

Implementasi Algoritma *Backpropagation* Dalam Memprediksi Harga Bahan Pangan

Deni Saputra¹, M Safii², M Fauzan³

^{1,3}STIKOM Tunas Bangsa, Pematangsiantar, Medan, Indonesia

²AMIK Tunas Bangsa, Pematangsiantar, Medan, Indonesia

¹Denisaputra1804@gmail.com, ²m.safii@amiktunasbangsa.ac.id,
m.fauzan@stikomtb.ac.id

Abstract

Foodstuffs are raw materials in the form of agricultural, vegetable and animal products used by the food processing industry to produce a food product. Prices of foodstuffs sometimes rise and fall erratically. The purpose of this research is to predict the price of foodstuffs by using the Backpropagation algorithm. The data used in this study is food price data from 2016 to 2019 originating from the Pusat Informasi Harga Pangan Statistik (PIHPS). This research uses the neural network method of the Backpropagation algorithm which uses several architectural models and the results of this test will yield the best accuracy value

Keywords : Prediction, Foodstuffs, Price, Backpropagation

Abstrak

Bahan Pangan adalah bahan baku berupa hasil pertanian, nabati dan hewani yang digunakan oleh industri pengolahan pangan untuk menghasilkan suatu produk pangan. Harga bahan pangan terkadang naik dan turun tidak menentu. Tujuan dilakukannya penelitian ini adalah untuk memprediksi harga bahan pangan dengan menggunakan algoritma Backpropagation. Data yang digunakan pada penelitian ini adalah data harga bahan pangan bulan 2016 sampai dengan 2019 yang berasal dari Pusat Informasi Harga Pangan Strategis (PIHPS) Nasional. Penelitian ini menggunakan metode jaringan saraf tiruan algoritma Backpropagation yang menggunakan beberapa model arsitektur dan hasil dari pengujian ini akan menghasilkan nilai akurasi yang terbaik.

Kata Kunci : Prediksi, Bahan Pangan, Harga, Backpropagation

1. Pendahuluan

Kecerdasan buatan atau disebut juga dengan AI (*Artificial Intelligence*) merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Salah satu yang bisa dipelajari pada kecerdasan buatan adalah metode Jaringan Saraf Tiruan (JST) [1], [2]. Jaringan Saraf Tiruan (JST) merupakan salah satu representasi buatan otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia [3]. Teknik atau Algoritma yang sangat populer pada metode jaringan saraf tiruan adalah algoritma *Backpropagation*, karena algoritma *Backpropagation* memungkinkan untuk menghindari kesulitan yang dijelaskan menggunakan aturan belajar yang mirip dengan plastisitas lonjakan waktu yang tergantung pada sinapsis [4].

Bahan Pangan adalah bahan baku berupa hasil pertanian, nabati dan hewani yang digunakan oleh industri pengolahan pangan untuk menghasilkan suatu produk pangan. Bahan pangan terdiri dari bahan pangan nabati dan bahan pangan hewani, Bahan pangan nabati merupakan bahan makanan yang berasal dari tumbuh-tumbuhan atau yang berbasah dasar dari tumbuhan, sedangkan bahan pangan hewani merupakan produk atau olahan makanan yang berasal dari hewan. Pasar Horas merupakan salah satu tempat untuk

membeli bahan pangan yang mengakomodasi kebutuhan sehari-hari. Setiap bulannya harga bahan pangan mengalami kenaikan ataupun penurunan. Untuk mengatasi masalah diatas perlu sebuah metode yang bisa memprediksi harga bahan pangan tersebut. Metode yang cocok untuk digunakan dalam hal memprediksi/peramalan yaitu metode *Backpropagation*.

Pada penelitian sebelumnya [5] telah dilakukan penelitian untuk memprediksi harga eceran beras di pasar tradisional di 33 kota di Indonesia menggunakan algoritma *Backpropagation*. Hasil yang didapatkan dari penelitian ini adalah prediksi harga eceran beras dengan tingkat akurasi sebesar 88%.

2. Metodologi Penelitian

2.1. Jaringan Saraf Tiruan (JST)

Jaringan Saraf Tiruan (JST) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia. Jaringan Saraf Tiruan (JST) tercipta sebagai suatu generalisasi model matematika dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron [6]. Isyarat mengalir diantara sel saraf melalui suatu sambungan penghubung, setiap sambungan penghubung memiliki bobot yang bersesuaian dan setiap sel saraf merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya [7].

2.2. Komponen Jaringan Saraf Tiruan

Setiap pola-pola informasi *input* dan *output* yang diberikan kedalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron layers. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu Lapisan *input*, Lapisan tersembunyi, Lapisan *output* [8]

2.3. Backpropagation

Backpropagation adalah sebuah metode sistematis untuk pelatihan multilayer jaringan saraf tiruan. *Backpropagation* merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Algoritma ini memiliki dasar matematis yang kuat dan dilatih dengan menggunakan metode belajar terbimbing [7].

Ciri khas *Backpropagation* melibatkan tiga lapisan: lapisan *input*, dimana data diperkenalkan ke jaringan; *hidden layer*, dimana data diproses; dan lapisan *output*, di mana hasil dari masukan yang diberikan oleh lapisan *input* [9]. Langkah-langkah dalam algoritma *backpropagation* oleh Fausett (1994) :

Langkah 0: Inisialisasi bobot (set bobot pada nilai random yang kecil).

Langkah 1: Ketika pada kondisi berhenti salah, lakukan langkah 2 – 9.

Langkah 2: Untuk setiap pasangan *training*, lakukan langkah 3 – 8.

Feedforward

Langkah 3: Setiap unit *input* ($X_i, i = 1, \dots, n$) menerima sinyal *input* X_i dan memancarkan sinyal ini kepada semua unit pada lapisan diatasnya (*hidden unit*).

Langkah 4: Setiap *hidden unit* ($Z_j, j = 1, \dots, p$) menjumlahkan bobot sinyal *input*.

$$z_{in_j} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

mengaplikasikan fungsi aktivasi untuk menghitung sinyal *output*

$$z_j = f(z_{in_j}) \quad (2)$$

dan mengirim sinyal ke semua unit di lapisan di atasnya (*output unit*).

Langkah 5: Setiap unit *output* ($Y_k, k = 1, \dots, m$) menjumlahkan bobot sinyal *input*.

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (3)$$

dan mengaplikasikan fungsi aktivasinya untuk menghitung sinyal *output*.

$$y_k = f(y_{in_k}) \quad (4)$$

Backpropagation

Langkah 6: Setiap unit *output* ($Y_k, k = 1, \dots, m$) menerima pola target sesuai dengan pola *training input*, menghitung informasi *error*.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (5)$$

menghitung koreksi bobotnya (digunakan untuk memperbaharui w_{jk})

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (6)$$

menghitung koreksi bias (digunakan untuk memperbaharui w_{0k})

$$\Delta w_{0k} = \alpha \delta_k \quad (7)$$

mengirim ke unit lapisan dibawahnya.

Langkah 7: Setiap *hidden unit* ($Z_j, j = 1, \dots, p$) menjumlahkan delta *input* (dari unit di lapisan atas).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (8)$$

dikalikan dengan turunan dari fungsi aktivasi untuk menghitung informasi *error*.

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (9)$$

menghitung koreksi bobot (digunakan untuk memperbaharui v_{ij})

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (10)$$

dan menghitung koreksi bias (digunakan untuk memperbaharui v_{0j})

$$\Delta v_{0j} = \alpha \delta_j \quad (11)$$

Update bobot dan bias

Langkah 8: Setiap unit *output* ($Y_k, k = 1, \dots, m$) memperbaharui bias dan bobot ($j = 0, \dots, p$)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (12)$$

Setiap *hidden unit* ($Z_j, j = 1, \dots, p$) memperbaharui bobot dan bias ($i = 0, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (13)$$

Langkah 9: Tes kondisi berhenti.

3. Hasil dan Pembahasan

Pengolahan data dilakukan dengan bantuan *software Matlab*. Sampel data adalah data harga bahan pangan dari tahun 2016 sampai 2019. Data ini akan digunakan pada data pelatihan dan data pengujian. Sampel data yang telah diproses adalah sebagai berikut.

Tabel 1. Data Harga Bahan Pangan Tahun 2016 – 2019

Komoditas	Tahun			
	2016	2017	2018	2019
Beras	9700	10400	10400	11650
Daging Ayam	27300	31100	28550	31650
Daging Sapi	105450	108300	107500	105000
Telur Ayam (kg)	13550	13000	13500	22850
Bawang Merah	29200	23050	26000	34050
Bawang Putih	35600	19050	20100	29450
Cabai Merah	55150	45550	23400	29800
Cabai Rawit	66200	55350	29250	26750
Minyak Goreng	13350	12950	12800	12100
Gula Pasir	13250	12000	12000	13000

3.1. Implementasi Pemrosesan Algoritma Backpropagation

a) Proses Normalisasi

Proses Normalisasi adalah suatu langkah kerja dalam memindahkan angka dari kolom menjadi baris dan dari bilangan bulat menjadi pecahan, hal ini dilakukan agar data yang akan digunakan lebih mudah untuk dilakukan proses perkalian bobot pada *Matlab* karena memiliki angka pecahan atau dinormalisasikan. Untuk mentransformasikan seluruh data real tersebut, digunakan fungsi rumus sebagai berikut :

$$x' = \frac{0.8(x-a)}{b-a} + 0.1 \quad (14)$$

b) Proses Normalisasi Data Harga Bahan Pangan

Sebelum proses pengolahan data dilakukan, perlu dilakukan proses penentuan masukan (*input*) serta target atau hasil yang ingin diinginkan dari proses pengolahan data, berikut data *input* berdasarkan data harga bahan pangan yang telah didapat :

Tabel 2. Data Normalisasi Harga Bahan Pangan

Komoditas	Tahun			
	2016	2017	2018	2019
Beras	0,1000	0,1057	0,1057	0,1158
Daging Ayam	0,2428	0,2736	0,2529	0,2781
Daging Sapi	0,8769	0,9000	0,8935	0,8732
Telur Ayam	0,1312	0,1268	0,1308	0,2067
Bawang Merah	0,2582	0,2083	0,2323	0,2976
Bawang Putih	0,3101	0,1759	0,1844	0,2602
Cabai Merah	0,4688	0,3909	0,2112	0,2631
Cabai Rawit	0,5584	0,4704	0,2586	0,2383
Minyak Goreng	0,1296	0,1264	0,1252	0,1195
Gula Pasir	0,1288	0,1187	0,1187	0,1268

c) Data Pelatihan dan Pengujian

Setelah data *input* dan data target di tentukan, maka langkah selanjutnya adalah menentukan data pelatihan dan pengujian, berikut data pelatihan dan pengujian :

Tabel 3. Data Pelatihan Harga Bahan Pangan

Komoditas	Input		Target
	X1	X2	
Beras	0,1000	0,1057	0,1057
Daging Ayam	0,2428	0,2736	0,2529
Daging Sapi	0,8769	0,9000	0,8935
Telur Ayam	0,1312	0,1268	0,1308
Bawang Merah	0,2582	0,2083	0,2323
Bawang Putih	0,3101	0,1759	0,1844
Cabai Merah	0,4688	0,3909	0,2112
Cabai Rawit	0,5584	0,4704	0,2586
Minyak Goreng	0,1296	0,1264	0,1252
Gula Pasir	0,1288	0,1187	0,1187

Tabel 4. Data Pengujian Harga Bahan Pangan

Komoditas	Input		Target
	X1	X2	
Beras	0,1057	0,1057	0,1158
Daging Ayam	0,2736	0,2529	0,2781
Daging Sapi	0,9000	0,8935	0,8732
Telur Ayam	0,1268	0,1308	0,2067
Bawang Merah	0,2083	0,2323	0,2976
Bawang Putih	0,1759	0,1844	0,2602
Cabai Merah	0,3909	0,2112	0,2631
Cabai Rawit	0,4704	0,2586	0,2383
Minyak Goreng	0,1264	0,1252	0,1195
Gula Pasir	0,1187	0,1187	0,1268

3.2. Perancangan Manual Jaringan Saraf Tiruan

Tahap berikutnya yaitu merancang arsitektur JST backpropagation. Dalam hal ini menggunakan beberapa model jaringan multi-layer (banyak lapisan) yang digunakan untuk mendapatkan arsitektur terbaik adalah 2-2-1, 2-10-1, 2-12-1, 2-3-6-1 dan 2-3-8-1.

Berikut tahapan-tahapan perhitungan manual yang akan dilakukan pada sampel data *input*, yaitu:

a) Tahap *Initialitation*

Inisialisasi (*initialitation*), adalah tahap menginisialisasi data *input*, bobot dari *input* kepada *hidden layer* lalu ke *output layer*. Nilai *input* yang diberikan :

$$X1 = 0,1000$$

$$X2 = 0,1057$$

$$\text{Target} = 0,1057 \quad \text{learning rate } (a) = 0,01$$

Berikan nilai bobot dari *input* ke *hidden layer*. Pada tabel 4.4 dan 4.6 akan terlihat bobot *input layer* ke *hidden layer*, dari *hidden layer* ke *output* yang diambil dari sistem aplikasi *Matlab*.

Tabel 5. Nilai Bobot yang diberikan dari *Input* ke *Hidden Layer*

	Z1	Z2
X1	6,5736	-7,6205
X2	7,1411	7,1152
1	-3,3876	-3,1061

Tabel 6. Nilai Bobot yang diberikan dari *Hidden Layer* Ke *Output Layer*

	L
Z1	0,2647
Z2	-0,8049
	-0,4430

b) Menentukan *input* pada *neuron hidden*

c) Tentukan *output neuron* pada *neuron hidden*

d) Menentukan nilai *output* di *neuron output*, yaitu dengan mengalikan nilai keluaran

e) Mencari nilai *output*

f) Mencari nilai *input* dari *output*

Hitung nilai *error* yang terjadi antara *output actual* dengan *output* yang ingin dicapai, baik pada *hidden layer* maupun *output layer* dalam bentuk masih ternormalisasi.

Nilai *error* pada *ouput* iterasi 1 :

$$Ek(1) = 0,1507 - (-0,4449) = 0,5506$$

Kemudian hitung faktor *error gradient* :

$$\delta(1) = -0,4449(1 - (-0,4449)) * 0,5506 = -0,3540$$

Kemudian hitung koreksi nilai bobot yang akan digunakan untuk memperbaharui nilai bobot W_{jk} pada *neuron output*, dengan menggunakan rumus :

$$\Delta W_{jk} = a * Y_j * \delta_k =$$

Hitung nilai *error* pada *ouput*, yaitu dengan menghitung koreksi nilai bias pada *neuron output* yang akan digunakan untuk memperbaharui nilai. Lalu hitung faktor *error gradient* pada *neuron hidden*. Selanjutnya Hitung koreksi nilai bobot pada *neuron hidden* yang nantinya akan digunakan untuk memperbaharui nilai $V[i,j]$. Nilai perubahan bobot pada masing-masing *neuron input* ke *neuron hidden*. Untuk selanjutnya perhitungan dilanjutkan, sehingga didapatkan data seperti tabel dibawah ini :

Tabel 7. Koreksi Nilai Bobot

Baru	V1	V2
X1	0,000015591	-0,000018015
X2	0,000016479	-0,000019042

Kemudian hitung koreksi nilai bias pada *neuron hidden* yang nantinya digunakan untuk memperbaharui nilai.

g) Langkah selanjutnya memperbaharui nilai bobot dan nilai bias yang terdapat pada *hidden* dan *output layer*.

Tabel 8. Perubahan Bobot Akhir Pada P=... Dari *Input* Ke *Hidden Layer*

Baru	V1	V2
X1	6,5736156	-7,620518
X2	7,14111648	7,115181

Tabel 9. Perubahan Bobot Akhir Pada P=... Dari *Hidden* ke *Output Layer*

Baru	L
Z1	0,2643
Z2	-0,80505
1	-0,4434

Setelah hasil perubahan bobot dan bias pada proses iterasi pertama selesai dilakukan, maka akan menghasilkan nilai perubahan bobot dan bias yang baru. Untuk menghasilkan nilai *error* yang lebih kecil, maka perlu dilakukan terus pelatihan dengan memasukkan nilai bobot dan bias baru, sehingga akan dihasilkan nilai *error* yang jauh lebih kecil dari sebelumnya.

3.3. Hasil Percobaan

a) *Input Data Real*

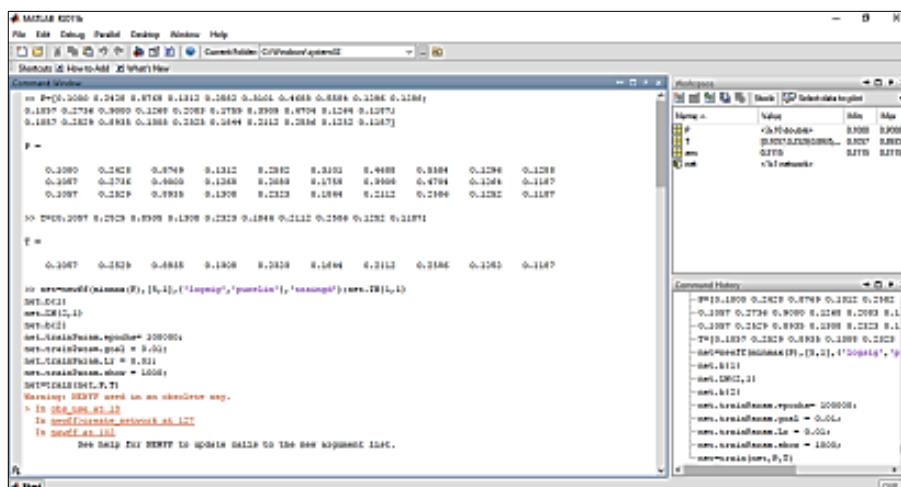
Tahap normalisasi dilakukan setelah data *input* dan *real* dimasukkan dalam *tools* oleh user ke *Microsoft Office Excel*, maka *tools* akan melakukan proses normalisasi sesuai dengan perintah yang diberikan oleh *user*, tampilan hasil normalisasi seperti tabel dibawah ini :

Tabel 10. Normalisasi Data

No	X1	X2	Target
1	0,1000	0,1057	0,1057
2	0,2428	0,2736	0,2529
3	0,8769	0,9000	0,8935
4	0,1312	0,1268	0,1308
5	0,2582	0,2083	0,2323
6	0,3101	0,1759	0,1844
7	0,4688	0,3909	0,2112
8	0,5584	0,4704	0,2586
9	0,1296	0,1264	0,1252
10	0,1288	0,1187	0,1187

b) Koding Data *Training*

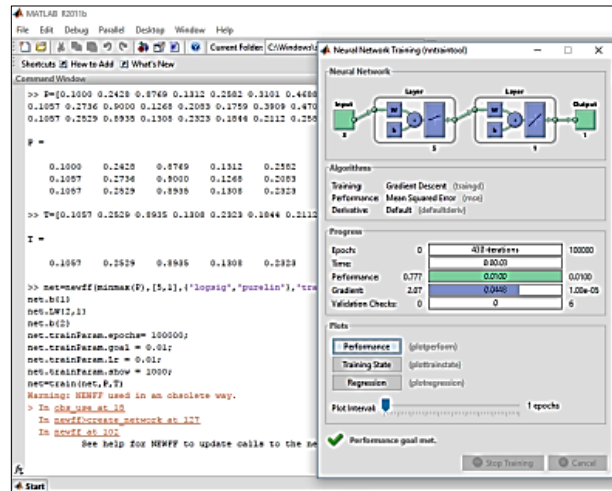
Dari gambar dibawah ini dapat dilihat koding dalam *mentraining* data.



Gambar 1. Koding Data *Training*

c) Proses *Training*

Dari gambar dibawah ini dapat dijelaskan bahwa proses *training* data *backpropagation* pada *matlab*.



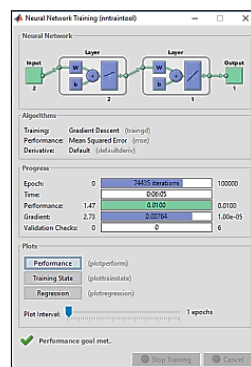
Gambar 2. Proses *Training* Data

Hasil percobaan menunjukkan bahwa jaringan saraf tiruan dengan algoritma *backpropagation* dibentuk dengan membuat generalisasi aturan pelatihan dan pengujian dalam model *Windrow-Hooff* dengan cara menambahkan lapisan tersembunyi (*hidden layer*). Standar algoritma *backpropagation* menggunakan algoritma penurunan gradien (*gradient descent*). Variasi terhadap model standar dilakukan dengan mengganti algoritmanya dengan algoritma lain.

d) Pelatihan dan Pengujian Data dengan Pola 2-2-1

Koding Pelatihan	Koding Pengujian
<pre>>> net=newff(minmax(P),[2,1],{'logsig','purelin'},'traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs= 100000; >> net.trainParam.goal = 0.001; >> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],[],T)</pre>	<pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)</pre>

Setelah semua parameter pelatihan dimasukkan, selanjutnya dilakukan *running* pada *software matlab*, dan hasil pelatihan dapat di lihat pada gambar dibawah ini:

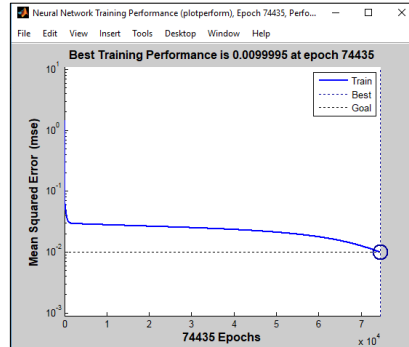


Gambar 3. Hasil Pelatihan pola Arsitektur 2-2-1

Pada gambar diatas didapat hasil pelatihan sebagai berikut :

- 1) Proses pelatihan berhenti pada *iterations* 74435 dari 100000.
- 2) *Error minimum* yang dihasilkan yaitu 0,0099995.
- 3) *Gradient* yang dihasilkan 0,00764.

Sedangkan untuk grafik *performance* pada pola arsitektur 2-2-1 dapat dilihat pada gambar dibawah ini :



Gambar 4. Grafik *Performance* pola 2-2-1

Untuk hasil terinci dan mengetahui *output* dan *error* dapat dilihat pada tabel 4.10 dibawah ini :

Tabel 11. Hasil Pelatihan dengan pola Arsitektur 2-2-1

No	Target	Output JST	Error	SSE	Hasil
1	0,1057	0,1545	-0,0488	0,00238	1
2	0,2529	0,2119	0,0410	0,00168	1
3	0,8935	0,7439	0,1496	0,02238	0
4	0,1308	0,1512	-0,0204	0,00041	1
5	0,2323	0,1424	0,0899	0,00807	0
6	0,1844	0,0505	0,1339	0,01792	0
7	0,2112	0,3133	-0,1021	0,01043	1
8	0,2586	0,4463	-0,1877	0,03522	1
9	0,1252	0,1519	-0,0267	0,00072	1
10	0,1187	0,1462	-0,0275	0,00076	1
				0,09999	70
			MSE	0,01000	

Tabel 12. Hasil Pengujian dengan pola Arsitektur 2-2-1

No	Target	Output JST	Error	SSE	Hasil
1	0,1158	0,1509	-0,0351	0,00123	1
2	0,2781	0,1811	0,0970	0,00941	0
3	0,8732	0,7295	0,1437	0,02066	0
4	0,2067	0,157	0,0497	0,00247	1
5	0,2976	0,1917	0,1059	0,01121	0
6	0,2602	0,1701	0,0901	0,00813	0
7	0,2631	0,0169	0,2462	0,06061	0
8	0,2383	0,0248	0,2135	0,04560	0
9	0,1195	0,1529	-0,0334	0,00112	1
10	0,1268	0,1527	-0,0259	0,00067	1
				0,16109	40
			MSE	0,01611	

Dan begitu seterusnya untuk mencari Pelatihan dan Pengujian Data dengan Pola 2-10-1, Pelatihan dan Pengujian Data dengan Pola 2-12-1, Pelatihan dan Pengujian Data dengan Pola 2-3-6-1, Pelatihan dan Pengujian Data dengan Pola 2-3-8-1. Selanjutnya Pemilihan model arsitektur terbaik dapat terlihat dari beberapa aspek seperti *epoch*, *error minimum* dan akurasi kebenaran. Untuk lebih jelas hasil perbandingan dari setiap model pada arsitektur, dapat dilihat pada tabel dibawah ini :

Tabel 13. Rekapitulasi Model Arsitektur

Arsitektur	Training			Testing	
	Epoch	MSE	Akurasi	MSE	Akurasi
2-2-1	74435	0,009999500	70	0,01610916	40
2-10-1	817	0,010000000	80	0,009812388	70
2-12-1	687	0,009989400	80	0,011276175	60
2-3-6-1	288	0,009985700	100	0,003897474	60
2-3-8-1	23184	0,000999970	90	0,004177626	50

Tabel 14. Tabel data kategorisasi

Keterangan	Error Minimum
1 Benar	0,05-0,001
0 Salah	>0,05

Pada tabel diatas merupakan hasil akurasi dan tingkat MSE dari model arsitektur terbaik, yaitu 2-10-1 yang dihitung dengan menggunakan *software Matlab* dan *Microsoft Office Excel*, menunjukkan dari data yang didapat bahwa *performance* perhitungan jaringan saraf tiruan dengan algoritma *backpropagation* adalah 70%.

Tahap terakhir adalah proses prediksi harga bahan pangan, tahap ini dilakukan dengan menggunakan arsitektur terbaik yaitu 2-10-1. Data yang akan Diprediksi adalah data harga bahan pangan tahun 2020 dan 2021. Tahap selanjutnya adalah melakukan proses denormalisasi, yaitu mengubah bilangan angka yang telah diprediksi kedalam bilangan data tahap awal. Proses ini dilakukan dengan menggunakan rumus

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

Data denormalisasi pada penelitian ini dapat dilihat pada tabel dibawah ini :

1) Prediksi Tahun 2020

Dalam melakukan prediksi harga bahan pangan tahun 2020 data yang akan digunakan merupakan data tahun 2016,2017,2018, dan 2019. Model arsitektur yang akan digunakan adalah 2-10-1. Prediksi harga tahun 2020 dapat dilihat pada tabel dibawah ini :

Tabel 15. Prediksi Harga Bahan Pangan Tahun 2020

No.	Komoditas	Tahun				Prediksi 2020
		2016	2017	2018	2019	
1	Beras	0,1000	0,1057	0,1057	0,1158	0,1968
2	Daging Ayam	0,2428	0,2736	0,2529	0,2781	0,1183
3	Daging Sapi	0,8769	0,9000	0,8935	0,8732	0,8484
4	Telur Ayam (kg)	0,1312	0,1268	0,1308	0,2067	0,1904
5	Bawang Merah	0,2582	0,2083	0,2323	0,2976	0,0933
6	Bawang Putih	0,3101	0,1759	0,1844	0,2602	0,1500
7	Cabai Merah	0,4688	0,3909	0,2112	0,2631	0,2225
8	Cabai Rawit	0,5584	0,4704	0,2586	0,2383	0,2392
9	Minyak Goreng	0,1296	0,1264	0,1252	0,1195	0,1913
10	Gula Pasir	0,1288	0,1187	0,1187	0,1268	0,1936

2) Prediksi Tahun 2021

Dalam melakukan prediksi harga bahan pangan tahun 2021 data yang akan digunakan merupakan data tahun 2017,2018,2019 dan 2020. Model arsitektur yang akan digunakan adalah 2-10-1. Prediksi harga tahun 2021 dapat dilihat pada tabel dibawah ini :

Tabel 16. Prediksi Harga Bahan Pangan Tahun 2021

No.	Komoditas	Tahun				Prediksi 2021
		2017	2018	2019	2020	
1	Beras	0,1057	0,1057	0,1158	0,1968	0,1968
2	Daging Ayam	0,2736	0,2529	0,2781	0,1183	0,1183
3	Daging Sapi	0,9000	0,8935	0,8732	0,8484	0,8484
4	Telur Ayam	0,1268	0,1308	0,2067	0,1904	0,1904
5	Bawang Merah	0,2083	0,2323	0,2976	0,0933	0,0933
6	Bawang Putih	0,1759	0,1844	0,2602	0,1500	0,1500
7	Cabai Merah	0,3909	0,2112	0,2631	0,2225	0,2225
8	Cabai Rawit	0,4704	0,2586	0,2383	0,2392	0,2392
9	Minyak Goreng	0,1264	0,1252	0,1195	0,1913	0,1913
10	Gula Pasir	0,1187	0,1187	0,1268	0,1936	0,1936

4. Kesimpulan

Hasil penelitian yang telah dilakukan dalam memprediksi harga bahan pangan menggunakan algoritma *Backpropagation*. Adapun kesimpulan dari penelitian ini yaitu :

- a) Akuratnya sebuah hasil pelatihan dan pengujian terhadap nilai *output* dalam memprediksi harga bahan pangan, tergantung pada pola arsitektur jaringan yang digunakan.
- b) Arsitektur terbaik untuk penelitian implementasi algoritma *backpropagation* dalam memprediksi harga bahan pangan adalah model 2-10-1 dengan akurasi sebesar 70%.
- c) Implementasi metode *Backpropagation* dengan menggunakan *Matlab* dalam memprediksi harga bahan pangan sudah berhasil dilakukan. Sehingga proses dalam melakukan prediksi menjadi lebih cepat dan akurat dengan menggunakan aplikasi yang mudah digunakan.

Daftar Pustaka

- [1] A. P. Windarto, L. S. Dewi, and D. Hartama, "Implementation of Artificial Intelligence in Predicting the Value of Indonesian Oil and Gas Exports With BP Algorithm," *Int. J. Recent Trends Eng. Res.*, vol. 3, no. 10, pp. 1–12, 2017, doi: 10.23883/IJRTER.2017.3482.J5BBS.
- [2] Sumijan, A. P. Windarto, A. Muhammad, and Budiharjo, "Implementation of Neural Networks in Predicting the Understanding Level of Students Subject," *Int. J. Softw. Eng. Its Appl.*, vol. 10, no. 10, pp. 189–204, 2016.
- [3] A. Wanto and A. P. Windarto, "Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode Backpropagation," *J. Penelit. Tek. Inform.*, vol. 2, no. 2, pp. 37–44, 2017.
- [4] Budiharjo, T. Soemartono, A. P. Windarto, and T. Herawan, "Predicting School Participation in Indonesia using Back-Propagation Algorithm Model," *Int. J. Control Autom.*, vol. 11, no. 11, pp. 57–68, 2018.
- [5] A. A. Fardhani, D. I. N. Simanjuntak, and A. Wanto, "Prediksi Harga Eceran Beras Di Pasar Tradisional Di 33 Kota Di Indonesia Menggunakan Algoritma Backpropagation," *J. Infomedia*, vol. 3, no. 1, 2018, doi: 10.30811/jim.v3i1.625.
- [6] H. Pratiwi *et al.*, "Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks," *J. Phys. Conf. Ser.*, vol. 1471, no. 1, 2020, doi: 10.1088/1742-6596/1471/1/012010.
- [7] A. P. Windarto, M. R. Lubis, and Solikhun, "Implementasi JST Pada Prediksi Total Laba Rugi Komprehensif Bank Umum Konvensional Dengan Backpropagation," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, pp. 411–418, 2018, doi: 10.25126/jtiik.201854767.
- [8] A. R. Youllia Indrawaty, Asep Nana Hermana, "Implementasi Model Backpropagation Dalam Mengenali Pola Gambar Untuk Mendiagnose Penyakit Kulit," *J. Inform.*, vol. 3, no. 1, pp. 1–7, 2012.
- [9] A. P. Windarto, "Implementasi Jst Dalam Menentukan Kelayakan Nasabah Pinjaman Kur Pada Bank Mandiri Mikro Serbelawan Dengan Metode Backpropagation," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 1, no. 1, pp. 12–23, 2017.