

Peramalan *Close Price* Mata Uang *Crypto* Solana Menggunakan Jaringan Syaraf Tiruan Model *Backpropagation*

Mochammad Rizal¹, Wiyli Yustanti²

^{1,2} Sistem Informasi, Teknik Informatika, Universitas Negeri Surabaya

¹mochammad.19048@mhs.unesa.ac.id

²wiyliyustanti@unesa.ac.id

Abstrak— Beberapa tahun terakhir, *crypto* menjadi instrumen keuangan yang paling menarik. Solana, sebagai salah satu mata uang *crypto* yang berkembang pesat, menawarkan kecepatan dan efisiensi. Dalam hal ini, peramalan harga Solana menjadi hal yang penting untuk melakukan investasi. Dalam penelitian ini, peneliti mencoba menggunakan dan membandingkan penggunaan jaringan syaraf tiruan *backpropagation* dengan ARIMA dengan mencari parameter dan nilai optimal dari masing-masing pemodelan melalui hasil akurasi MAPE dan MSE yang terkecil dengan tools MATLAB. Hasil penelitian ini menunjukkan bahwa jaringan syaraf tiruan *backpropagation* yang memakai beberapa parameter seperti variabel input memakai High dan Low Price, transfer function purelin, training function traingdx, momentum sebesar 0.3, learning rate sebesar 0.2 dan node pada *hidden layer* sebanyak 4 node menghasilkan akurasi MAPE dan MSE yang lebih baik yaitu 98.9186% dan 1190.01 daripada model ARIMA yang memakai nilai paling optimal yaitu senilai 0, 1, 0 yang sebesar 69.9537% dan 30583.

Kata Kunci — Peramalan, Solana, Jaringan Syaraf Tiruan, *Backpropagation*, ARIMA, MAPE, MSE

I. PENDAHULUAN

Sejak munculnya bitcoin pada tahun 2009 sebagai pelopor mata uang *crypto* semakin tahun banyak mata uang *crypto* yang hadir sebagai pilihan alternatif sebagai tujuan investasi yang memiliki inovasi dan teknologi yang diusung, Solana adalah salah satu alternatif mata uang *crypto* yang hadir didalamnya untuk Solana fitur yang menonjol ialah skalabilitasnya dengan kemampuan untuk memproses lebih dari 50.000 transaksi per detik (TPS) [1].

Pada awal kemunculan Solana yaitu tahun 2017 melalui *whitepaper* dari Anatoly Yakovenko yang berjudul “Solana: A new architecture for a high performance blockchain” [2]. Dan pada 16 Maret 2020 blok pertama dari Solana dibuat dan menandai dimulai nya perjalanan mata uang *crypto* Solana. Setiap mata uang *crypto* memiliki satu kesamaan yaitu perubahan harga harian yang selalu berubah-ubah.

Dengan harganya yang berubah-ubah dengan cepat maka menjadi kekhawatiran bagi investor Solana. Karena sifat dari mata uang kripto itu sendiri ialah naik atau turun nya harga nya

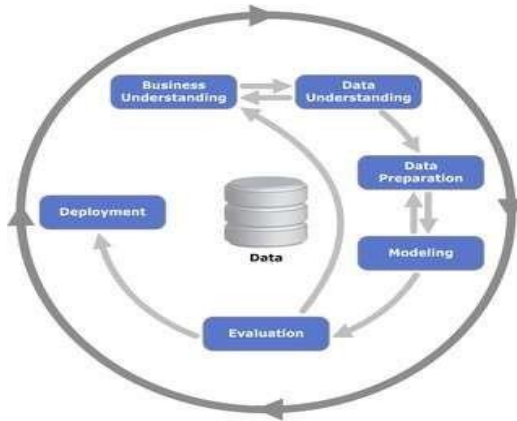
ditentukan oleh pemilik mata uang kripto tersebut dan tidak dikendalikan seperti mata uang konvensional yang dikendalikan oleh bank sentral. Dari ketidakpastian harga ini yang tidak disukai oleh para investor yang dapat menyebabkan resiko seperti kehilangan total aset yang disediakan sebagai modal investasi, akan tetapi dalam sebuah investasi terdapat resiko yang tidak dapat dihindarkan. Untuk mencegah resiko dalam berinvestasi dapat menggunakan teknik peramalan (*forecasting*) untuk meramalkan harga naik atau turun dari harga Solana. Peramalan ini digunakan dengan memprediksi sesuatu yang akan terjadi di masa depan dengan menggunakan data yang ada dari masa lalu agar pengguna dapat menentukan keputusan yang benar dan mendapatkan keuntungan dari pengambilan keputusan tersebut berdasarkan data peramalan.

Seiring dengan kemajuan ilmu pengetahuan muncul metode untuk melakukan peramalan atau prediksi harga. Dengan menggunakan metode *machine learning* melalui data *training* harga Solana untuk terlebih dahulu mempelajari pola yang terbentuk dari perubahan harga Solana dengan menggunakan algoritma yang dapat mempelajari pola yang terbentuk. Algoritma tersebut menghasilkan fungsi pendekatan yang menyambungkan variabel-variabel input-nya terhadap variabel output-nya. Variabel input dapat berupa data *open price*, *high price* dan *low price*. Variabel output dapat digunakan data *close price* [4].

Penelitian ini melakukan peramalan harga dengan menemukan model jaringan syaraf tiruan model *backpropagation* yang memiliki hasil hubungan regresi yang baik dari beberapa input variabel yang beragam serta melakukan perbandingan keakuratan dari masing-masing variabel lalu hasil terbaik dari model jaringan syaraf tiruan *backpropagation* akan dibandingkan dengan penemuan model optimal metode ARIMA dengan melakukan uji keakuratan dengan membandingkan hasil MAPE (*Mean Absolute Percentage Error*) dan nilai MSE (*Mean Squared Error*) dan hasilnya akan diperbandingkan untuk membantu trader/investor Solana yang akan membeli Solana.

II. METODE PENELITIAN

Pada penelitian ini menggunakan metode CRISP-DM (*Cross Industry Standard Process for Data Mining*) yang dijabarkan oleh Tuga dan Faisal [3]. Berikut ini grafik alur tahapannya :



Gbr. 1 Grafik alur tahapan CRISP-DM

A. Business Understanding

Business Understanding adalah tahapan yang memerlukan pemahaman tujuan dan kebutuhan dari sudut pandang penelitian dan memperoleh data serta menyesuaikan dengan tujuan penelitian untuk mencapai tujuan yang diinginkan.

B. Data Understanding

Data Understanding adalah tahapan untuk memenuhi kondisi dataset serta mengidentifikasi masalah yang mungkin ada dalam dataset tersebut. Data yang digunakan dalam penelitian ini yakni data *open price*, *low price*, *high price* dan *close price crypto* Solana yang diambil dari website (<https://coinmarketcap.com>) dalam kurun periode harian mulai dari 10 April 2020 hingga 07 Juli 2023.

C. Data Preparation

Data Preparation adalah tahapan data mentah menuju data matang yang berguna dalam penelitian. Berikut ini adalah tahapan *Data Preparation* yang dilakukan pada penelitian ini :

- 1) Melakukan uji korelasi
- 2) Melakukan normalisasi data
- 3) Melakukan pengelompokan variabel input kedalam skenario untuk pemodelan jaringan syaraf tiruan backpropagation.
- 4) Membagi dataset training dan testing.

Hasil dari *Data Preparation* diatas akan digunakan untuk tahapan selanjutnya.

D. Modelling

Modelling adalah tahapan membuat model prediktif dan deskriptif, pada tahap ini dilakukan penggunaan *Machine Learning* untuk menentukan model atau algoritma yang terbaik untuk proses prediksi. Pada penelitian ini, proses modelling akan dilakukan dengan beberapa tahapan berikut ini :

1. Pemodelan Jaringan Syaraf Tiruan Backpropagation

Dalam proses ini, dimulai dari membuat arsitektur jaringan syaraf tiruan backpropagation berdasarkan model skenario yang dipakai, melakukan tuning hyperparameter untuk menemukan parameter yang memiliki hasil MAPE dan MSE yang optimal, melakukan perancangan model jaringan syaraf

tiruan menggunakan tools MATLAB, melakukan proses training dan testing dan terakhir membandingkan semua skenario yang telah di ujicoba dengan melihat nilai MAPE dan MSE yang terkecil.

2. Pemodelan ARIMA

Dimulai dengan melakukan differencing data untuk melihat apakah data yang dipakai mengalami perubahan sepanjang waktu berjalan (tidak berfluktuasi pada nilai tengah yaitu 0). Setelah melakukan differencing data dilakukan pencarian nilai Q, pencarian nilai P. Sedangkan untuk nilai D didapatkan dari seberapa banyaknya dilakukan proses differencing data. Setelah mendapatkan nilai P,D, Q yang optimal maka selanjutnya dilakukan proses training dan testing dihasilkan nilai MAPE dan MSE.

E. Evaluation

Evaluation adalah tahapan komparasi hasil dari pemodelan yang dilakukan dengan membandingkan nilai MAPE dan MSE yang paling kecil dari pemodelan yang digunakan serta ditampilkan grafik pembandingan dengan data actual untuk melihat perbedaan hasil dari pemodelan yang dipakai. Kemudian digunakan juga untuk melakukan peramalan untuk harga 1 hari kemudian atau harga tanggal 8 Juli 2023.

F. Deployment

Deployment merupakan rencana penggunaan model terbaik dan mengabungkan dengan keputusan dalam sistem operasi. Meskipun model sudah digunakan tetapi model ini juga perlu untuk dipantau diganti dengan model yang lebih baik dimasa mendatang. Finalisasi model merupakan tahapan akhir untuk menyesuaikan model terbaik kedalam dataset yang utuh, termasuk test set. Memiliki tujuan untuk melatih model pada dataset yang lengkap sebelum disebarluaskan untuk diproduksi.

III. HASIL DAN PEMBAHASAN

Bagian ini membahas mengenai hasil dari setiap tahapan yang sudah dijabarkan pada bagian sebelumnya mulai dari *Business Understanding* sampai dengan tahap *Deployment*.

A. Business Understanding

Penelitian ini mengangkat permasalahan yang ada yaitu untuk melakukan peramalan *close price* mata uang crypto Solana dengan melakukan perbandingan dua pemodelan yang digunakan untuk melakukan peramalan harga *crypto* Solana. Dari pengujian pemodelan ini diharapkan dapat digunakan sebagai bahan acuan untuk melakukan investasi yang berujung mendapatkan keuntungan bagi yang menggunakan metode ini.

B. Business Understanding

Data yang digunakan dalam penelitian ini adalah data yang berasal dari website (<https://coinmarketcap.com>). Dalam dataset tersebut terdapat beberapa batasan:

- 1) Scope Harga : *Open price*, *High price*, *Low price* dan *Close price*.
- 2) Periode Waktu : 10 April 2020 hingga 07 Juli 2023.

Beberapa informasi mengenai dataset tersebut antara lain:

- 1) Dataset ini menggunakan data harian yang meliputi data *open price*, *high price*, *low price* dan *close price*.
- 2) Dataset ini menggunakan harga dalam mata uang rupiah.

C. Data Preparation

1. Melakukan uji korelasi

Dari data yang didapatkan, dilihat hubungan antara *open price*, *high price*, *low price* yang akan digunakan sebagai variabel input dengan *close price* sebagai variabel output untuk melihat hubungan antar variabel yang dipakai. Berikut ini hasil uji korelasi :

TABEL III
HASIL UJI KORELASI

Open	High	Low
0.9970	0.9988	0.9883

2. Melakukan normalisasi data

Setelah melihat hubungan variabel input dengan variabel output dilakukan proses normalisasi data untuk memudahkan proses komputasi dalam melakukan peramalan. Normalisasi data ini menggunakan rumus :

$$X' = \frac{0,8 (X - b)}{(a - b)} + 0,1$$

Keterangan :

- X' = Data hasil normalisasi
X = Data asli/awal
a = nilai maksimum data awal
b = nilai minimum data awal

TABEL IV
DATA SEBELUM DINORMALISASI

Tanggal	Open	High	Low	Close
10-Apr-2020	13298	20759	10891	15028
11-Apr-2020	15028	16577	12088	12275
12-Apr-2020	14082	14096	12172	12335
...	...	...	...	...
07-Jul-2023	299765	327518	292572	325744

TABEL V
DATA SETELAH DINORMALISASI

Tanggal	Open	High	Low	Close
10-Apr-2020	0.1012	0.1026	0.1007	0.1015
11-Apr-2020	0.1015	0.1017	0.1010	0.1009
12-Apr-2020	0.1010	0.1014	0.1010	0.1013
...	...	...	...	...
07-Jul-2023	0.1631	0.1687	0.1648	0.1687

3. Melakukan pengelompokan berdasarkan variabel input.

Setelah dilakukan proses normalisasi data, maka selanjutnya adalah mengelompokkan berdasarkan variabel input kedalam beberapa skenario untuk digunakan untuk melihat skenario mana yang memiliki hasil MAPE, MSE dan regresi yang paling baik. Berikut ini adalah skenario yang akan digunakan dalam pencarian model terbaik dari pemodelan jaringan syaraf tiruan *backpropagation* :

TABEL VI
SKENARIO

Skenario	Input	Output
1	Open price	Close price
2	High price	Close price
3	Low price	Close price
4	Open, high price	Close price
5	Open, low price	Close price
6	High, low price	Close price
7	Open, high, low price	Close price

4. Membagi dataset training dan testing

Setelah dilakukan pembagian skenario dengan input yang berbeda-beda, selanjutnya dilakukan pembagian dataset untuk proses training dan testing dengan rasio pada table dibawah ini :

TABEL VI
PEMBAGIAN DATA TRAINING DAN TESTING

Keterangan	Data Training	Data Testing	Total
Presentase	80%	20%	100%
Jumlah	947	237	1184

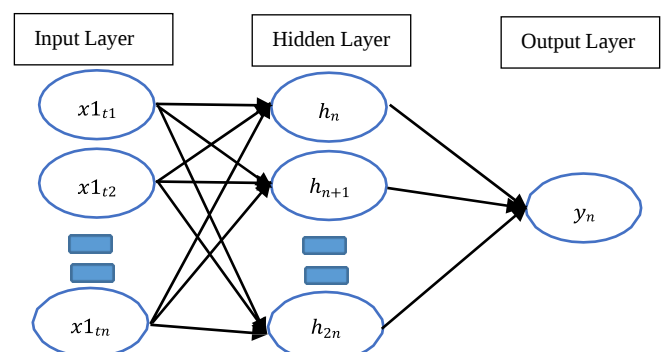
D. Modelling

1. Melakukan pembuatan model arsitektur jaringan syaraf tiruan

a) Pembuatan arsitektur jaringan syaraf tiruan

Pembuatan model ini bertujuan untuk melihat bagaimana arsitektur model yang digunakan dalam penelitian ini. Setelah melihat dari skenario yang dihasilkan dari variabel input yang digunakan, maka dibuat tiga model arsitektur jaringan syaraf berdasarkan jumlah variabel input.

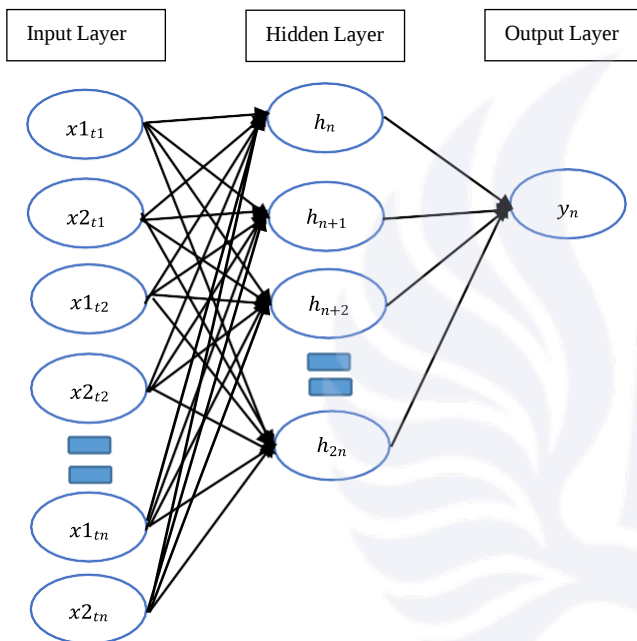
Untuk skenario 1, 2 dan 3 yang menggunakan satu variabel input, maka model arsitektur yang digunakan adalah sebagai berikut :



Gbr. 2 Arsitektur JST skenario 1,2 dan 3

Arsitektur ini menggunakan struktur jaringan syaraf tiruan yang terdiri dari satu node input dari variabel input yang digunakan dalam skenario 1, 2 dan 3, hidden layer terdiri dari satu layer yang berisi fungsi aktivasi dan terdapat node dengan kirsaran n hingga $2n$, dimana n adalah jumlah variabel input yang digunakan dalam tiap-tiap skenario. Jumlah ini dipilih agar mempermudah proses pencarian nilai optimal agar tidak terlalu banyak menggunakan node pada *hidden layer*. Output layer terdiri dari satu node yaitu *close price* sebagai variabel output.

Untuk arsitektur dari skenario 4, 5 dan 6 yang menggunakan dua variabel input maka untuk model arsitektur jaringan syaraf tiruan sebagai berikut :

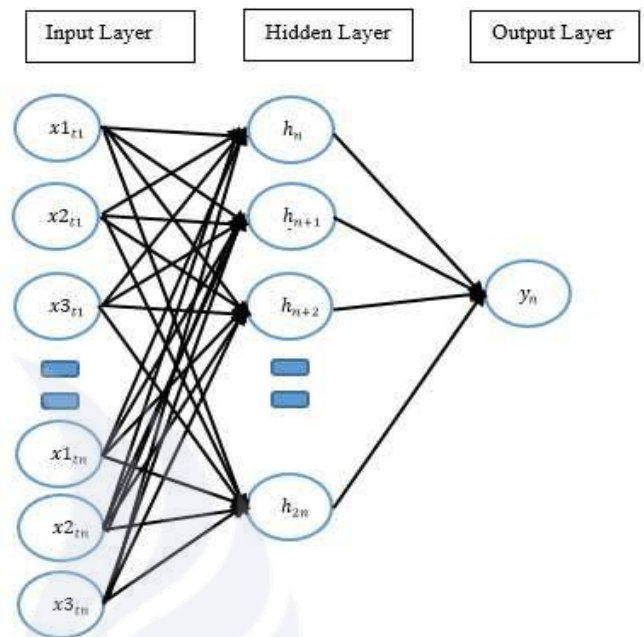


Gbr. 3 Arsitektur JST skenario 4, 5 dan 6

Arsitektur ini menggunakan struktur jaringan syaraf tiruan yang terdiri dari satu node input dari variabel input yang digunakan dalam skenario 4, 5 dan 6, hidden layer terdiri dari satu layer yang berisi fungsi aktivasi dan terdapat node dengan kirsaran n hingga $2n$, dimana n adalah jumlah variabel input yang digunakan dalam tiap-tiap skenario. Jumlah ini dipilih agar mempermudah proses pencarian nilai optimal agar tidak terlalu banyak menggunakan node pada *hidden layer*. Output layer terdiri dari satu node yaitu *close price* sebagai variabel output.

Untuk arsitektur dari skenario 7 yang menggunakan tiga variabel input ini terdiri dari tiga node input dari skenario 7. Hidden layer terdiri dari satu layer yang berisi fungsi aktivasi dan terdapat node dengan kirsaran n hingga $2n$, dimana n adalah jumlah variabel input yang digunakan dalam tiap-tiap skenario. Jumlah ini dipilih agar tidak terlalu banyak supaya merampingkan saat proses komputasi. Output layer terdiri dari satu node yakni variabel output *close price*.

Berikut ini tampilan arsitektur jaringan syaraf tiruan backpropagation yang digunakan untuk skenario 7 yang memiliki tiga variabel input:



Gbr. 4 Arsitektur JST skenario 7

b) Melakukan tuning hyperparameter

Proses tuning hyperparameter ini digunakan untuk menentukan parameter apa saja yang digunakan pada jaringan syaraf tiruan. Parameter tersebut yang akan digunakan dengan mengkombinasikan parameter satu sama lain untuk mencapai nilai akurasi MAPE dan MSE yang paling baik. Parameter yang digunakan terdiri dari training function, learning function, transfer function, momentum, learning rate, jumlah node pada hidden layer. Untuk penjabaran rancangan parameter akan ditampilkan pada table berikut ini :

TABEL VII
PARAMTER JARINGAN SYARAF TIRUAN

Parameter	Nilai	Deskripsi
Hidden layer	1,2,...,2n	Trial error
Epoch	70	Trial error
Momentum	0.1 hingga 0.9	Trial error
Learning rate	0.1 hingga 0.9	Trial error
Transfer function	3 (purerlin, logsig, tansig)	Trial error
Training function	3 (traingda, traingdx, trainlm)	Trial error
Output layer	1 Neuron	Nilai close price

c) Perancangan model jaringan syaraf tiruan

Untuk perancangan model jaringan syaraf tiruan backpropagation akan menggunakan aplikasi MATLAB. Pertama yang dilakukan adalah menentukan parameter

yang digunakan pada jaringan syaraf tiruan dengan perintah sebagai berikut :

```
%Parameter Yang digunakan
transferF ={'purelin'};
trainingF ={'traingdx'};
learningF ={'learnngdm'};
momentum =[0.3];
epoch =[70];
learningrateF =[0.2];
```

Gbr. 5 Sumber kode deskripsi parameter

Dalam script diatas, dijelaskan beberapa parameter yang dipakai untuk proses training dan testing. transferF berfungsi untuk menjelaskan fungsi transfer apa yang dipakai, trainingF berfungsi untuk penentuan fungsi training, learningF berfungsi untuk penentuan fungsi pembelajaran, momentum untuk mendeskripsikan array pada fungsi momentum, epoch untuk mendeskripsikan array pada fungsi epoch dan learningrateF untuk mendeskripsikan array pada fungsi learning rate.

Tahapan selanjutnya adalah pembuatan network atau jaringan yang akan digunakan untuk proses training. Dalam pembentukan jaringan syaraf tiruan ini menggunakan beberapa inputan seperti variabel input, parameter, output, melalui perintah sebagai berikut :

```
for node=4:4
net=newff(iTraining,tTraining,node,{cell2mat(t
ransferF),'purelin'});
net.trainFcn=cell2mat(trainingF);
net.trainParam.lr=learningrateF;
net.trainParam.mc=momentum;
net.trainParam.epochs=epoch;
net.trainParam.max_fail=1000;
net.layerWeights{1,1}.learnFcn=cell2mat(learn
ingF);
```

Gbr. 6 Sumber kode pembuatan jaringan syaraf tiruan

Setelah membuat jaringan ini, dilakukan pembagian data training dan testing seperti yang sudah dijelaskan. Selanjutnya dilakukan proses training dan testing pada data dengan perintah sebagai berikut :

```
[netTraining,tr]=train(net,iTraining,
tTraining);
outputTest=sim(netTraining,iTesting);
outputTrain=netTraining(iTraining);
```

Gbr. 7 Sumber kode training dan testing

Setelah dilakukan proses training dan testing, selanjutnya adalah mengukur akurasi dengan menggunakan perhitungan MAPE dan MSE. Setelah melakukan ini dilakukan proses percobaan dengan menggunakan skenario yang telah dibuat beserta dengan percobaan kombinasi parameter-parameter yang telah dijabarkan dan untuk node pada hidden layer akan ditampilkan dari n hingga 2n, dimana n adalah jumlah dari variabel input yang digunakan dan akan ditampilkan dari menggunakan node paling kecil hingga

yang telah ditentukan. Sebelum melakukan percobaan berikut ini perintah untuk menghitung MAPE dan MSE :

```
%Testing MAPE
mapeTest=(abs(tTesting-outputTest))./tTesting;
mapeTest=mean(mapeTest);
mapeTest=mapeTest *100;
Akurasi=100-mapeTest;
%Testing MSE
mseTrain=(tTest-outputTest).^2;
mseTrain=mean(mseTrain);
```

Gbr. 8 Sumber kode menghitung MAPE dan MSE

- Skenario 1

Skenario ini menggunakan variabel input yaitu open price yang artinya menggunakan satu node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 2 node hidden layer. Hasil model dari skenario 1 akan ditampilkan pada tabel berikut ini :

TABEL VIII
HASIL SKENARIO 1

No	Fungsi Aktivasi	M c	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingda	0.1	0.1	1	1.9987	1213.12	0.997
2	Logsig – Traingdx	0.3	0.4	1	4.2302	4312.23	0.983
3	Tansig – Trainlm	0.3	0.2	1	1.5391	1673.78	0.996
4	Purelin – Traingdx	0.3	0.3	2	1.5389	1668.15	0.996
5	Logsig – Traingda	0.2	0.3	2	7.8101	7923.78	0.965
6	Tansig – trainlm	0.2	0.1	2	1.5501	1681.95	0.996

Hasil skenario 1 yang menggunakan open price sebagai variabel input dapat disimpulkan bahwa model no 4 dengan kombinasi paramater seperti transfer function menggunakan purelin, training function menggunakan traingdx, momentum sebesar 0.3, learning rate sebesar 0.3 dan node pada hidden layer sebanyak 2 node sehingga menghasilkan nilai MAPE dan MSE yang paling kecil daripada kombinasi parameter lainnya.

- Skenario 2

Skenario ini menggunakan variabel input yaitu *high price* yang artinya menggunakan satu node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 2 node hidden layer. Dari hasil skenario 2 yang menggunakan *high price* sebagai variabel input memiliki hasil MAPE dan MSE terbaik ditemukan dengan menggunakan kombinasi parameter training function menggunakan purelin, training function menggunakan traingda, momentum sebesar 0.1, learning rate sebesar 0.1 dan node hidden layer sebanyak 2 node. Berikut ini hasil percobaan

parameter untuk skenario 2 dalam bentuk tabel nomer IX (sembilan).

TABEL IX
HASIL SKENARIO 2

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingda	0.2	0.1	1	2.6773	2773.12	0.991
2	Logsig – Traingdx	0.3	0.5	1	4.0299	4125.68	0.984
3	Tansig – Traingdx	0.2	0.3	1	5.0616	5160.98	0.979
4	Logsig – Traingda	0.1	0.4	2	6.3295	6492.54	0.973
5	Purelin – Traingda	0.1	0.1	2	1.2462	1343.26	0.998
6	Tansig – Trainlm	0.3	0.2	2	1.6828	1782.45	0.996

- Skenario 3

Skenario ini menggunakan variabel input yaitu *low price* yang artinya menggunakan satu node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 2 node hidden layer. Hasil model dari skenario 3 akan ditampilkan pada tabel berikut ini :

TABEL X
HASIL SKENARIO 3

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingdx	0.1	0.4	1	1.5737	1683.21	0.996
2	Logsig – Traingda	0.2	0.1	1	7.3431	7458.29	0.967
3	Tansig – Trainlm	0.3	0.5	1	1.7500	1845.32	0.995
4	Purelin – Traingda	0.1	0.6	2	1.4769	1563.54	0.997
5	Logsig – Trainlm	0.2	0.5	2	1.7287	1802.46	0.995
6	Tansig – Traingda	0.3	0.1	2	5.5939	5691.67	0.977

Hasil skenario 3 yang menggunakan open price sebagai variabel input dengan menggunakan kombinasi parameter seperti no 4 dengan fungsi aktivasi menggunakan Purelin sebagai transfer function, traingda sebagai training function, momentum sebesar 0.1, learning rate sebesar 0.6 dan node hidden layer sebanyak 2 node menghasilkan akurasi MAPE dan MSE yang lebih baik daripada kombinasi parameter yang lain.

- Skenario 4

Skenario ini menggunakan variabel input yaitu open dan high price yang artinya menggunakan satu node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 4 node hidden layer. Dari hasil tabel ini ditemukan bahwa kombinasi parameter

no 10 memiliki hasil MAPE dan MSE yang paling baik, berikut ini tabel hasil percobaan untuk skenario 4 :

TABEL XI
HASIL SKENARIO 4

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingda	0.2	0.3	1	2.1939	2944.12	0.994
2	Logsig – Traingdx	0.4	0.1	1	5.5595	5664.08	0.976
3	Tansig – Traingda	0.2	0.3	1	4.2304	4340.19	0.983
4	Purelin – Traingdx	0.2	0.5	2	1.4811	1583.43	0.997
5	Logsig – Trainlm	0.4	0.3	2	1.4283	1519.30	0.997
6	Tansig – Traingda	0.1	0.2	2	4.6023	4702.45	0.981
7	Purelin – Traingdx	0.1	0.5	2	1.5311	1642.17	0.997
8	Logsig – Trainlm	0.2	0.6	3	1.3975	1492.56	0.998
9	Tansig – Traingdx	0.2	0.3	3	5.9243	6023.08	0.975
10	Purelin – Traingda	0.1	0.1	4	1.2469	1347.09	0.998
11	Logsig – Traingdx	0.1	0.4	4	6.4874	6576.39	0.972
12	Tansig – Trainlm	0.3	0.4	4	1.4080	1501.21	0.997

- Skenario 5

Skenario ini menggunakan variabel input yaitu *open* dan *low price* yang artinya menggunakan dua node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 4 node hidden layer. Hasil model dari skenario 5 akan ditampilkan pada tabel berikut ini :

TABEL XII
HASIL SKENARIO 5

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingda	0.3	0.4	1	1.8349	1934.95	0.995
2	Logsig – Traingdx	0.4	0.2	1	4.6692	4769.32	0.981
3	Tansig – Trainlm	0.4	0.4	1	4.5321	4612.09	0.982
4	Purelin – Traingdx	0.2	0.5	2	1.4674	1574.20	0.997
5	Logsig – Traingda	0.5	0.2	2	5.2558	5350.87	0.978
6	Tansig – Trainlm	0.2	0.5	2	2.2276	2320.98	0.993
7	Purelin – Traingda	0.4	0.3	3	2.0993	2198.99	0.994
8	Logsig – Trainlm	0.2	0.6	3	1.4757	1576.09	0.997
9	Tansig – Traingdx	0.4	0.4	3	2.6635	2768.64	0.991
10	Purelin – Traingdx	0.2	0.5	4	1.4236	1532.09	0.997

11	Logsig – Trainlm	0.3	0.2	4	2.5246	2634.01	0.992
12	Tansig – Traingdx	0.2	0.3	4	5.6793	6781.01	0.976

Hasil skenario 5 yang menggunakan *open* dan *low price* sebagai variabel input menghasilkan no 10 sebagai kombinasi parameter dengan nilai MAPE dan MSE paling

kecil diantara kombinasi-kombinasi parameter yang digunakan dengan menggunakan transfer function purelin, training function traingdx, momentum sebesar 0.2, learning rate sebesar 0.5 dan penggunaan node pada hidden layer sebanyak 4 node.

- Skenario 6

Skenario ini menggunakan variabel input yaitu *high* dan *low price* yang artinya menggunakan dua node input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 4 node hidden layer. Hasil model dari skenario 6 akan ditampilkan pada tabel berikut ini :

TABEL XIII
HASIL SKENARIO 6

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin – Traingdx	0.3	0.4	1	1.2109	1309.03	0.998
2	Logsig – Traingda	0.3	0.5	1	3.4040	3506.15	0.987
3	Tansig – Trainlm	0.4	0.4	1	1.3154	1426.27	0.998
4	Purelin – Traingdx	0.3	0.2	2	1.0654	1176.21	0.999
5	Logsig – Traingda	0.5	0.5	2	6.8901	6992.19	0.970
6	Tansig – Trainlm	0.4	0.4	2	1.2414	1341.47	0.998
7	Purelin – Trainlm	0.2	0.2	3	1.2439	1350.86	0.998
8	Logsig – Traingda	0.1	0.5	3	6.3651	6456.09	0.973
9	Tansig – Trainlm	0.2	0.6	3	1.3712	1486.43	0.998
10	Purelin – Traingdx	0.3	0.2	4	1.0804	1190.01	0.999
11	Logsig – Traingda	0.4	0.1	4	3.6587	3765.76	0.986
12	Tansig – Trainlm	0.3	0.5	4	1.4090	1502.94	0.997

Skenario ini menghasilkan kombinasi parameter terbaik pada no 10 yang menghasilkan nilai MAPE dan MSE paling kecil diantara kombinasi parameter yang ada dengan menggunakan transfer function purelin, training function traingdx, momentum sebesar 0.3, learning rate sebesar 0.2 dan node pada hidden layer menggunakan 4 node.

- Skenario 7

Skenario ini menggunakan variabel input yaitu *open*, *high* dan *low price* yang artinya menggunakan dua node

input pada input layer, untuk node pada hidden layer yang akan digunakan berkisar antara 1 hingga 6 node hidden layer. Hasil model dari skenario 7 akan ditampilkan pada tabel berikut ini :

TABEL XIV
HASIL SKENARIO 7

No	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin - Traingda	0.2	0.2	1	1.3904	1491.12	0.998
2	Logsig – Trainlm	0.4	0.2	1	2.3560	2457.62	0.993
3	Tansig – Traingdx	0.5	0.5	1	3.1668	3267.64	0.989
4	Purelin - Trainlm	0.6	0.4	2	1.7354	1821.09	0.996
5	Logsig - Traingda	0.1	0.5	2	3.0716	3180.15	0.989
6	Tansig – Traingdx	0.2	0.3	2	3.2224	3321.43	0.988
7	Purelin – Traingda	0.3	0.3	3	2.3289	2412.99	0.993
8	Logsig – Traingdx	0.1	0.5	3	5.7762	5874.12	0.975
9	Tansig – Traingda	0.4	0.5	3	3.2977	3384.97	0.988
10	Purelin – Trainlm	0.2	0.5	4	1.5217	1613.91	0.997
11	Logsig - Traingda	0.3	0.4	4	1.5743	1665.76	0.997
12	Tansig – Traingda	0.6	0.5	4	2.7068	2803.09	0.991
13	Purelin – Traingdx	0.3	0.5	5	1.6356	1723.65	0.996
14	Logsig – Trainlm	0.5	0.2	5	1.9503	2274.97	0.995
15	Tansig – Trainlm	0.4	0.3	5	1.6892	1787.96	0.996
16	Purelin-Traingdx	0.1	0.5	6	1.1210	1241.76	0.999
17	Logsig – Traingda	0.3	0.4	6	3.9603	4062.15	0.985
18	Tansig – Traingdx	0.5	0.6	6	4.8765	4965.17	0.980

Hasil skenario 7 yang menggunakan *open*, *high* dan *low price* sebagai variabel input menghasilkan no 16 sebagai kombinasi parameter dengan nilai MAPE dan MSE paling kecil diantara kombinasi-kombinasi parameter yang digunakan dengan menggunakan transfer function purelin, training function traingdx, momentum sebesar 0.1, learning rate sebesar 0.5 dan penggunaan node pada hidden layer sebanyak 6 node.

d) Perbandingan tiap skenario

Setelah melakukan percobaan dengan ketujuh skenario yang menggunakan variabel input yang bermacam-macam dari satu variabel input saja hingga menggunakan tiga variabel input serta penggunaan tuning hyperparameter untuk menemukan kombinasi parameter-parameter dari training function, transfer function, momentum, learning rate dan node pada hidden layer yang akan menghasilkan hasil paling baik dari melihat hasil

MAPE (Mean Absolute Percentage Error) dan MSE (Mean Squared Error) dan nilai regresi. Berikut ini hasil terbaik dari ketujuh skenario yang sudah dilakukan percobaan.

TABEL XV
HASIL TERBAIK DARI TIAP SKENARIO

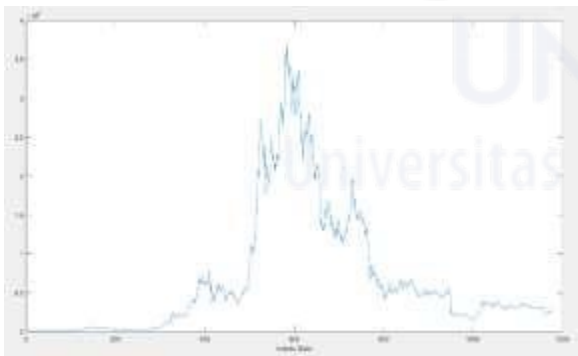
Skenario	Fungsi Aktivasi	Mc	Lr	Node Hidden Layer	MAPE test	MSE	Regresi
1	Purelin - Traingdx	0.3	0.3	2	1.5389	1668.15	0.996
2	Purelin - Traingda	0.1	0.1	2	1.2462	1343.26	0.998
3	Purelin - Traingda	0.1	0.6	2	1.4769	1563.54	0.997
4	Purelin - traingda	0.1	0.1	4	1.2469	1347.09	0.998
5	Purelin - Traingdx	0.3	0.2	4	1.4236	1532.09	0.997
6	Purelin - Traingdx	0.3	0.2	4	1.0804	1190.01	0.999
7	Purelin - Traingdx	0.1	0.5	6	1.1210	124.176	0.999

Dapat dilihat dari tabel diatas, hasil terbaik dari tiap-tiap skenario yang dilakukan. Skenario 6 yang memakai variabel input high, low price memiliki hasil MAPE, MSE dan regresi yang paling baik diantara hasil terbaik dari tiap-tiap skenario yang diujikan.

2. Melakukan pembuatan model arsitektur jaringan syaraf tiruan

a) Melakukan differencing data

Sebelum dilakukan peramalan ARIMA terlebih dahulu dilakukan differencing data untuk melihat pola stasioneritas data.



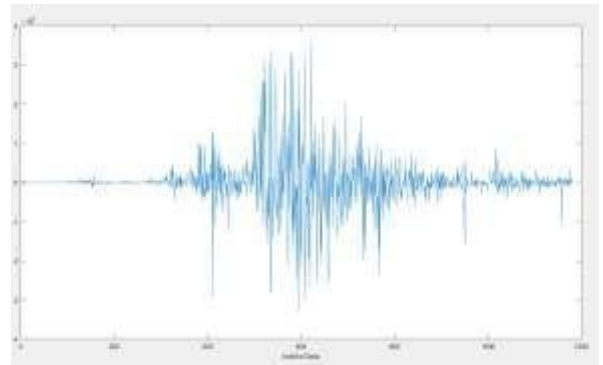
Gbr. 9 Data awal

Dapat dilihat bahwa data mengalami kenaikan yang signifikan dan turun juga signifikan. Data inilah yang tidak bersifat stasioner. Dilakukan differencing data agar data menjadi stasioner dengan menggunakan perintah dibawah ini :

```
Differenced_data= diff(data,1)
```

Gbr.10 Sumber kode differencing data

Untuk hasilnya sebagai berikut :



Gbr. 11 Data hasil differencing

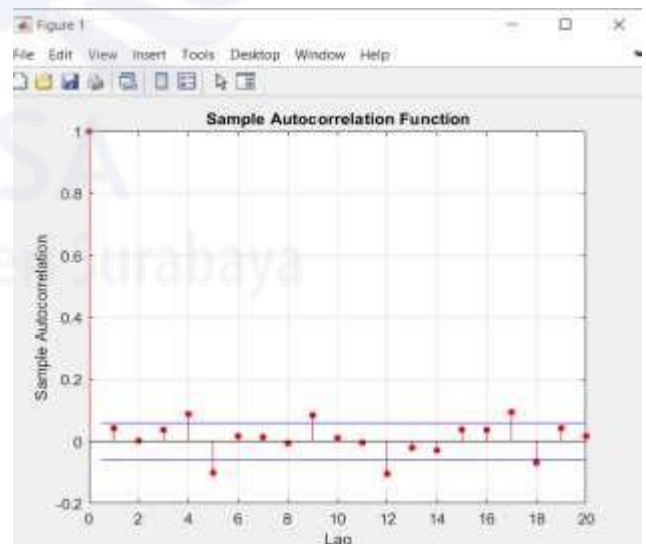
Setelah dilakukan *differencing*, maka dapat dilihat bahwa data mengalami pergerakan naik turun diangka 0 sebagai pusatnya, data hasil *differencing* ini menghasilkan data yang sudah stasioner, jadi untuk nilai D adalah satu (1) karena untuk mencari nilai D dengan melihat berapa kali data dilakukan *differencing*.

b) Pencarian nilai Q

Setelah melakukan differencing data, maka tahapan kedua adalah mencari nilai Q. Nilai Q dapat ditemukan dengan melihat hasil grafik ACF (*Autocorrelation Function*). Untuk mencari hasil grafik ACF menggunakan perintah dibawah ini:

```
%Mencari nilai Q
autocorr(differenced_data);
```

Gbr. 12 Sumber kode mencari nilai Q



Gbr. 13 Hasil ACF

Dari gambar 13 dapat dilihat bahwa grafik ACF turun setelah lag (cutoff) ke 0. Sehingga identifikasi untuk nilai Q adalah 0.

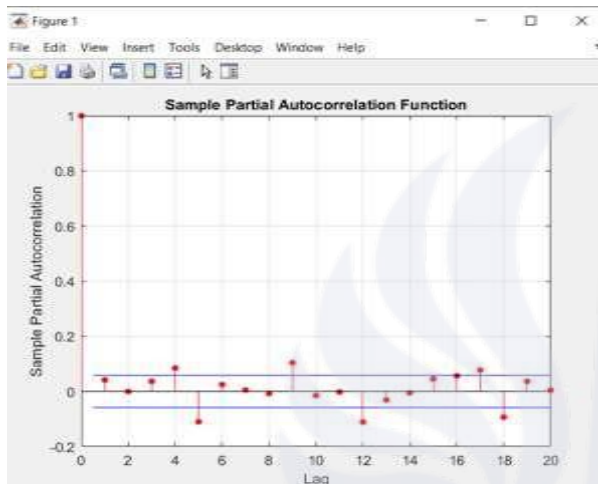
c) Pencarian nilai P

Setelah melalui differencing data dan pencarian nilai Q, selanjutnya adalah mencari nilai P dengan melihat hasil grafik PACF (Sample Partial Autocorrelation Function). Untuk menampilkan grafik PACF menggunakan perintah dibawah ini :

```
%Mencari nilai P
parrrcorr(differenced_data);
```

Gbr. 14 Sumber kode mencari nilai P

Gbr. 15 Hasil PACF



Dari gambar 15 dapat dilihat bahwa grafik PACF turun setelah lag (cutoff) ke 0. Sehingga identifikasi untuk nilai P adalah 0.

d) Proses training dan testing

Setelah menemukan nilai P, D, Q. Selanjutnya adalah melakukan proses training dan testing dengan membagi data training dan testing dengan rasio 80% untuk data training dan 20% untuk data testing untuk mengukur tingkat akurasi menggunakan MAPE dan MSE. Untuk menjalankan proses training dan testing menggunakan perintah dibawah ini :

```
filename = 'Solana.xlsx';
sheet = 'Daftar Harga';
range = 'E5:E1186';
harga = xlsread(filename, sheet, range);

fullData = [harga];
fullDates = 1:numel(fullData);
proporsi_latih = 0.8

jumlah_latih = round(proporsi_latih *
numel(fullData));
jumlah_uji = numel(fullData) - jumlah_latih;

data_latih = fullData(1:jumlah_latih);
data_uji = fullData(jumlah_latih+1:end);

model = arima(0, 1, 0);
fit = estimate(model, data_latih);

prediksi = forecast(fit, 236, 'Y0', data_uji);
```

Gbr. 16 Sumber kode training dan testing MAPE

Selanjutnya dilakukan perhitungan akurasi dengan nilai MAPE dan MSE yang menggunakan perintah dibawah ini :

```
% Hitung akurasi MAPE (Mean Absolute
Percentage Error)
mape = mean(abs((data_uji -
prediksi) ./ data_uji)) * 100;

% Hitung MSE (Mean Squared Error)
mse = mean((data_uji -
prediksi).^2);
```

Gbr. 17 Sumber kode MAPE dan MSE ARIMA

Berikut ini hasil MAPE dan MSE menggunakan pemodelan ARIMA dengan menggunakan nilai P, D, Q senilai 0, 1, 0 :

TABEL XVI
HASIL MAPE DAN MSE ARIMA

Hasil MAPE	Hasil MSE
30.0463	30583

E. Evaluation

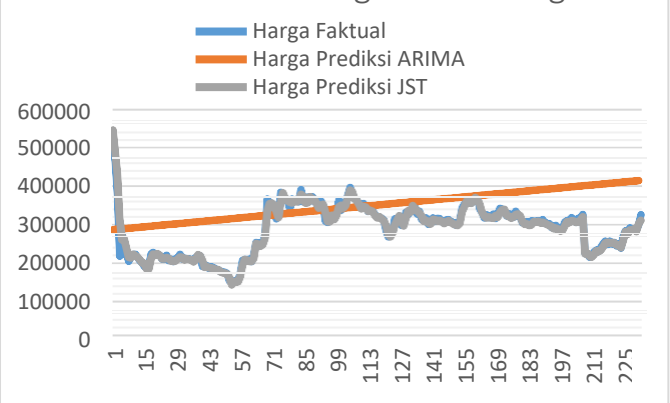
Setelah melakukan pemodelan dengan menemukan model terbaik dari jaringan syaraf tiruan backpropagation dan ARIMA. Selanjutnya adalah membandingkan hasil akurasi MAPE dan MSE beserta grafik hasil perbedaannya dengan data testing. Berikut ini tabel hasil perbandingannya :

TABEL XVII
HASIL PERBADNINGAN JST DENGAN ARIMA

Modelling	Hasil MAPE	Hasil MSE
Jaringan Syaraf Tiruan Backpropagation	1.0804	1190
ARIMA	30.0463	30583

Hasil perbandingan ini akan diperjelas dengan menggunakan gambar grafik. Berikut ini hasil gambar grafik perbedaan modelling yang dilakukan :

Grafik Perbandingan Modelling



Gbr. 18 Grafik perbandingan modelling

F. Deployment

Setelah melihat grafik hasil perbandingan, dapat disimpulkan bahwa modelling menggunakan jaringan syaraf tiruan backpropagation memiliki hasil MAPE dan MSE yang lebih baik daripada modelling ARIMA dan diperjelas oleh gambar 9 dengan hasil modelling JST menunjukkan hasil yang mendekati data testing yang digunakan dalam penelitian ini. Untuk prediksi akan digunakan modelling JST backpropagation. Prediksi akan dilakukan dengan melakukan peramalan untuk harga satu hari kemudian atau pada tanggal 8 Juli 2023.

Untuk harga close price Solana pada tanggal 8 Juli 2023 berdasarkan penggunaan JST backpropagation adalah :

TABEL XVIII
PREDIKSI HARGA CLOSE SOLANA

Harga Prediksi 8 Juli 2023	Rp 310.952,00
-----------------------------------	----------------------

IV. PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian untuk memprediksi harga close price crypto Solana menggunakan jaringan syaraf tiruan backpropagation yang telah dilakukan, didapatkan kesimpulan sebagai berikut :

1. Hasil dengan menggunakan modelling jaringan syaraf tiruan backpropagation menghasilkan model jaringan dengan menggunakan struktur jaringan syaraf tiruan (2, 4, 1) dengan rincian parameter yang digunakan sebagai berikut:
 - Variabel input adalah High dan Low Price,
 - Transfer function menggunakan purelin
 - Training function menggunakan Traingdx
 - Momentum sebesar 0.
 - Learning rate sebesar 0.2
 - Epoch sebesar 70
 - Node Hidden Layer menggunakan 4 node
 - Menghasilkan tingkat akurasi MAPE sebesar 98.9186% dan nilai MSE sebesar 1190
2. Hasil dengan menggunakan modelling ARIMA dengan mencari nilai p, d, q yang optimal menghasilkan nilai p, d dan q sebagai berikut:
 - Nilai P sebesar 0
 - Nilai D sebesar 1
 - Nilai Q sebesar 0
 - Menghasilkan tingkat akurasi MAPE sebesar 69.9537% dan nilai MSE sebesar 30583
3. Dari hasil perbandingan yang dilakukan dapat disimpulkan bahwa model jaringan syaraf tiruan memiliki nilai akurasi MAPE dan MSE yang lebih baik daripada model ARIMA.

B. Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk perbaikan dan pengembangan penelitian untuk topik peramalan harga crypto khususnya menggunakan jaringan syaraf tiruan backpropagation adalah sebagai berikut:

1. Mencoba dengan variabel input lain-nya seperti menggunakan data volume harian sebagai variasi variabel input.
2. Membuat GUI dengan aplikasi MATLAB agar lebih bervariasi.
3. Menggunakan pemodelan lainnya seperti *Fuzzy Time Series* sebagai perbandingan.

REFERENSI

- [1] G.B.Blocks. An Introduction to Solana, (Online), (https://grayscale.com/wpcontent/uploads/2021/12/grayscale_building_blocks-solana-1.pdf), diakses 8 Maret 2023)
- [2] Yakovenko, Anatoly. 2017. Solana: A new architecture for a high performance blockchain v0.8.13, (Online), (<https://solana.com/solana-whitepaper.pdf>), diakses 8 Maret 2023)
- [3] Mauristus, Tuga dan Binsar, Faisal.(2020). Cross Industry Standard Proess for Data Mining.[Online], (<https://mmsi.binus.ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/>), tanggal akses 22 Januari 2023.
- [4] Bayu Aji K, 2018. Peramalan harga Bitcoin menggunakan Backpropagation Neural Network. Skripsi tidak diterbitkan. Surabaya:PPS Institut Teknologi Sepuluh November.