

Penerapan Jaringan Syaraf Tiruan untuk Meningkatkan Akurasi Sensor Arus PZEM-004T

Application of an Artificial Neural Network to Improve the Accuracy of the PZEM-004T Current Sensor

Reydbo Trihandika Putra, Muhammad Aria Rajasa Pohan*

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia

Jl. Dipati Ukur No. 112, Bandung, Indonesia 40132

Email* : muhammad.aria@email.unikom.ac.id

Abstrak – Sensor PZEM-004T digunakan untuk *monitoring* arus listrik yang dikonsumsi oleh perangkat elektronik. Namun, data yang dihasilkan oleh sensor tersebut masih kurang akurat dan memerlukan optimasi. Dalam penelitian ini, digunakan metode Jaringan Saraf Tiruan (JST) untuk optimasi sistem sensor PZEM-004T dengan menggunakan data amperemeter fluke 112 akurasi 1,5% sebagai target pembelajaran. Arsitektur JST yang digunakan adalah 1-10-1 karena memiliki nilai MSE terkecil sebesar 0,0009 dibandingkan 1-5-1 dan 1-15-1. Hasil simulasi Matlab menunjukkan bahwa arsitektur tersebut sangat efektif dengan selisih *error* 0,0027. Kemudian, parameter JST seperti bobot dan bias diterapkan pada sistem dan berhasil meningkatkan akurasi dengan rata-rata selisih *error* 0,0075. Meskipun terdapat selisih antara nilai *error* pada simulasi Matlab dan implementasi pada Arduino, nilai *error* masih dapat diminimalisir. Dengan demikian, penggunaan sistem optimasi sensor PZEM-004T dengan arsitektur 1-10-1 dan parameter JST pada Arduino dapat menjadi solusi yang efektif dalam meningkatkan akurasi pengukuran arus listrik.

Kata kunci : Sensor PZEM-004T, Jaringan Syaraf Tiruan, Optimasi Sensor

Abstract – The PZEM-004T sensor is used for monitoring the electric current consumed by electronic devices. However, the data generated by these sensors is still inaccurate and requires optimization. In this study, the Artificial Neural Network (ANN) method was used to optimize the PZEM-004T sensor system using fluke 112 ammeter data with an accuracy of 1.5% as a learning target. The ANN architecture used is 1-10-1 because it has the smallest MSE value of 0.0009 compared to 1-5-1 and 1-15-1. Matlab simulation results show that the architecture is very effective with an error difference of 0.0027. Then, ANN parameters such as weights and bias were applied to the system and succeeded in increasing accuracy with an average error difference of 0.0075. Even though there is a difference between the error values in the Matlab simulation and the Arduino implementation, the error values can still be minimized. Thus, the use of the PZEM-004T sensor optimization system with 1-10-1 architecture and ANN parameters on Arduino can be an effective solution in increasing the accuracy of electric current measurements..

Keywords : PZEM-004T Sensor, Artificial Neural Network, Sensor Optimization

I. PENDAHULUAN

Energi listrik digunakan konsumen sesuai dengan kebutuhan. Penggunaan energi listrik tidak dibatasi Perusahaan Listrik Negara (PLN) karena konsumen bertanggung jawab terhadap energi listrik yang digunakan sendiri. Penggunaan energi listrik yang besar membuat pembangkit listrik harus memproduksi daya listrik lebih banyak. Penghematan energi perlu dilakukan karena masih kurangnya pasokan listrik secara merata di

Indonesia[1][2]. Arus merupakan salah satu parameter penting yang ada pada listrik. Akan tetapi, jika nilai arus yang dihasilkan tidak sesuai dengan yang dibaca tentu saja hal itu dapat merugikan konsumen itu sendiri. Tagihan yang akan dibayar oleh konsumen pun dapat tidak sesuai, sehingga dapat merugikan konsumen maupun PLN itu sendiri.

Sensor arus ini sangat penting dalam dunia industri, khususnya pada bidang otomasi dan kontrol

proses[3]. Hasil pengukuran sensor arus ini sangat berpengaruh pada keamanan, keandalan, dan efisiensi suatu sistem. Namun, dalam pengukuran sensor arus sering kali terjadi kesalahan atau ketidakakuratan hasil pengukuran. Oleh karena itu, diperlukan suatu optimasi pada sensor arus agar output yang dihasilkan sesuai dengan yang diinginkan.

Optimasi sensor arus merupakan suatu kegiatan yang penting dilakukan untuk memastikan bahwa pengukuran arus yang dilakukan sesuai dengan yang diinginkan. Hal ini dilakukan karena setiap sensor arus memiliki karakteristik yang berbeda-beda dan dapat mengalami pergeseran pada waktu tertentu[4]. Oleh karena itu, optimasi perlu dilakukan secara berkala agar hasil pengukuran arus yang diperoleh akurat dan dapat diandalkan.

Dalam penelitian ini, salah satu metode yang dapat digunakan untuk meningkatkan akurasi pengukuran sensor arus adalah dengan menerapkan Jaringan Syaraf Tiruan. Jaringan Syaraf Tiruan merupakan suatu model matematika yang dapat meniru cara kerja otak manusia dalam memproses informasi[5]. Dengan menerapkan Jaringan Syaraf Tiruan pada sensor arus, diharapkan hasil pengukuran yang dihasilkan menjadi lebih akurat dan kesalahan pengukuran dapat diminimalisir. Oleh karena itu, penelitian ini bertujuan untuk menemukan cara yang efektif untuk melakukan optimasi akurasi sensor arus dan menerapkan Jaringan Syaraf Tiruan pada sensor arus agar akurasi pengukuran dapat ditingkatkan.

Dengan melakukan penelitian mengenai cara efektif melakukan optimasi akurasi sensor arus dan menerapkan jaringan syaraf tiruan dalam pengukuran sensor arus, diharapkan hasil pengukuran arus yang diperoleh lebih akurat dan dapat diandalkan, serta dapat meningkatkan kualitas pengukuran arus dalam berbagai aplikasi. Penelitian sebelumnya telah melakukan berbagai metode kalibrasi sensor, baik secara manual maupun komputasi. Metode kalibrasi ini akan memakan waktu lama karena kalibrasi ini biasanya dilakukan dengan cara manual dan identifikasi data secara berulang. Selain itu, algoritma yang dibuat hanya dapat diterapkan pada berbagai jenis data sensor dan tidak sesuai untuk digunakan pada sistem pengukuran umum [3].

Pada penelitian Jumrianto pada tahun 2019 menerapkan kalibrasi sensor menggunakan Rumus Regresi Linear. Penerapan Rumus Regresi Linear untuk menentukan nilai arus masukan pada sensor arus ZMCT103C mendapatkan akurasi pengukuran 92,26%.[4]. Pada penelitian tugas akhir Iyan Anugrah, pengujian untuk kerja alat mendapatkan

nilai rata-rata persentase kesalahan alat saat mengukur arus listrik sebesar 4,45%. Nilai rata-rata persentase kesalahan saat mengukur tegangan listrik sebesar 0,25% dan rata-rata persentase kesalahan saat mengukur daya listrik sebesar 4,35% [1].

Penelitian ini memiliki peranan penting dalam meningkatkan akurasi pengukuran sensor arus PZEM-004T dan mengoptimalkan hasil pengukuran yang dihasilkan. Penelitian sebelumnya telah melakukan berbagai metode kalibrasi sensor, namun belum banyak yang menggunakan metode khusus seperti Jaringan Syaraf Tiruan (JST) dengan algoritma *Backpropagation*. Oleh karena itu, penelitian ini mengisi kesenjangan dalam penelitian sebelumnya dengan mengusulkan penggunaan JST untuk mengkalibrasi sensor arus PZEM-004T agar mendekati nilai yang dibaca oleh amperemeter dengan tingkat akurasi yang tinggi. Melalui penelitian ini, diharapkan dapat meningkatkan efisiensi penggunaan energi listrik, meningkatkan kualitas pengukuran arus listrik, serta mengurangi kemungkinan kerugian yang mungkin terjadi baik bagi konsumen maupun PLN.

Penelitian ini memiliki kontribusi kebaruan yang signifikan dengan menerapkan JST dengan algoritma *Backpropagation* sebagai metode kalibrasi sensor arus PZEM-004T. Metode ini diharapkan menjadi alternatif yang lebih efisien dan akurat dalam melakukan kalibrasi sensor arus, terutama pada sensor arus PZEM-004T. Dengan menggunakan JST, diharapkan dapat mencapai tingkat akurasi pengukuran yang lebih tinggi dan mengurangi kesalahan pengukuran. Hal ini akan membantu meningkatkan efisiensi penggunaan energi listrik, meningkatkan kualitas pengukuran arus listrik, serta mengurangi kemungkinan kerugian yang mungkin terjadi baik bagi konsumen maupun PLN.

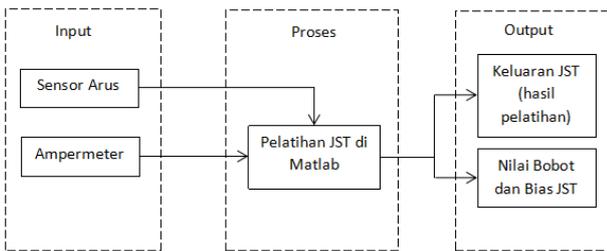
Dalam konteks penelitian sebelumnya, penelitian ini membawa kontribusi baru dengan menerapkan metode JST untuk kalibrasi sensor arus, yang sebelumnya belum banyak digunakan. Dengan demikian, penelitian ini menawarkan pendekatan yang inovatif dan dapat menjadi landasan untuk pengembangan metode kalibrasi sensor arus yang lebih baik di masa depan.

II. METODOLOGI

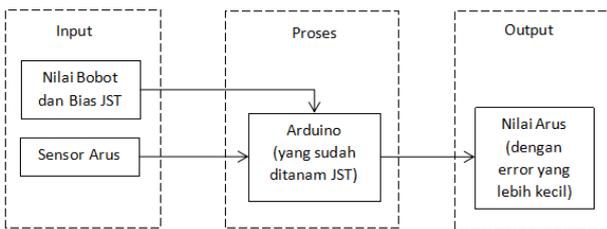
Pada bagian ini akan membahas tentang perancangan sistem beserta implementasi melalui simulasi. Dalam tahapan perancangan ini terdapat perancangan perangkat keras dan perancangan perangkat lunak. Perangkat keras sendiri terdiri dari mikrokontroler, sensor, amperemeter dan beban yang akan diukur untuk pengujian sistem di perangkat

lunak. Sedangkan pada bagian perangkat lunak dalam perancangan sistem ini terdapat program pelatihan sensor dan ampermeter serta penerapannya dengan metode JST.

Pada awal pembahasan, akan difokuskan pada blok diagram dari sistem yang akan dibuat. Terdapat 2 blok diagram yang dibagi menjadi pelatihan di Matlab dan penerapannya ke arduino. Ini merupakan ilustrasi dasar mengenai sistem yang akan dibuat, terdiri dari tiga bagian, yaitu bagian *input*, proses, dan *output*. Untuk blok diagram pelatihan di Matlab dapat dilihat pada **Gambar 1**. Sedangkan blok diagram penerapan ke arduino dapat dilihat pada **Gambar 2**.



Gambar 1. Blok Diagram Sistem Pelatihan di Matlab



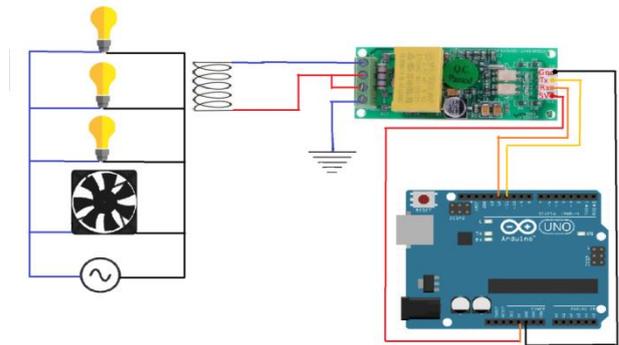
Gambar 2. Blok Diagram Penerapan JST ke Mikrokontroler

Pada blok diagram **Gambar 1**, sensor arus akan berfungsi sebagai *input* dan data yang diperoleh dibaca oleh Arduino IDE. *Input* ampermeter juga akan diamati dan dicatat, lalu diolah menggunakan Matlab. Dalam proses ini, data sensor arus yang sudah didapatkan akan dilatih dengan data sensor arus yang baru saja didapatkan. Setelah dilatih, data yang dibaca oleh arduino akan diolah kembali menggunakan Matlab dan JST.

Setelah pelatihan dilakukan maka hasil pelatihan tersebut diterapkan ke arduino dengan menggunakan parameter pelatihan bobot dan bias yang sudah didapatkan. Melalui proses ini dapat menentukan nilai arus listrik yang akurat dari keluaran JST seperti yang dapat dilihat pada **Gambar 2**.

Untuk mencapai tujuan dari sistem yang akan dibuat, seluruh komponen harus terintegrasi dengan baik dan saling bekerja sama. Dalam hal ini, **Gambar 3**, menunjukkan bagaimana seluruh komponen yang telah disebutkan pada blok diagram dapat digabungkan dan dikabelkan dengan baik pada skematik rangkaian. Skematik rangkaian ini akan

membantu dalam memahami bagaimana seluruh komponen bekerja secara keseluruhan untuk menciptakan sistem yang efisien.



Gambar 3. Skematik Rangkaian

Untuk memasang sensor PZEM-004T ke Arduino, diperlukan beberapa hal yang harus dilakukan. Pertama, *ground* dari sensor PZEM-004T dihubungkan ke *ground* dari Arduino. Selanjutnya, pin VCC dari sensor PZEM-004T dihubungkan ke 5V dari Arduino. Pin RX dari sensor PZEM-004T dihubungkan ke pin digital 2 dari Arduino, sedangkan pin TX dari sensor harus dihubungkan ke pin digital 3 dari Arduino. Setelah rangkaian alat ukur telah selesai dibuat, beban dipasangkan ke alat ukur untuk pengambilan data arus yang ditampilkan pada arduino IDE. Rangkaian alat ukur menggunakan sensor arus PZEM-004T dapat dilihat pada **Gambar 4** dan **Gambar 5**.



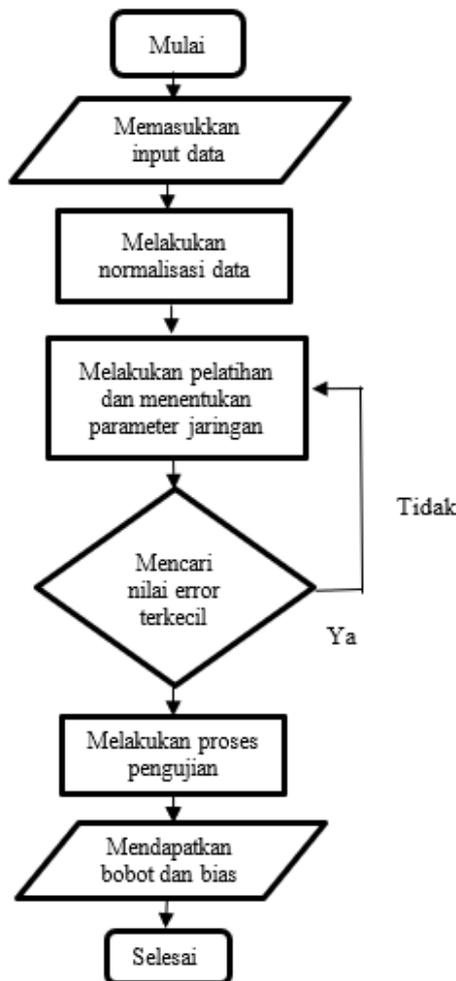
Gambar 4. Rangkaian Alat Ukur Sensor Arus



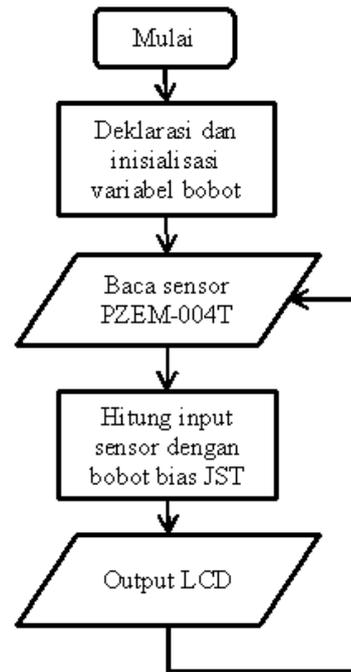
Gambar 5. Beban dari Rangkaian Alat Ukur

Untuk beban yang akan diukur yaitu 3 lampu pijar, 2 kipas angin, TV, kulkas dan setrika yang tersambung ke stop kontak yang sudah dibuat. Karena kebutuhan data untuk melatih di JST cukup banyak dan beban yang akan diukur terbatas, maka pengambilan data dilakukan berulang dan ditambahkan dengan beban yang sudah tersedia.

Dalam penelitian ini, perancangan perangkat lunak dipaparkan dalam bentuk flowchart. Flowchart ini sangat membantu dalam memberikan gambaran bagaimana setiap proses dalam program berlangsung. Ia juga memastikan bahwa program yang dibuat telah sesuai dengan alur yang direncanakan. Flowchart ini membantu mempermudah proses verifikasi dan validasi program agar sesuai dengan yang diharapkan. Persis seperti blok diagram, flowchart pun terbagi menjadi dua yaitu untuk pelatihan di Matlab dan Penerapannya di arduino. Flowchart pelatihan di Matlab dapat dilihat pada **Gambar 6**. Sedangkan flowchart penerapan di arduino dapat dilihat pada **Gambar 7**.



Gambar 6. Flowchart Pelatihan Matlab



Gambar 7. Penerapan ke Arduino

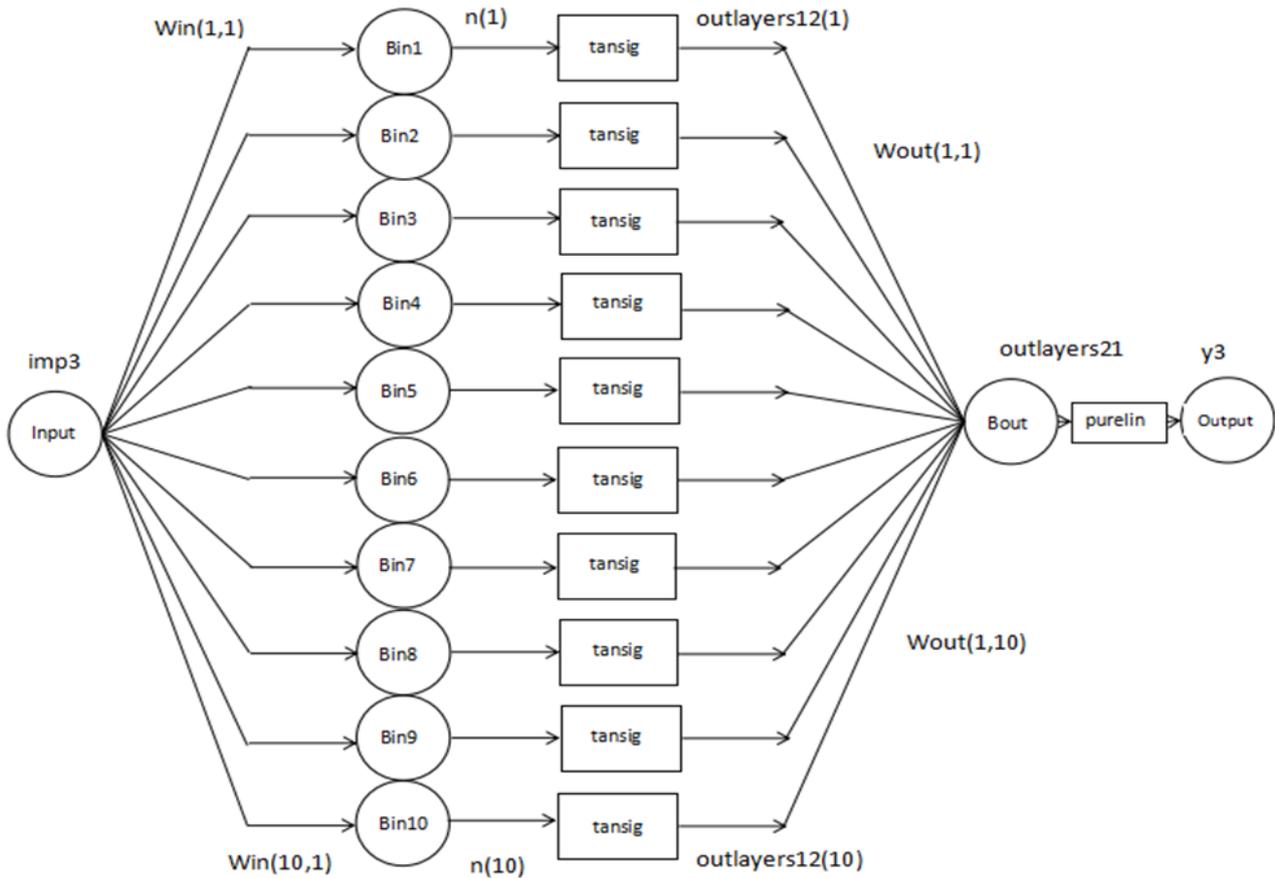
Setelah data-data untuk pelatihan sudah didapatkan, maka langkah awal untuk melakukan pelatihan yaitu membuat arsitektur *backpropagation* nya. Data sensor digunakan sebagai *input*, sedangkan data amperemeter digunakan sebagai target. Data-data tersebut akan dinormalisasi agar penghitungan bobot dan *error* menjadi mudah. Normalisasi data akan dikonversi pada nilai antara -1 dan 1 dengan rumus sebagai berikut [6].

$$Z_i = 2 \cdot \frac{x_i - x_{min}}{x_{max} + x_{min}} - 1 \dots(1)$$

Keterangan :

- Z_i = Hasil normalisasi
- X_i = Data ke- i
- X_{min} = Data minimum
- X_{max} = Data maksimum

Untuk meningkatkan akurasi JST dalam menganalisis data sensor, algoritma *backpropagation* digunakan. *Backpropagation* adalah suatu algoritma yang memodifikasi bobot-bobot yang terhubung dengan setiap neuron pada lapisan tersembunyi. Dalam mengubah bobot ini, *error output* dari hasil analisis akan digunakan sebagai acuan. *Error output* ini diperoleh melalui tahap perambatan maju (*forward propagation*), dimana fungsi aktivasi pada setiap neuron pada lapisan tersembunyi akan diaktifkan. Proses ini dilakukan dengan tujuan untuk mengarahkan bobot-bobot tersebut dalam arah mundur (*backward*) seiring dengan perubahan *error output*[7].



Gambar 8. Model Jaringan Syaraf Tiruan

Pada tahap ini jaringan akan dilatih menggunakan metode pelatihan algoritma *backpropagation*. Algoritma ini terdiri dari 3 layer/lapisan utama : layer *input*, berfungsi untuk masuknya data; layer tersembunyi berfungsi sebagai tempat memproses data; dan layer *output* yang merupakan hasil akhir dari pengolahan data. Pada **Gambar 8**, menjelaskan sistem suatu model perhitungan pada JST. Dimana saat *input* masuk dilakukan normalisasi terlebih dahulu, kemudian dikalikan dan dijumlahkan dengan bobot dan bias *input*. Data yang sudah dijumlahkan bobot dan bias kemudian diaktivasi menggunakan fungsi *tansig sigmoid tangen* lalu dikalikan dan dijumlahkan kembali dengan bobot dan bias *output* sehingga dapat diaktivasi kembali menggunakan fungsi *purelin linear*.

III. HASIL DAN PEMBAHASAN

Pada bab ini, hasil pengujian dan analisis data sensor PZEM-004T akan dipaparkan secara detail. Tujuan dari pengujian ini adalah untuk menilai kemampuan sensor dalam mencapai target yaitu data amperemeter. Dengan melakukan pelatihan pada JST, diharapkan dapat tercapai akurasi yang memadai pada sensor tersebut. Proses pelatihan dilakukan dengan menggunakan algoritma

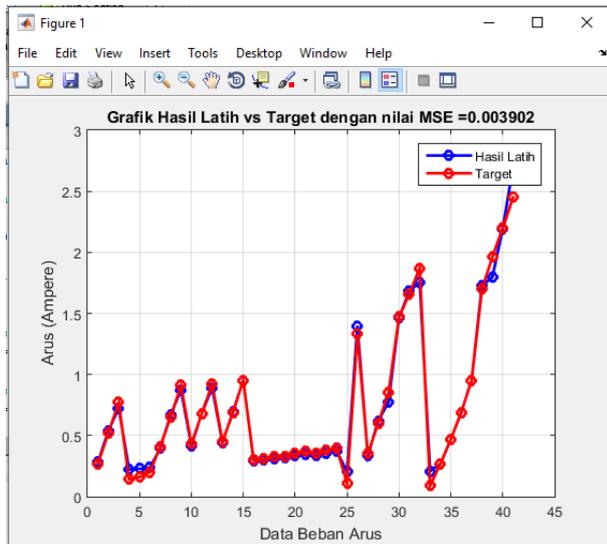
backpropagation, yang memperbaharui bobot-bobot pada lapisan tersembunyi dan memanfaatkan *error output* sebagai dasar perubahan bobot.

Kualitas arsitektur jaringan akan diuji melalui hasil pengujian. Dalam hal ini, parameter-parameter yang mempengaruhi kinerja jaringan akan diamati dan dinilai. Apabila hasil pengujian menunjukkan bahwa jaringan belum mampu mencapai akurasi yang diinginkan, maka dapat dilakukan perbaikan pada arsitektur jaringan, seperti menambah jumlah lapisan atau mengubah jumlah neuron pada lapisan tersembunyi. Dengan melalui proses pengujian dan analisis ini, diharapkan dapat tercapai solusi untuk memperbaiki akurasi data sensor PZEM-004T sehingga dapat digunakan dengan lebih optimal dan efektif dalam berbagai aplikasi.

A. Pengujian Arsitektur Jaringan Syaraf Tiruan

Pelatihan data merupakan validasi model jaringan. Pelatihan data diperlukan beberapa kali *trial and error* untuk mendapatkan jaringan terbaik dengan menentukan jumlah neuron. Pada **Gambar 9**, digunakan arsitektur 1-5-1 dimana 1 diawal merupakan *input*, 5 ditengah jumlah neuron pada

hidden layer, dan 1 diakhir adalah output. Arsitektur 1-5-1 memiliki nilai MSE sebesar 0,003902.



Gambar 9. Grafik Keluaran JST, Target dan MSE neuron 5

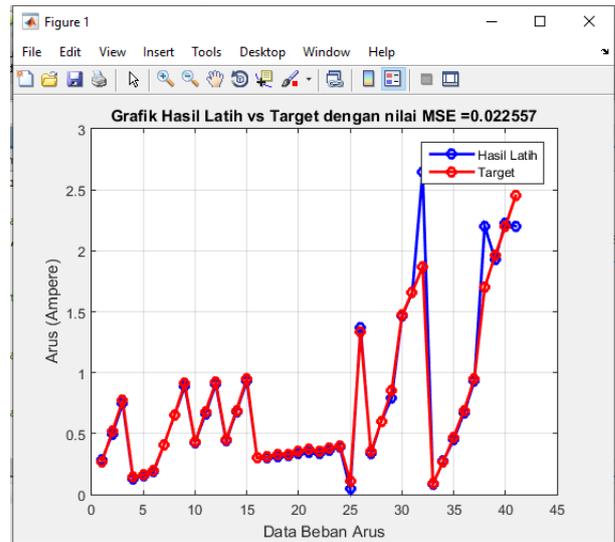
Pada Gambar 10. arsitektur yang digunakan 1-15-1 memiliki nilai MSE yang lebih tinggi dari arsitektur 1-5-1 dengan nilai MSE 0,02257. Sedangkan pada Gambar 11. merupakan arsitektur 1-10-1 yang dimana nilai MSE nya lebih kecil dibandingkan dengan arsitektur yang dibuat sebelumnya dengan nilai MSE sebesar 0,0009529. Dengan perubahan arsitektur yang dibuat dapat dilihat arsitektur dengan jumlah neuron 10 memiliki MSE terkecil. Pada Tabel I. dapat dilihat perbandingannya.

B. Pengujian Hasil Pelatihan Jaringan Syaraf Tiruan

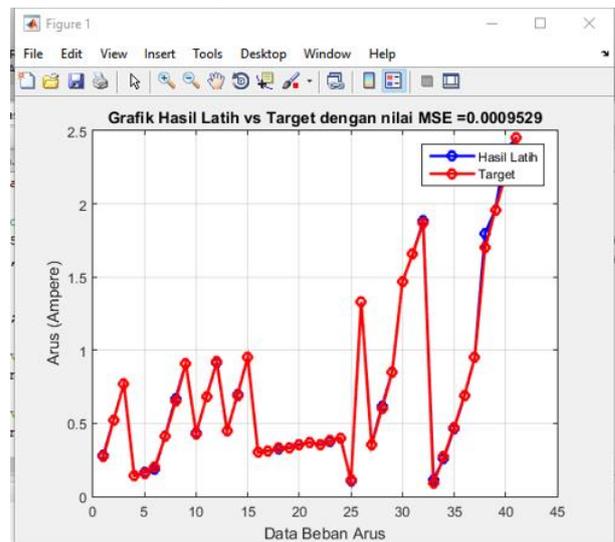
Dalam bab ini, dilakukan pengujian dan analisis data sensor PZEM-004T melalui pelatihan yang bertujuan untuk memperoleh hasil output yang mendekati target, yaitu data amperemeter. Data input diambil dari hasil pengambilan data arus dan amperemeter dan diproses menggunakan JST. Parameter yang digunakan seperti jumlah hidden layer = 1, jumlah output layer = 1, jumlah neuron = 10, serta fungsi aktivasi pada hidden layer menggunakan tansig dan pada output layer menggunakan purelin. Setelah proses pelatihan selesai, diperoleh nilai keluaran dari JST beserta nilai selisih error yang tercantum dalam Tabel II.

Dalam pengujian, perbandingan dibuat antara keluaran dari JST dan target. Hasil analisis menunjukkan bahwa tingkat akurasi meningkat pada beberapa data. Sebagai contoh, pada data ke-2, sensor memiliki nilai 0.89 A dan target bernilai 0.91, dengan selisih error sebesar 0.02. Setelah pelatihan

dengan menggunakan JST, keluaran menjadi 0.998, sehingga selisih error berkurang menjadi 0. Namun masih ada beberapa data yang tidak memperkecil nilai error. Dari nilai error yang diperoleh, rata-rata nilai sebelum dilatih 0.0276. Setelah melakukan pelatihan rata-rata nilai error menjadi 0.0027. Oleh karena itu, dapat disimpulkan bahwa pelatihan dengan menggunakan JST dapat membantu memperkecil perbedaan antara target dan hasil. Namun, masih ada beberapa data yang error nya masih cukup tinggi, sehingga masih dibutuhkan pengujian lebih lanjut untuk mencapai error yang lebih rendah.



Gambar 10. Grafik Keluaran JST, Target dan MSE neuron 15



Gambar 11. Grafik Keluaran JST, Target dan MSE neuron 10

Tabel I. Perbandingan Arsitektur Jaringan Syaraf Tiruan

No.	Input Neuron	Output Neuron	MSE
1.	1	5	0.003902
2.	1	15	0.02257
3.	1	10	0.000952

Tabel II.. Keluaran Jaringan Syaraf Tiruan pada Matlab

Komponen yang Diukur	Sensor Arus (A)	Amperemeter (A)	JST Matlab(A)	Error Sensor Arus	Error JST Matlab
Kipas Angin Lvl 3 + 2 Lampu	0.68	0.69	0.693919066737959	0.01	0.01
Kipas Angin Lvl 1 + 3 Lampu	0.89	0.91	0.998579400670196	0.02	0
2 Kipas Angin Lvl 1 & 1	0.28	0.3	0.301066040404051	0.02	0
2 Kipas Angin Lvl 1 & 2	0.29	0.31	0.312062338216661	0.02	0
2 Kipas Angin Lvl 2 & 3	0.34	0.37	0.367270983548724	0.03	0.01
Kulkas (ON) + 1 Lampu	1.37	1.47	1.47000303772278	0.10	0
Kulkas (ON) + 2 Lampu	1.58	1.66	1.65999968768170	0.08	0
Setrika + 1 Lampu	1.92	1.96	1.96000009160088	0.04	0
Setrika + 2 Lampu	2.17	2.2	2.36526476030588	0.03	0
Setrika + 3 Lampu	2.42	2.45	2.44999995843161	0.03	0
Rata - Rata Error				0.0276	0.0027

C. Pengambilan Parameter Bobot Bias JST

Dalam proses pelatihan JST, parameter bobot dan bias sangat penting untuk diperhatikan dan diambil. Parameter ini akan mempengaruhi hasil dari pelatihan dan tingkat akurasi yang didapatkan. Bobot dan bias ini akan diubah dan ditentukan oleh algoritma pelatihan JST seperti *backpropagation* [8]. Dalam proses ini, akan diterapkan teknik optimasi untuk menentukan nilai bobot dan bias yang optimal. Ini dilakukan dengan membandingkan antara hasil yang diperoleh dari jaringan dengan target yang ditentukan. Setelah beberapa iterasi pelatihan, bobot dan bias akan diperbaiki dan dioptimalkan agar menghasilkan nilai *error* yang lebih kecil dan tingkat akurasi yang lebih tinggi. Oleh karena itu, pengambilan parameter bobot dan bias sangat penting dalam proses pelatihan JST dan harus dilakukan dengan cermat.

Setelah dilakukan pelatihan pada jaringan syaraf tiruan, parameter bobot dan bias yang diperoleh dapat ditanamkan pada *platform* mikrokontroler seperti Arduino. Tujuan dari penanaman parameter ini adalah untuk memudahkan pembacaan data sensor dan memberikan hasil yang lebih akurat seperti data amperemeter. Proses ini bisa dilakukan dengan cara menuliskan kode program yang berisi parameter yang diperoleh pada pelatihan dan memasukkan kode tersebut ke dalam Arduino. Dengan demikian, alat yang menggunakan sensor akan lebih akurat dalam membaca data dan memberikan hasil yang lebih baik dan sesuai dengan data target

Setelah melakukan pelatihan menggunakan arsitektur 1-10-1, Representasi visual dari hasil pembelajaran ini dapat ditemukan pada **Tabel III.** dan **Tabel IV.**

Tabel III. Parameter Pelatihan JST

Bobot Input (Win)	Bias Input (Bin)	Bobot Output (Wout)
13.717	-14.282	-0.273
13.998	-10.889	0.257
-14.030	7.724	-0.066
-14.010	4.630	-0.112
-14.000	1.603	-0.113
13.792	2.403	0.176
-13.483	-5.107	-0.095
13.472	7.785	0.099
-13.642	-10.089	-0.067
13.615	-14.186	-0.004

D. Perbandingan Keluaran JST di Matlab dengan Perhitungan Bobot dan Bias

Untuk membuktikan bahwa bobot dan bias dapat diterapkan ke Arduino, perlu dilakukan perhitungan sehingga nilai output pada simulasi MATLAB sesuai dengan perhitungan dari bobot dan bias tersebut. Adapun persamaan perhitungan yang digunakan adalah sebagai berikut.

$$imp3 = \frac{(y_{maxi}-y_{mini}).(sensor-x_{mini})}{(x_{maxi}-x_{mini})+y_{mini}} \quad \dots(2)$$

Persamaan (2) merupakan persamaan untuk normalisasi data agar rentang -1 s.d 1. Karena y_{maxi} , y_{maxo} , y_{mini} dan y_{mino} memiliki rentang -1 s.d 1.

$$n(i) = (Win(i).imp3) + Bin(i) \quad \dots(3)$$

Persamaan (3) merupakan persamaan perkalian antara bobot *input* dan data normalisasi lalu ditambahkan dengan bias *input*. Bobot *input* adalah matriks 1x10 sehingga nilai n menjadi 1x10.

$$outlayers12(i) = \frac{2}{(1+exp(-2.n(i)))} - 1 \quad \dots(4)$$

Persamaan (4) merupakan persamaan untuk melakukan aktivasi dengan fungsi *sigmoid tangen*. Nilai maksimal *output* dari fungsi ini adalah 1 dan minimal -1. nilai masih matriks 10x1.

$$outlayers21 = (W2(i).outlayers12(i)) + (Bout) \quad \dots(5)$$

Persamaan (5) merupakan persamaan perkalian antara bobot *output* dan nilai yang sudah diaktivasi. Bobot *output* merupakan matriks 1x10 sehingga saat dikalikan dengan nilai aktivasi matriks 10x1, maka menjadi matriks 1x1 kembali. Dimana hasil *outlayers21* merupakan *output* normalisasi.

$$y3 = \frac{(xmaxo-xmino).(outlayers21-ymino)}{(ymaxo-ymino)+xmino} \quad \dots(6)$$

Persamaan (6) merupakan persamaan untuk denormalisasi untuk *output* yang telah didapat agar menjadi nilai asli. Persamaan persamaan diatas kemudian dilakukan perbandingan dengan hasil simulasi yang telah dilakukan. Berikut merupakan perbandingannya pada **Tabel V**.

E. Analisis Hasil Pengujian Keluaran JST di Arduino dengan Penerapan JST

Setelah terbukti nilai bobot dan bias sesuai dengan persamaan diatas, nilai bobot dan bias dapat diinisialisasi pada Arduino IDE dengan fungsi 'float'. Setelah diinisialisasi maka data *input* yang masuk dari sensor PZEM-004T dinormalisasi menggunakan persamaan (2). Data *input* yang sudah dinormalisasi dikalikan dengan bobot *input* matriks 1x10 lalu dijumlahkan dengan bias *input* matriks 1x10 seperti pada persamaan (3). Lalu nilai n dengan matriks 1x10 diaktivasi dengan fungsi *sigmoid tangen* seperti pada persamaan (4). Setelah diaktivasi *outlayers12* dengan matriks 1x10 dikalikan dengan bobot *output* matriks 10x1, sehingga menjadi matriks 1x1 dan dapat dijumlahkan dengan bias *output*. Persamaan yang digunakan yaitu persamaan

(5). *Output* telah didapatkan, namun masih dalam bentuk data normalisasi. Sehingga diperlukan denormalisasi agar menjadi data asli. Persamaan yang digunakan untuk denormalisasi yaitu persamaan (6). Semua persamaan yang dilakukan menggunakan bahasa Arduini IDE. Dengan menerapkan persamaan-persamaan yang telah dibuat dari bobot dan bias ke Arduino IDE, maka sistem optimasi sensor PZEM-004T sudah dapat digunakan dalam pengujian secara *real-time*. Pada **Gambar 12** dan **Gambar 13**. merupakan hasil pengujian penerapan JST ke arduino.



Gambar 12. Hasil pengukuran arus dengan JST



Gambar 13. Hasil pengukuran arus dengan JST

Dalam bab ini, dilakukan pengujian dan analisis data sensor PZEM-004T melalui penerapan JST yang bertujuan untuk memperoleh hasil *output* yang mendekati target, yaitu data amperemeter. Data *input* diambil dari **Tabel II**. dan diproses menggunakan JST. Hasil dari penerapan JST ke arduino menggunakan parameter bobot dan bias memperoleh keluaran nilai arus beserta *error* yang ditunjukkan pada **Tabel VI**.

Tabel IV. Parameter Pelatihan JST

Bias Ouput (Bout)	Xmaxi (max data sensor)	Xmini (min data sensor)	Ymaxi (max data norm sensor)	Ymini (min data norm sensor)	Xmaxo (max data ampermeter)	Xmino (min data ampermeter)	Ymaxo (max data norm ampermeter)	Ymino (min data norm ampermeter)
-0.199	2.42	0.01	1	-1	2.45	0.09	1	-1

Tabel V. Perbandingan Simulasi Matlab & Perhitungan Bobot dan Bias

Data	Simulasi Matlab	Perhitungan Bobot & Bias
Kipas Angin Lvl 1 + 3	$imp3 = (y_{maxi} - y_{mini}) * (0.89 - x_{mini}) / (x_{maxi} - x_{mini}) + y_{mini};$ $n = W11 * imp3 + b1;$ $OutLayers12 = 2 / (1 + \exp(-2 * n)) - 1;$ $OutLayers21 = W21 * OutLayers12 + b2;$ $y3 = (x_{maxo} - x_{mino}) * (OutLayers21 - y_{mino}) / (y_{maxo} - y_{mino}) + x_{mino};$ $y3 = 0.9085$	$imp3 = \frac{(1 - (-1)) * (0.89 - 0.01)}{(2.42 - 0.01) + (-1)} = -0.2697$ $n(i) = (Win(i) * -0.2697) + bin(i)$ $n(1) = -17.9820428636223$ $n(2) = -14.6654271599579$ $n(3) = 11.5086914936327$ $n(4) = 8.40953275204423$ $n(5) = 5.37979352767409$ $n(6) = -1.31647782344386$ $n(7) = -1.47065774346999$ $n(8) = 4.15125675778009$ $n(9) = -6.40963975229224$ $n(10) = -8.76302261939138$ $outlayers12(i) = \frac{2}{(1 + \exp(-2 * n(i)))} - 1$ $outlayers12(1) = -1$ $outlayers12(2) = -1$ $outlayers12(3) = 1$ $outlayers12(4) = 1$ $outlayers12(5) = 1$ $outlayers12(6) = -0.8659$ $outlayers12(7) = -0.8997$ $outlayers12(8) = 0.9995$ $outlayers12(9) = -1$ $outlayers12(10) = -1$ $outlayers21 = (Wout(i) * outlayers12(i)) + (-0.1996)$ $outlayers21 = -0.3063$ $y3 = \frac{(2.45 - 0.09) * (-0.3063 - (-1))}{(1 - (-1)) + 0.09}$ $y3 = 0.9085$
Setrika + 1 Lampu Input 1.92 Target 1.96	$imp3 = (y_{maxi} - y_{mini}) * (1.92 - x_{mini}) / (x_{maxi} - x_{mini}) + y_{mini};$ $n = W11 * imp3 + b1;$ $OutLayers12 = 2 / (1 + \exp(-2 * n)) - 1;$ $OutLayers21 = W21 * OutLayers12 + b2;$ $y3 = (x_{maxo} - x_{mino}) * (OutLayers21 - y_{mino}) / (y_{maxo} - y_{mino}) + x_{mino};$ $y3 = 1.959$	$imp3 = \frac{(1 - (-1)) * (1.92 - 0.01)}{(2.42 - 0.01) + (-1)} = -0.5851$ $n(i) = (Win(i) * -0.5851) + bin(i)$ $n(1) = -6.25648049570023$ $n(2) = -2.69981927138420$ $n(3) = -0.484467674799110$ $n(4) = -3.56657931294743$ $n(5) = -6.58722524813627$ $n(6) = 10.4731943786939$ $n(7) = -12.9959111610843$ $n(8) = 15.6674341746443$ $n(9) = -18.0710633789507$ $n(10) = -21.6486443525232$ $outlayers12(i) = \frac{2}{(1 + \exp(-2 * n(i)))} - 1$ $outlayers12(1) = -1$ $outlayers12(2) = -0.991$ $outlayers12(3) = -0.4498$ $outlayers12(4) = -1$ $outlayers12(5) = -1$ $outlayers12(6) = 1$ $outlayers12(7) = -1$ $outlayers12(8) = 1$ $outlayers12(9) = -1$ $outlayers12(10) = -1$ $outlayers21 = (Wout(i) * outlayers12(i)) + (-0.1996)$ $outlayers21 = 0.5847$ $y3 = \frac{(2.45 - 0.09) * (0.5847 - (-1))}{(1 - (-1)) + 0.09}$ $y3 = 1.959$

Tabel VI. Keluaran Nilai Arus dengan Penerapan JST ke Arduino

Komponen yang Diukur	Sensor Arus (A)	Amperemeter (A)	JST Arduino(A)	Error Sensor Arus	Error JST Arduino
Kipas Angin Lvl 3 + 2 Lampu	0.68	0.69	0.698	0.01	0,008
Kipas Angin Lvl 1 + 3 Lampu	0.89	0.91	0.95	0.02	0
2 Kipas Angin Lvl 1 & 1	0.28	0.3	0.297	0.02	0,003
2 Kipas Angin Lvl 1 & 2	0.29	0.31	0.301	0.02	0,009
2 Kipas Angin Lvl 2 & 3	0.34	0.37	0.322	0.03	0,008
Kulkas (ON) + 1 Lampu	1.37	1.47	1.469	0.10	0,001
Kulkas (ON) + 2 Lampu	1.58	1.66	1.646	0.08	0,014
Setrika + 1 Lampu	1.92	1.96	1.998	0.04	0,038
Setrika + 2 Lampu	2.17	2.2	2.33	0.03	0,13
Setrika + 3 Lampu	2.42	2.45	2.481	0.03	0,031
Rata - Rata Error				0.0276	0.0075

Dalam pengujian ini akurasi meningkat secara signifikan, dibuktikan dengan banyaknya data memiliki *error* 0. Seperti pada beberapa hasil pengujian pada kipas angin *error* awal memiliki nilai 0.02 menjadi 0. Hal ini disebabkan karena sensor PZEM-004T tidak bisa membaca lebih dari 3 angka dibelakang koma, sehingga nilai keluarannya dibulatkan dan mendekati nilai target yaitu amperemeter. Namun saat dilakukan pengujian dengan nilai arus yang lebih tinggi tingkat akurasi menjadi rendah seperti pada kulkas, TV, dan setrika. Pada beberapa data pun ada yang tidak meningkatnya akurasi, seperti pada setrika dimana *error* awal 0.03 menjadi 0.031.

Dengan rata-rata selisih *error* sebelum diterapkan JST 0.0276 menjadi 0.0075 dapat disimpulkan jika metode JST sangat berpengaruh untuk meningkatkan akurasi dengan optimal. Optimasi sensor dengan menggunakan JST dapat digunakan pada beberapa sistem agar output yang diinginkan sesuai dengan keadaan ril nya.

Penelitian ini memberikan kontribusi kebaruan dengan menerapkan metode JST untuk mengoptimasi akurasi sensor arus PZEM-004T. Hasil penelitian menunjukkan bahwa sistem optimasi sensor menggunakan arsitektur 1-10-1 sangat efektif, baik dalam simulasi Matlab maupun penerapannya pada platform Arduino. Rata-rata selisih *error* yang diperoleh sangat rendah, yaitu 0.0027 untuk simulasi Matlab dan 0.0075 untuk Arduino.

Dalam perbandingan dengan penelitian sebelumnya yang menggunakan metode kalibrasi sensor, terdapat beberapa perbedaan. Penelitian sebelumnya menggunakan metode kalibrasi berbasis Rumus Regresi Linear dengan hasil akurasi pengukuran sebesar 92,26%[4]. Sedangkan penelitian ini mengaplikasikan metode JST yang mampu meningkatkan akurasi pengukuran hingga

tingkat yang sangat tinggi, yaitu rata-rata selisih *error* sebesar 0.0027.

Dari segi pentingnya penelitian ini, hasil penelitian menunjukkan bahwa optimasi akurasi sensor arus PZEM-004T melalui penerapan metode JST memiliki kontribusi yang signifikan. Penggunaan sensor arus yang akurat sangat penting dalam berbagai aplikasi industri, terutama dalam bidang otomasi dan kontrol proses[3]. Dengan menggunakan metode JST, hasil pengukuran sensor arus dapat menjadi lebih akurat, memastikan keamanan, keandalan, dan efisiensi suatu sistem.

Meskipun terdapat perbedaan nilai *error* antara hasil simulasi Matlab dan penerapan pada Arduino, nilai *error* yang terjadi masih cukup kecil dan dapat diminimalisir dengan baik. Hal ini menunjukkan bahwa metode JST memiliki potensi untuk digunakan dalam berbagai aplikasi atau sistem pengukuran arus listrik.

Dalam keseluruhan diskusi perbandingan hasil penelitian ini, penelitian ini berhasil membawa kontribusi kebaruan dengan menerapkan metode JST dalam optimasi akurasi sensor arus PZEM-004T. Hasil penelitian ini menunjukkan performa yang sangat baik dalam meningkatkan akurasi pengukuran sensor arus, menghasilkan rata-rata selisih *error* yang rendah, dan memiliki potensi untuk digunakan dalam berbagai aplikasi industri.

IV. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa sistem optimasi sensor PZEM-004T dengan menggunakan data amperemeter sebagai target pembelajaran dan arsitektur 1-10-1 sangat efektif digunakan. *Output* yang dihasilkan dari simulasi Matlab sesuai dengan yang diinginkan dan arsitektur tersebut dapat diterapkan pada Arduino dengan rata-rata selisih *error* sebesar 0.0027. Penerapan parameter JST seperti bobot dan bias juga berhasil meningkatkan akurasi Arduino dengan rata-rata

selisih *error* 0.0075, meskipun terdapat sedikit selisih dengan simulasi Matlab. Meskipun demikian, nilai *error* yang dihasilkan masih cukup kecil sehingga dapat digunakan pada beberapa aplikasi atau sistem dengan melakukan upaya minimalisasi *error* yang cukup baik.

Dalam pengembangan selanjutnya, disarankan untuk membatasi pengambilan data pada kalibrasi sensor menggunakan JST dengan menggunakan potensiometer khusus agar data yang diambil sesuai dengan yang diinginkan dan mampu mengurangi nilai *error* pada beberapa data atau bahkan meningkatkan *error*.

DAFTAR PUSTAKA

- [1] Anugrah, I. (2017). Pengukur Daya Listrik Menggunakan Sensor Arus ACS712-05A Dan Sensor Tegangan ZMPT101B. *Program Studi Teknik Elektro Fakultas Teknik Universitas Negeri Yogyakarta*.
- [2] Muda, E. I. (2017). Perbandingan Data Sensor Arus SCT 013 Dan Sensor Arus ACS 712 Pada Pengukuran Arus Listrik AC (*Doctoral Dissertation, Universitas Brawijaya*).
- [3] Almassri, A. M., Wan Hasan, W. Z., Ahmad, S. A., Shafie, S., Wada, C., & Horio, K. (2018). Self-Calibration Algorithm for a Pressure Sensor with a Real-Time Approach Based on an Artificial Neural Network. *Sensors*, 18(8), 2561.
- [4] Jumrianto, J. (2021). Kalibrasi Sensor Tegangan dan Sensor Arus dengan Menerapkan Rumus Regresi Linear menggunakan Software Bascom AVR. *Journal of System, Information Technology and Electronics Engineering (J-SITEE)*, 1(1), 1-14.
- [5] Wuryandari, M. D., & Afrianto, I. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *Jurnal Komputa Universitas Komputer Indonesia*, 1(1)
- [6] Niu, D., Wang, H., Chen, H., & Liang, Y. (2017). The General Regression Neural Network Based on The Fruit Fly Optimization Algorithm and The Data Inconsistency Rate for Transmission Line Icing Prediction. *Energies*, 10(12), 2066.
- [7] Iryanto, T. (2019). Pemodelan Jaringan Saraf Tiruan untuk Prediksi Konsumsi Listrik Mesin Uji pada Laboratorium Otomotif. *Jurnal Teknologi Bahan dan Barang Teknik*, 9(2), 39-46.
- [8] Febriana, M., Arina, F., & Ekawati, R. (2013). Peramalan Jumlah Permintaan Produksi menggunakan Metode Jaringan Syaraf Tiruan (JST) Backpropagation. *Jurnal Teknik Industri Untirta*, 1(2).
- [13] Harahap, P., Pasaribu, F. I., & Adam, M. (2020). Prototype Measuring Device for Electric Load in Households Using the Pzem-004T Sensor. *Budapest International Research in Exact Sciences (BirEx) Journal*, 2(3), 347-361.
- [14] Sibarani, P. E., Sunarya, U., & Putri, H. (2017). Perancangan Dan Realisasi Voltmeter Dan Amperemeter Dc Menggunakan Mikrokontroler. *eProceedings of Applied Science*, 3(3).