Science & Business Media.
- Pilkington, M., (2016). *Blockchain technology: principles and applications*. *Research handbook on digital transformations*, 225.
- Primartha, Rifkie. (2018). "*Belajar Machine Learning Teori dan Praktik*". Bandung : Informatika Bandung.
- Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. **61**: 85–117
- Schueffel, Patrick (2017). *The Concise Fintech Compendium*. Fribourg: School of Management Fribourg/Switzerland. Archived from the original on 24 October 2017.
- SOVBETOV, Yhlas. (2018). *Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litecoin, and Monero*. *Journal of Economics and Financial Analysis*, Vol:2, No:2 (2018) 1-27
- Siegel, Eric (2013). *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die* (1st ed.). Wiley. ISBN 978-1-1183-5685-2.

Yelowitz, A. and Wilson, M., (2015).
*Characteristics of Bitcoin users: an
analysis of Google search
data. Applied Economics
Letters*, 22(13), pp.1030-1036.