Koreksi Pergerakan Robot Linier Pada Meteran Listrik Pintar Menggunakan Algoritma *Backpropagation Neural Network*

Correction of Linear Robot Movement on Smart Electricity Meters Using Backpropagation Neural Network Algorithm

1st Alan Rezky Dwiputra Nur
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
alanrezky@student.telkomuniversity.ac.id

2nd Budhi Irawan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
budhiirawan@telkomuniversity.ac.id

3rd Faisal Candrasyah Hasibuan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
faicanhasfcb@telkomuniversity.ac.id

Abstrak—Listrik Pintar merupakan program Perusahaan Listrik Negara (PLN), pelanggan listrik mengatur listriknya secara mandiri melalui berlangganan listrik menggunakan sistem pulsa. Penggunaan robot linier bisa dimaksimalkan untuk kegiatan pekerjaan. Robot linier yang bergerak secara vertikal dan horizontal dapat melakukan penekanan tombol pada meter listrik prabayar. Robot linier seringkali salah dalam menekan tombol dikarenakan stepper motor robot linier yang tidak berputar dengan baik. Dirancang sebuah sistem agar robot linier dapat memperbaiki kesalahan ketika robot akan menekan tombol. sistem dirancang dengan menggunakan pendekatan deep learning dengan menggunakan pendekatan *Backpropagation Neural Network*. Pendekatan digunakan untuk memprediksi pergerakan *stepper motor* pada robot linier. Robot akan terus bergerak hingga tombol berhasil ditekan. Dalam tugas akhir ini tombol 1 dari pengembangan model *Backpropagation Neural Network* menjadi fokus utama penelitian. Pengujian model dilakukan menggunakan *hyperparameter*. Pengujian *hyperparameter* bertujuan untuk mendapatkan model terbaik dalam melakukan prediksi. Hasil pengujian dibagi menjadi dua yaitu pengujian *hyperparameter* pada model vertikal dan model horizontal pergerakan robot. Model vertikal mendapat *loss Mean Square Error* (MSE) sebesar 23,4351 dan pada model horizontal sebesar 60,9091. kedua model tersebut diuji pada robot linier dengan menekan tombol 1 sebanyak 20 kali. Keberhasilan robot linier dalam menekan tombol sekitar 45%.

Kata Kunci— *backpropagation neural network,* meteran listrik prabayar, robot linier.

Abstract—Smart Electricity is a program of the State Electricity Company (PLN), electricity customers regulate their electricity independently by subscribing to electricity using a pulse system. The use of linear robots can be maximized for work activities. A linear robot that moves vertically and horizontally can perform keystrokes on prepaid electricity meters. Linear robots are often wrong in pressing the button because the stepper motor of the linear robot does not rotate properly. A system is designed so that the linear robot can correct errors when the robot presses a button. The system is designed using a deep learning approach using the Backpropagation Neural Network approach. The approach is used to predict the movement of the stepper motor on a linear robot. The robot will continue to move until the button is successfully pressed. In this final project, button 1 of the Backpropagation Neural Network model

development becomes the main focus of research. Model testing is done using hyperparameters. Hyperparameter testing aims to get the best model in making predictions. The test results are divided into two, namely hyperparameter testing on the vertical model and the horizontal model of the robot's movement. The vertical model got a Mean Square Error (MSE) loss of 23.4351 and the horizontal model of 60.9091. both models were tested on a linear robot by pressing the 1 button 20 times. The success of linear robots in pressing buttons is about 45%.

Keywords— backpropagation neural network, linear robot, prepaid electricity meter.

I. PENDAHULUAN

Pada Tahun 2008 Perusahaan Listrik Negara (PLN) mengeluncurkan program "Listrik Pintar" yakni layanan listrik prabayar dengan menggunakan sistem pulsa sehingga pelanggan dapat megatur penggunaan listriknya sendiri [1]. Seiring berkembangnya teknologi informasi metode pembayaran listrik prabayar memanfaatkan teknologi finansial dengan membeli token listrik secara daring untuk memudahkan pelanggan. Namun, pelanggan harus tetap memasukkan kode token secara manual pada meteran listrik prabayarnya. Hal ini tentunya menyulitkan pelanggan untuk mengisi pulsa listrik apabila sedang tidak berada di rumah. Selain itu, pelanggan juga tidak dapat memantau sisa pulsa listrik pada meteran. Pelanggan hanya bisa melihat informasi listrik rumahnya pada meteran listrik prabayarnya secara langsung.

Robot linier merupakan robot yang bergerak dengan arah sumbu x dan y menggunakan motor linier yang terhubung dengan motor linier lainnya [2]. Robot Linier banyak diimplementasikan pada alat elektronik seperti printer, printer 3D, Laser cuting dan masih banyak lagi. Robot linier dapat dipasangkan pada meteran listrik untuk menekan tombol, sehingga memasukkan nomor token untuk mengisi pulsa listrik dapat dilakukan menggunakan automasi robot.

Akan tetapi robot linier tidak dapat melakukan koreksi error apabila robot belum menekan tombol meteran listrik. Makadari itu Algoritma Backpropagation Neural Network digunakan untuk memprediksi putaran stepper motor yang diperlukan untuk menggerakkan sumbu vertikal dan horizontal pada robot agar sampai ke angka yang akan ditekan. Kemudian robot linear mengkalibrasi dan mengkoreksi pergerakan robot linier ketika tidak menekan tombol dengan benar menggunakan hasil prediksi putaran *stepper motor*.

II. DASAR TEORI /MATERIAL DAN METODOLOGI/PERANCANGAN

A. Neural Network (NN)

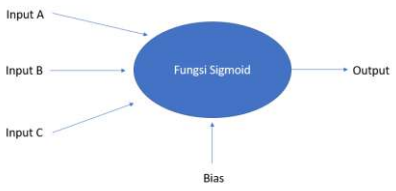
Neural Network adalah teknik mensimulasikan otak manusia yang diterapkan ke dalam proses komputasi. Seorang ahlisyaraf McCulloch dan ahli statistik Pitts memaparkan cara otak mengambil keputusan. Neuron pada otak saling bekerja sama untuk mengirimkan sinyal ke otak. Neuron pada otak manusia berjumlah sekitar 1011 unit yang bekerja secara paralel untuk saling bertukar informasi [1].

Neural Network bekerja dengan cara yang sama menggunakan neuron buatan yang saling terhubung melalui konektor bobot. Neuron tersebut akan menghitung jumlah *input* yang akan datang dan mencari tahu *output*-nya menggunakan teknik *squashing*. fungsi squashing, di mana neuron membatasi rentang amplitudo sinyal output kenilai yang terbatas [1]. *Output* neuron dapat didefenisikan kedalam fungsi linier

$$y_k = v_k \quad (4)$$

Neural Network terdiri dari banyak fungsi seperti menggunakan fungsi sigmoid yang biasa disebut fungsi logistik. *Neural Network* menggunakan metode modifikasi bobot pada proses belajarnya. Fungsi aktivasi diperlukan dalam memodifikasi bobotnya. Berikut persamaan sigmoidnya.

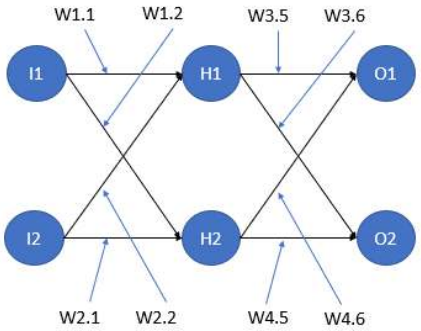
$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (5)$$



GAMBAR 1
INPUT DAN *OUTPUT* PADA NEURON.

Pada Gambar 1.2 menunjukkan sebuah neuron yang menerima *input* dari jumlah neuron kemudian mengirimkan sinyal ke neuron lainnya. Pada *Neural Network* pada lapisan pertama merupakan lapisan *input* yang mengirimkan data melalui sinapsis ke lapisan yang kedua atau biasa disebut *hidden layer*. *Hyden layer* biasa dijuluki sebagai

"kotak hitam" karena memiliki hasil yang tidak dapat ditafsirkan. Kemudian data dikirim ke lapisan ketiga atau disebut lapisan *output* hasil akan ditunkukkan pada lapisan ini [2].



GAMBAR 2
KONEKTIVITAS DAN BOBOT PADA *NEURAL NETWORK.*

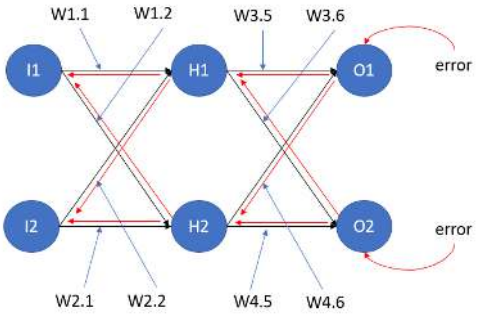
Dalam pembelajaran *Neural Network* terdapat metode untuk meningkatkan tingkat akurasi pada outputnya yakni dengan menambahkan bobot (*weight*). Pembelajaran *Neural Network* dengan menambahkan bobot dapat menyesuaikan nilai *input*. Jumlah variable diperbarui untuk menyesuaikan koneksi antar neuron [2].

B. Backpropagation Neural Network (BNN)

Backpropagation Neural Network adalah metode mengevaluasi kesalahan dalam kumpulan data yang telah diproses pada setiap pada layer ganda neuron [3]. Tujuan dari *backpropagation* ini untuk melatih jaringan syaraf dalam memetakan *input* agar sesuai dengan *output*. *Backpropagation* dapat dilatih untuk mempelajari bobot pada setiap *multi-layer*. Kesalahan minimal pada tiap fungsi nilainya didapat dari penurunan gradien. Jadi metode *backpropagation* dapat digunakan dalam mencari nilai agar sesuai gradiennya untuk meminimalkan kesalahan [4].

Dalam melakukan sebuah training data dengan menggunakan algoritma *backpropagation* dapat dilihat dari data *input*, bobot awal dan data *output*. Pada lapisan syaraf akan menerima umpan balik berupa kelebihan nilai *output*, setelah itu mencari total *input* tiap lapisan syaraf [5]. Total *input* dikompres menggunakan fungsi sigmoid, kemudian ulangi proses tersebut pada lapisan *output.* Kesalahan data *output* dapat dihitung, lalu kesalahan data dimasukkan ke dalam *Backward pass* dengan bobot yang baru. Proses ini diulang terus menerus hingga kesalahan berkurang secara signifikan [4]. Setiap kesalahan output yang dihitung dalam rumus matematika. ∂k Merupakan faktor koreksi, o_k merupakan nilai *output* dan t_k merupakan target *output* [4].

$$\partial_k = o_k(1 - o_k)(t_k - o_k) \quad (6)$$

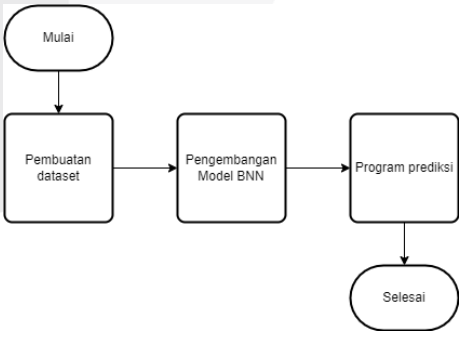


GAMBAR 3
BACKPROPAGATION DARI *OUTPUT* KE *HIDDEN LAYER, HIDDEN LAYER* KE *INPUT LAYER.*

Pada Gambar 4 terdapat 2 *input* (I1, I2), 2 *hiden layer* (H1, H2), dan 2 *output* (O1, O2). Setiap *input*¸ *hyden layer,* dan *output* terhubung melalui bobot W. algoritma Backpropagation Neural Network menghitung *error* dan memperbaharui setiap bobot dari *output* ke *hidden layer* kemudian ke *input layer* untuk mencapai output yang lebih mirip dengan target yang ingin dicapai. Pada setiap percobaan *Backpropagation Neural Network* terus mengoreksi bobotnya hingga sedekat mungkin dengan tujuannya[6][7].

III. PERANCANGAN SISTEM

A. Desain Sistem

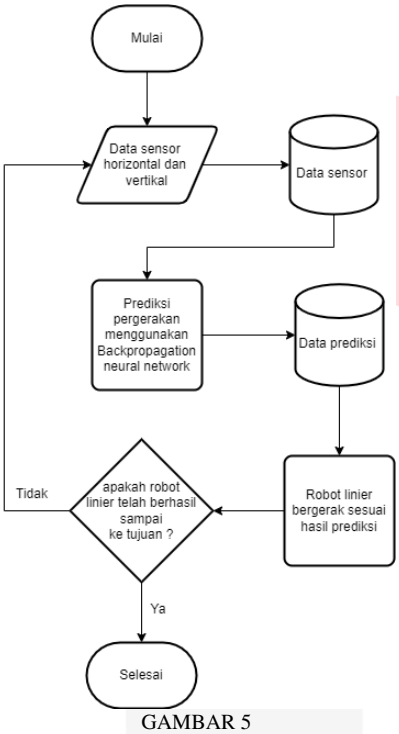


GAMBAR 4
FLOWCHART DESAIN SISTEM.

Pengembangan sistem diawali dengan pembuatan dataset yakni mengumpulkan data sensor dan putaran *stepper motor*. Kemudian membuat model *Backpropagation Neural*

Network untuk memprediksi putaran *stepper motor* yang deperlukan berdasarkan data *training* yang telah dipelajari. Setelah itu membuat program prediksi untuk menyatukan semua sistem dan menjalankan model *Backpropagation Neural Network.*

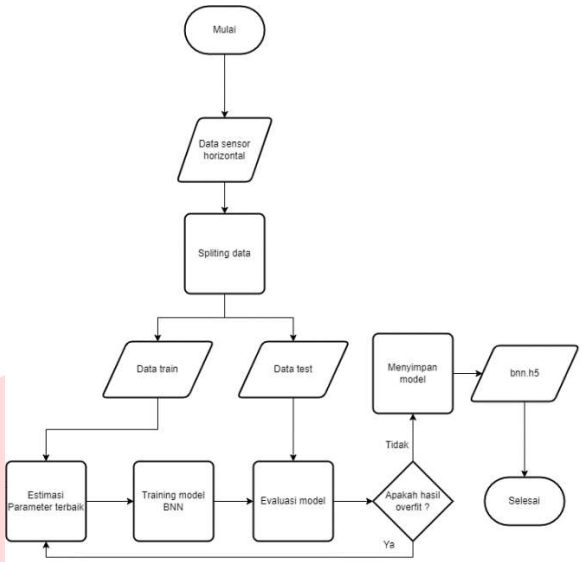
B. Sistem Robot Linier menggunakan Backpropagation Neural Network



GAMBAR 5
FLOWCHART SISTEM ROBOT LINIER MENGGUNAKAN *BACKPROPAGATION NEURAL NETWORK.*

Sistem koreksi pergerakan robot linier menggunakan Algoritma *Backpropagation Neural Network* untuk mengoreksi kesalahan pergerakan robot bekerja melalui proses pengiriman data sensor menggunakan serial komunikasi antara robot linier dan raspberry pi, kemudian data sensor disimpan kedalam database lokal. Data sensor diprediksi secara lokal menggunakan algoritma *Backpropagation Neural Network* dan disimpan ke database lokal. Hasil prediksi berupa jumlah putaran stepper motor, kemudian stepper motor pada robot linier bergerak sesuai dengan hasil prediksi. Apabila robot linier belum sampai ke tujuan maka data sensor terakhir dikirim dan diprediksi, kemudian proses tersebut diulang hingga robot linier sampai ke tujuan.

C. Pengembangan model Backpropagation Neural Network



GAMBAR 1
FLOWCHART PENGEMBANGAN MODE*L BACKPROPAGATION NEURAL NETWORK.*

Pembuatan model *Backpropagation Neural Network* dimulai dengan memproses dataset vertikal sebanyak 549 data dan dataset horizontal sebanyak 464 data. Setiap dataset dilakukan pemisahan label, dilanjutkan dengan *splitting* data, yaitu memecah data menjadi data *train* yang digunakan untuk pembuatan model *Backpropagation Neural Network* dan data *test* digunakan untuk evaluasi model *Backpropagation Neural Network*.

Setelah data train dan data test dibuat, dilanjutkan dengan proses inisialisasi parameter. Setiap parameter yang digunakan dapat mempengaruhi hasil pembuatan model prediksi. Parameter yang digunakan dalam proses *training Backpropagation Neural Network* sebagai berikut.

1. Epoch
2. *Learning rate*

Parameter model *Backpropagation Neural Network* yang digunakan kemudian masuk kedalam proses *training*. Proses *training* bertujuan untuk membangun model yang dapat dipahami algoritma yang terdapat dalam dataset. Setelah itu menguji parameter yang digunakan hingga mendapatkan model terbaik kemudian disimpan dalam file dengan format ".h5".

IV. HASIL DAN PEMBAHASAN

A. Konfigurasi Epoch

Parameter *epoch* yang diuji adalah 50, 100, 150, 200, 250 dan 300. Pengujian dilakukan pada model vertikal dan horizontal,

berikut konfigurasi pengujian *epoch* pada model vertikal dan horizontal.

TABEL 1
KONFIGURASI PENGUJIAN PARAMETER *EPOCH.*

Pengujian	Total pengujian *Epoch*	*Epoch*
Uji ke-1	3 kali	50
Uji ke-2	3 kali	100
Uji ke-3	3 kali	150
Uji ke-4	3 kali	200
Uji ke-5	3 kali	250
Uji ke-6	3 kali	300

Hasil pengujian parameter *epoch* dengan 3 kali pengujian didapat parameter *epoch* terbaik adalah 200 yang menunjukkan tidak terlalu banyak mengalami *overfit*. Dengan nilai minimum *loss* pada uji ke-1 40,7702, uji ke-2 28,5481 dan uji ke-3 24,3051. Hasil terbaik pada model vertikal dengan nilai minimum loss sebesar 24,3051. terlihat pada grafik dengan parameter *epoch* 200 terjadi penurunan *loss* dalam 3 kali pengujian sehingga didapatlah konfigurasi parameter *epoch* terbaik.

Hasil pengujian parameter *epoch* dengan 3 kali pengujian didapat parameter *epoch* terbaik adalah 300 yang menunjukkan tidak terlalu banyak mengalami *overfit*. Dengan nilai minimum *loss* pada uji ke-1 60,8703, uji ke-2 59,9081 dan uji ke-3 59,7771. Hasil terbaik pada model horizontal dengan nilai minimum loss sebesar 59,7771. terlihat pada grafik dengan parameter *epoch* 300 terjadi penurunan *loss* dalam 3 kali pengujian sehingga didapatlah konfigurasi parameter *epoch* terbaik.

B. Konfigurasi learning rate

Parameter learning rate yang diuji adalah 0.004, 0.005, 0.006, 0,007 dan 0,008. Pengujian dilakukan pada model vertikal dan horizontal, berikut konfigurasi pengujian *learning rate* pada model vertikal dan horizontal.

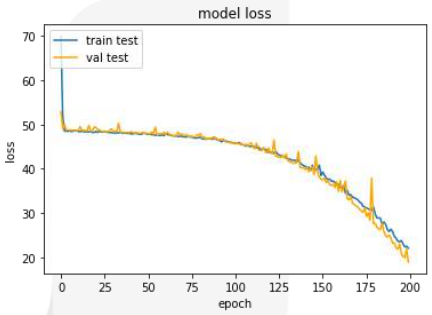
TABEL 2
KONFIGURASI PENGUJIAN PARAMETER *LEARNING RATE.*

Pengujian	Total pengujian *Leratning Rate*	*Learning Rate*
Uji ke-1	5 kali	0.004
Uji ke-2	5 kali	0.005
Uji ke-3	5 kali	0.006
Uji ke-4	5 kali	0.007
Uji ke-5	5 kali	0.008

Hasil pengujian parameter *learning rate* dengan 3 kali pengujian didapat *learning rate* terbaik adalah 0.004 yang menunjukkan tidak terlalu banyak mengalami *overfit*. Dengan nilai minimum *loss* pada uji ke-1 45,7058, uji ke-2 25,7234 dan uji ke-3 23,4351. Hasil terbaik pada model horizontal dengan nilai minimum loss sebesar 23,4351. terlihat pada grafik dengan parameter *learning rate* 0.004 terjadi penurunan *loss* dalam 3 kali pengujian sehingga didapatlah konfigurasi parameter *learning rate* terbaik.

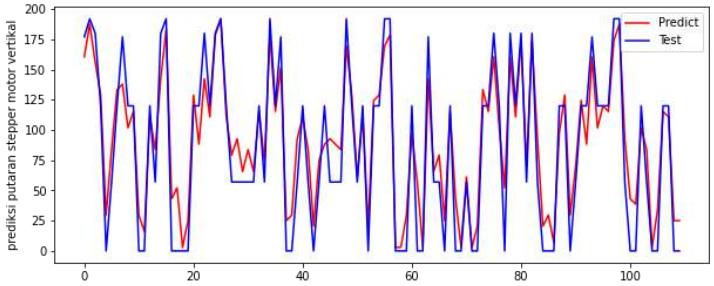
C. Pengujian Model Terbaik Data Vertikal

Model terbaik yang telah didapatkan dalam proses pengujian menggunakan parameter *epoch* 200 dan *learning rate* 0.004 dengan nilai minimum *loss* sebesar 23,4351. dengan menggunakan parameter tersebut model tidak banyak mengalami *overfit.* Berikut grafik model loss yang telah diuji.



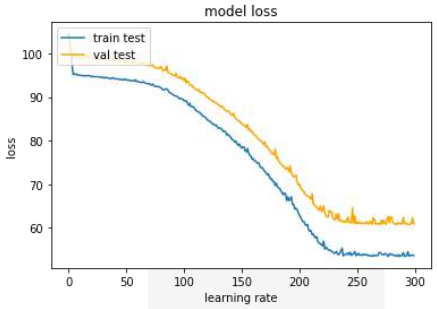
GAMBAR 7
GRAFIK MODEL *LOSS* DATA VERTIKAL.

Hasil evaluasi pengujian model *Backpropagation Neural Network* menunjukkan nilai prediksi yang didapatkan sangat mendekati label. Bisa dilihat pada gambar 4.5 menunjukkan bahwa pola prediksi mendekati pola data *test.* Hasil prediksi membuat pergerakan putaran *stepper motor* vertikal dapat mencapai target lebih baik untuk menekan tombol 1.



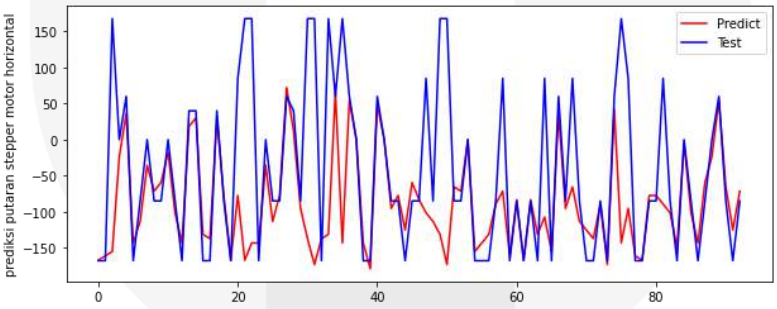
GAMBAR 8
GRAFIK PREDIKSI PUTARAN *STEPPER MOTOR* VERTIKAL.

Model terbaik yang telah didapatkan dalam proses pengujian menggunakan parameter *epoch* 300 dan *learning rate* 0.005 dengan nilai minimum *loss* sebesar 60,9091. dengan menggunakan parameter tersebut model tidak banyak mengalami *overfit.* Berikut grafik model loss yang telah diuji.



GAMBAR 9
GRAFIK MODEL *LOSS* DATA HORIZONTAL.

Hasil evaluasi pengujian model *Backpropagation Neural Network* menunjukkan nilai prediksi cukup mendekati nilai label. Bisa dilihat pada gambar 4.7 menunjukkan bahwa pola prediksi cukup mendekati pola data *test.* Hasil prediksi membuat pergerakan putaran *stepper motor* horizontal dapat mencapai target cukup baik untuk menekan tombol 1.



GAMBAR 10
GRAFIK PREDIKSI PUTARAN *STEPPER MOTOR* HORIZONTAL.

D. Hasil Pengujian Robot linier Menggunakan Model Backpopagation Neural Network

Pengujian koreksi kesalahan pergerakan robot linier dengan algoiritma *Backpropagation Neural Network* dilakukan dengan mengintegrasikan model terbaik yang telah diuji untuk memprediksi putaran stepper motor yang diperlukan untuk mencapai tombol 1. Hasil pengujian dilakukan dengan menekan tombol sebanyak 20 kali. Hasil pengujian koreksi pergerakan robot linier menggunakan *Backpropagation Neural Network* menunjukkan bahwa tingkat keberhasilan menekan tombol 1 sebanyak 9 kali atau 45% sedangkan tingkat kegagalan menekan tombol sebanyak 11 kali atau 55%. Hasil tersebut menunjukkan bahawa tingkat kegagalan dalam menekan tombol lebih besar dibandingkan keberhasilan dalam menekan tombol. Hal tersebut disebabkan karena kurangnya dataset untuk memprediksi

pergerakan stepper motor. Akan tetapi tingkat keberhasilan menekan tombol memiliki perbedaan 10% dari kegagalan menekan tombol yang artinya kesempatan penggunaan algoritma *Backpropagation Neural Network* bisa dilakukan.

V. KESIMPULAN

Model Backpropagation Neural Network terbagi menjadi dua yaitu model vertikal dan model horizontal. Model vertikal dengan parameter *epoch* 200 dan *learning rate* 0.004 dengan loss sebesar 23,4351 kemudian model horizontal dengan parameter *epoch* 300 dan *learning rate* 0.005 dengan loss sebesar 60,9091. Model terbaik *Backpropagation Neural Network* dapat memprediksi pergerakkan stepper motor vertikal dan horizontal yang dibutuhkan untuk menekan tombol 1 dengan percobaan sebanyak 20 kali, tombol berhasil ditekan sebanyak 9 kali atau sekitar 45%.

REFERENSI

[1] A. Zayegh and N. Al Bassam, "Provisional chapter Neural Network Principles and Applications," 2018, [Online]. Available: www.intechopen.com.

[2] S.-H. Han, K. W. Kim, S. Kim, and Y. C. Youn, "Artificial Neural Network: Understanding the Basic Concepts without Mathematics," *Dement. Neurocognitive Disord.*, vol. 17, no. 3, p. 83, 2018, doi: 10.12779/dnd.2018.17.3.83.

[3] Z. Musakulova, E. Mirkin, and E. Savchenko, "Synthesis of the backpropagation error algorithm for a multilayer neural network with nonlinear synaptic inputs," *Proc. 2018 IEEE Int. Conf. Electr. Eng. Photonics, EExPolytech 2018*, pp. 131–135, 2018, doi: 10.1109/EExPolytech.2018.8564433.

[4] J. Amrutha and A. S. Remya Ajai, "Performance analysis of backpropagation algorithm of artificial neural networks in verilog," *2018 3rd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2018 - Proc.*, pp. 1547–1550, 2018, doi: 10.1109/RTEICT42901.2018.9012614.

[5] K. Priandana, I. Abiyoga, Wulandari, S. Wahjuni, M. Hardhienata, and A. Buono, "Development of Computational Intelligence-based Control System using Backpropagation Neural Network for Wheeled Robot," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, vol. 17, pp. 101–106, 2019, doi: 10.1109/ICECOS.2018.8605183.

[6] M. Buscema, "Back Propagation Neural Networks," no. February 1998, 2015, doi: 10.3109/10826089809115863.

[7] C. Sekhar and P. S. Meghana, "A Study on Backpropagation in Artificial Neural Networks," *Asia-Pacific J. Neural Networks Its Appl.*, vol. 4, no. 1, pp. 21–28, 2020, doi: 10.21742/ajnnia.2020.4.1.03.