

Implementasi Jaringan Syaraf Tiruan untuk Mengendalikan Lampu Sein Pada Sepeda Motor

Meri Azmi^{1*} Dwi Sudarno Putra², Wawan Purwanto², Toto Sugiarto² dan Donny Fernandez²

¹Jurusan Teknologi Informasi, Politeknik Negeri Padang

²Jurusan Teknik Otomotif, Universitas Negeri Padang

*Corresponding author, e-mail: meriazmi@gmail.com

Abstrak— Pada umumnya lampu sein pada sepeda motor dikendalikan secara manual. Dan tidak seperti pada mobil yang secara mekanis dapat mematikan lampu sein setelah mobil berbelok, lampu sein sepeda motor sering terlihat masih hidup meskipun sepeda motor sudah berbelok. Hal ini biasanya disebabkan oleh kelalaian pengendara. Pada artikel kali ini dijelaskan bagaimana mendesain sebuah logika kendali dengan menggunakan metode Jaringan Syaraf Tiruan yang akan mematikan lampu sein secara otomatis melalui beberapa kondisi tertentu. Metode kontrol jaringan syaraf tiruan yang digunakan adalah Backpropagation yang dikembangkan dan diuji simulasi dengan MATLAB 2017a. Pembelajaran pada jaringan Backpropagation dilakukan untuk mendapatkan bobot jaringan yang sesuai kemudian bobot tersebut digunakan untuk logika kontrol. Dari pengujian terbukti bahwa Jaringan Syaraf Tiruan Backpropagation dengan 2 neuron hidden layer dapat digunakan untuk logika kontrol dalam mengendalikan lampu sein sesuai dengan kondisi yang diinginkan dengan tingkat akurasi 100%.

Kata Kunci : jaringan syaraf tiruan, backpropagation, lampu sein, sepeda motor

Abstract— In general, we control the turn signal on a motorcycle manually. And unlike a car, which can mechanically turn off the turn signal after the car turns, the motorcycle turn signal often seems to stay on even though the motorcycle has finished turning. This might be caused by human error. This article explains how to design a control logic using the Artificial Neural Network method that automatically turns off the turn signal through certain conditions. The method of neural network control was Backpropagation, which is developed and tested simulating with MATLAB 2017. Learning on Backpropagation network is done to get the appropriate network weights then these weights are used for control logic. From the test it was proven that the Artificial Neural Network Backpropagation with 2 neurons on hidden layer can be used for control logic to control the turn signal in accordance with the desired conditions with an accuracy rate of 100%.

Keywords : neural networks, backpropagation, turn signal lamp, motorcycle



This is an open access article distributed under the Creative Commons 4.0 Attribution License

I. PENDAHULUAN

Angka kecelakaan lalu lintas di Indonesia tergolong masih tinggi. Berbagai macam hal dapat memicu terjadinya kecelakaan kendaraan di jalan raya. Untuk itu beberapa upaya dilakukan berbagai pihak untuk mengurangi tingginya angka kecelakaan ini. Berbagai fitur dikembangkan dan ditawarkan oleh produsen kendaraan. Bagi produsen, menambah fitur berarti menambah biaya produksi yang sekaligus juga akan menaikkan

angka harga jual sebuah kendaraan. Cara ini mungkin kurang cocok untuk strategi pemasaran kendaraan. Namun bagi beberapa kalangan konsumen hal ini bisa dimengerti mengingat pentingnya keselamatan berkendara.

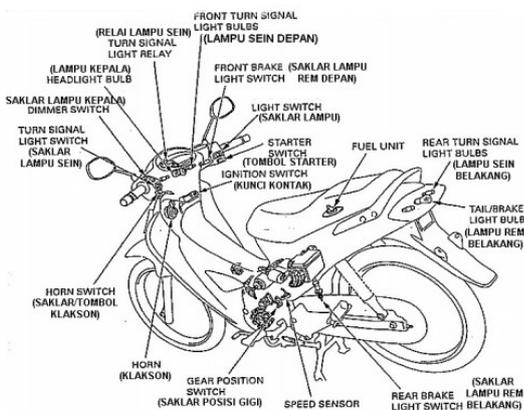
Peneliti juga terus berfikir agar dapat memberikan andil untuk mengurangi angka kecelakaan dan resiko yang dapat ditimbulkannya. Beberapa produk berhasil diimplementasikan langsung pada kendaraan melalui produsen. Beberapa yang lain dikembangkan dan dipasarkan

secara terpisah.

Beberapa penelitian diarahkan untuk memperbaiki sistem pada kendaraan [1]–[6]. Ada juga yang dilakukan pada infrastruktur [7], [8]. Dan beberapa yang lain melakukan penelitian pada alat tambahan [9]–[11]

Saat ini angka kecelakaan didominasi oleh kecelakaan sepeda motor dengan berbagai macam penyebabnya [12]. Salah satu yang menjadi penyebab kecelakaan adalah *human error*.

Sepeda motor, seperti halnya kendaraan lain diwajibkan untuk memiliki sistem pemberi isyarat seperti klakson, lampu sein, lampu rem dan beberapa lampu lain yang juga sebagai sistem keamanan berkendara.



Gambar 1. Contoh posisi saklar, lampu sein dan sistem kelistrikan lainnya yang ada pada satu type sepeda motor [13].

Lampu pemberi isyarat tanda belok atau lebih dikenal dengan Lampu Sein, merupakan isyarat yang penting bagi pengendara sebuah kendaraan jika ingin melakukan aksi berbelok. Sering terjadi kecelakaan ketika pengendara lupa menghidupkan lampu isyarat ini ketika ingin berbelok. Namun tak jarang pula kecelakaan terjadi karena pengendara lupa mematikan lampu sein. Khusus pada kasus kedua ini sering terjadi pada pengendara sepeda motor. Hal ini dikarenakan sistem pengendalian lampu sein pada sepeda motor masih dilakukan secara manual di handle kemudi (Gambar 1) [13]. Tidak seperti pada mobil yang dapat non-aktif secara otomatis melalui mekanisme rotasi kemudi.

Saat ini manusia terbiasa dengan sesuatu yang serba otomatis. Karena diyakini akan semakin memudahkan dan efisien. Untuk mewujudkan mekanisme otomatis diperlukan sebuah sistem pengendalian otomatis. Penelitian untuk mewujudkan sistem pengendalian otomatis ini terus berkembang dengan berbagai metode dan aplikasinya. Salah satu metode yang sering

digunakan adalah metode Jaringan Syaraf Tiruan yang menjadi awal dari berkembangnya teknologi *artificial intelligent*.

Penelitian ini akan menyajikan bagaimana mendesain dan mengimplementasikan sebuah algoritma pengendalian lampu sein untuk sepeda motor dengan beberapa kondisi masukan dan beberapa kondisi keluaran dengan menggunakan metode jaringan syaraf tiruan. Implementasi dan pengujian akan dilakukan menggunakan simulasi MATLAB 2017.

II. METODE PENELITIAN

Seperti yang telah tersirat sebelumnya bahwa penelitian ini bertujuan untuk membuat sebuah sistem otomatis dalam pengendalian lampu sein sehingga nantinya pengendara tidak perlu mematikan secara manual lampu sein pada sepeda motor. Untuk itu dalam melakukan perancangan sistem kendali kita awali dengan penyusunan algoritma pengendalian yang diinginkan kemudian dilanjutkan dengan implementasi algoritma tersebut dalam sebuah sistem JST.

A. Algoritma Pengendalian Lampu Sein

Untuk membuat algoritma ini kita terlebih dahulu menyusun skenario pengendalian. Selanjutnya, dari skenario tersebut akan diidentifikasi input dan output sistem untuk kemudian ditabelkan dalam sebuah tabel algoritma logika kontrol yang nantinya akan dijadikan acuan sebagai training data pada sistem jaringan syaraf tiruan.

Seperti sudah diketahui lampu sein dinyalakan dan dimatikan menggunakan sebuah tombol atau saklar oleh pengendara. Idealnya lampu sein dimatikan setelah kendaraan berbelok. Hal ini bisa kita lakukan dengan menambahkan saklar otomatis seperti *limit switch* [14] atau menambahkan sensor [5]. Hal lain dalam penggunaan lampu sein ini adalah pengendara terkadang juga menggunakan lampu sein untuk memberi tanda jika ingin menepi (tidak sampai berbelok). Untuk kasus ini peneliti akan menambahkan fungsi *timer*.

Dengan berdasarkan pada kondisi di atas maka skenario pengendalian yang direncanakan adalah :

1. Lampu sein akan menyala pada saat pengendara menekan tombol kendali
2. Lampu sein akan mati saat pengendara menekan kembali tombol kendali
3. Lampu sein juga akan mati setelah ada sinyal kendali dari saklar otomatis
4. Lampu sein juga akan mati setelah selang waktu tertentu

Dari 4 skenario tersebut akan ada beberapa komponen yang terlibat yaitu tombol (*trigger switch*), *Automatic switch*, *Timer* dan Lampu Sein. Atau jika dibuat lebih detail input dari sistem ini adalah *trigger switch* kanan dan kiri, *automatic switch* kanan dan kiri, *Timer* (kondisi penuh). Sedangkan outputnya adalah lampu sein kanan dan kiri, *Timer* (kondisi awal) dan trigger kanan dan trigger kiri.

Kemudian disusunlah tabel algoritma pengendalian dengan hasil seperti pada tabel 1.

Tabel 1. Tabel logika pengendalian

| No | Input | | | | | Output | | | | |
|----|-------|-----|-----|------|------|--------|------|-----|-----|-----|
| | TKa | TKi | TmF | ASKa | ASKi | LSKa | LSKi | TmS | TKa | TKi |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 2 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Keterangan

- TKa = Trigger Kanan
- TKi = Trigger Kiri
- TmF = Timmer Full
- ASKa = Automatic Switch Kanan
- ASKi = Automatic Switch Kiri
- LSKa = Lampu Sein Kanan
- LSKi = Lampu Sein Kiri
- TmS = Timmer Start
- 0 = Kondisi tidak aktif
- 1 = Kondisi aktif

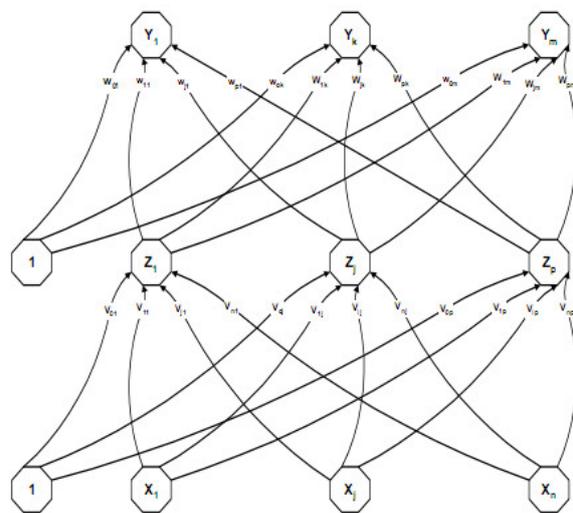
B. Jaringan Syaraf Tiruan Backpropagation

Selanjutnya dari kondisi yang ada pada tabel logika akan dilakukan pembelajaran pada jaringan syaraf tiruan yang dikembangkan. Pada penelitian ini digunakan metode jaringan syaraf tiruan backpropagation. Hal ini dikarenakan metode ini memiliki tingkat pengenalan yang relative lebih baik (akurat), terlebih jika kita memungkinkan untuk melakukan offline training terlebih dahulu [15]–[17]. Arsitektur metode back propagation seperti terlihat pada gambar 2.

Seperti terlihat pada Gambar 2, metode ini memiliki 3 layer yaitu *Input Layer* (X), *Hidden Layer* (Z) dan *Output Layer* (Y). Masing-masing layer memiliki jumlah neuron. Masing-masing Neuron pada sebuah layer memiliki keterkaitan dengan setiap neuron pada layer berikutnya. Secara

singkat, langkah yang digunakan dalam pengembangan simulasi jaringan syaraf tiruan backpropagation pada penelitian ini adalah sebagai berikut:

1. Inisiasi bobot awal
2. Proses feed forward input layer ke hidden layer dengan aktivasi sigmoid biner
3. Proses feed forward hidden layer ke output layer dengan aktivasi sigmoid biner
4. Menghitung factor kesalahan terhadap output dan melakukan koreksi / update terhadap bobot
5. Mengulangi langkah 2,3 dan 4 selama proses training hingga didapatkan factor kesalahan yang sangat kecil, mendekati nol



Gambar 2. Arsitektur Jaringan Syaraf Tiruan [16]

Pada penelitian ini kita memiliki 5 neuron masukan dan 5 neuron keluaran. Sedangkan jumlah neuron pada *hidden layer* akan divariasikan pada beberapa eksperimen yang dilakukan.

Untuk training kita menggunakan data tabel logika pada tabel 1 yang dilakukan perulangan sebanyak 10 kali. Hal ini dilakukan untuk memperbanyak data learning setiap epochnya.

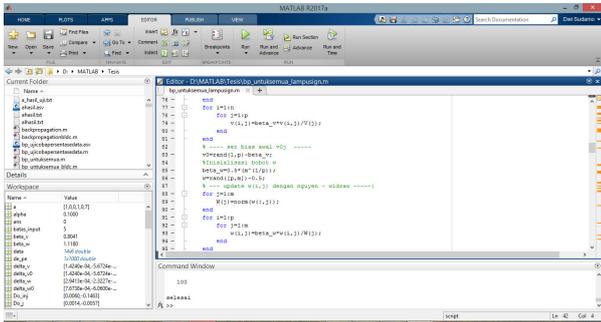
Setelah diperoleh jaringan yang sesuai dengan harapan (memiliki tingkat pengenalan 100%). Maka selanjutnya adalah mengujinya dengan data yang dimasukkan sesuai dengan kondisi yang ada.

C. Implementasi Metode Backpropagation

Pada penelitian ini digunakan MATLAB 2017a. (Gambar 3) Sebenarnya MATLAB memiliki fungsi yang sudah tersedia untuk proses Backpropagation. Tetapi dalam hal ini digunakan *generic coding* yang dibuat sendiri berdasarkan langkah yang telah dijelaskan sebelumnya.

Pada proses training setiap iterasi epoch akan memiliki nilai kesalahan. Diharapkan nilai

kesalahan ini semakin tinggi epochnya makan nilainya akan semakin kecil. Dengan MATLAB hal ini bisa ditunjukkan seperti pada Gambar 4.



Gambar 3. MATLAB 2017 Layout Editor

Pada proses pengujian akan dilakukan dengan variasi jumlah neuron di hidden layer dan jumlah epoch training. Masing-masing variasi akan dilakukan sebanyak 10 kali pengujian. Hal ini untuk mengetahui rerata waktu proses dan rerata tingkat pengenalan yang dilakukan oleh jaringan syaraf tiruan yang terbentuk pada masing-masing variasi. Karena dalam sistem kendali, akurasi dan waktu eksekusi proses adalah hal yang sangat penting. Tujuan akhirnya adalah bagaimana mendapatkan sistem yang akurat dengan waktu yang cepat.

III. HASIL DAN PEMBAHASAN

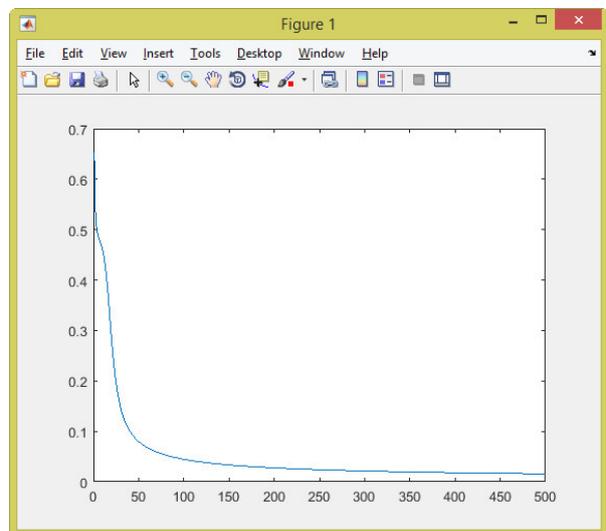
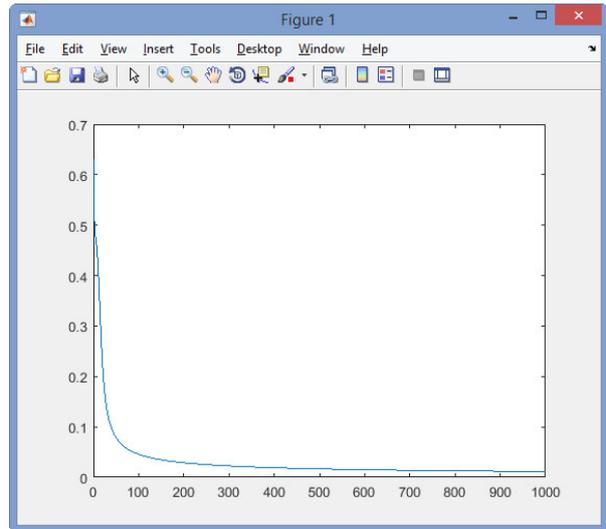
A. Hasil

Seperti sudah disebutkan sebelumnya variasi penelitian dilakukan pada jumlah epoch learning dan jumlah hidden layer. Jumlah Epoch adalah 100, 500 dan 1000. Sedangkan jumlah neuron pada hidden layer adalah 5, 3, 2 dan 1. Pada masing-masing variasi tersebut akan diambil data waktu proses dan persentasi pengenalan pola kendali.

Data hasil pengujian kemudian ditabulasi pada Tabel 2 sampai dengan Tabel 4

Tabel 2. Hasil pengujian dengan Epoch 100

| Epoch = 100 | hidden layer : 5 | | hidden layer : 3 | | hidden layer : 2 | | hidden layer : 1 | |
|------------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|
| | waktu (s) | % Pengenalan |
| Perulangan ke- 1 | 0.083 | 100 | 0.045 | 100 | 0.074 | 100 | 0.037 | 71.43 |
| 2 | 0.102 | 100 | 0.073 | 100 | 0.050 | 85.71 | 0.034 | 42.86 |
| 3 | 0.077 | 100 | 0.065 | 100 | 0.049 | 100 | 0.033 | 71.43 |
| 4 | 0.080 | 100 | 0.060 | 100 | 0.038 | 100 | 0.047 | 28.57 |
| 5 | 0.064 | 100 | 0.061 | 100 | 0.039 | 100 | 0.037 | 57.14 |
| 6 | 0.060 | 100 | 0.047 | 100 | 0.052 | 71.43 | 0.034 | 57.14 |
| 7 | 0.057 | 100 | 0.046 | 100 | 0.040 | 100 | 0.033 | 71.43 |
| 8 | 0.058 | 100 | 0.048 | 100 | 0.038 | 100 | 0.032 | 71.43 |
| 9 | 0.057 | 100 | 0.042 | 100 | 0.039 | 100 | 0.033 | 71.43 |
| 10 | 0.068 | 100 | 0.049 | 100 | 0.038 | 100 | 0.032 | 57.14 |
| Rerata | 0.071 | 100 | 0.054 | 100 | 0.046 | 95.71 | 0.035 | 60.00 |



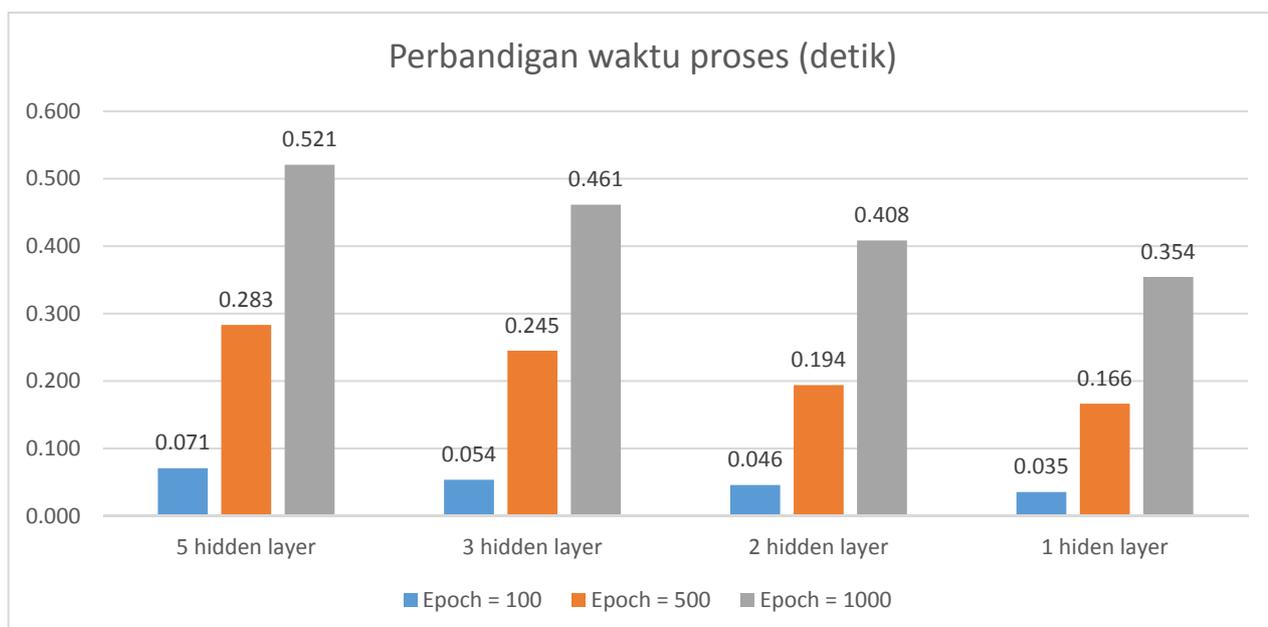
Gambar 4. Contoh grafik error pada setiap epoch dengan perulangan sebanyak 1000 epoch dan 500 epoch proses training. Dengan jumlah 5 neuron pada 1 hidden layer

Tabel 3. Hasil pengujian dengan Epoch 500

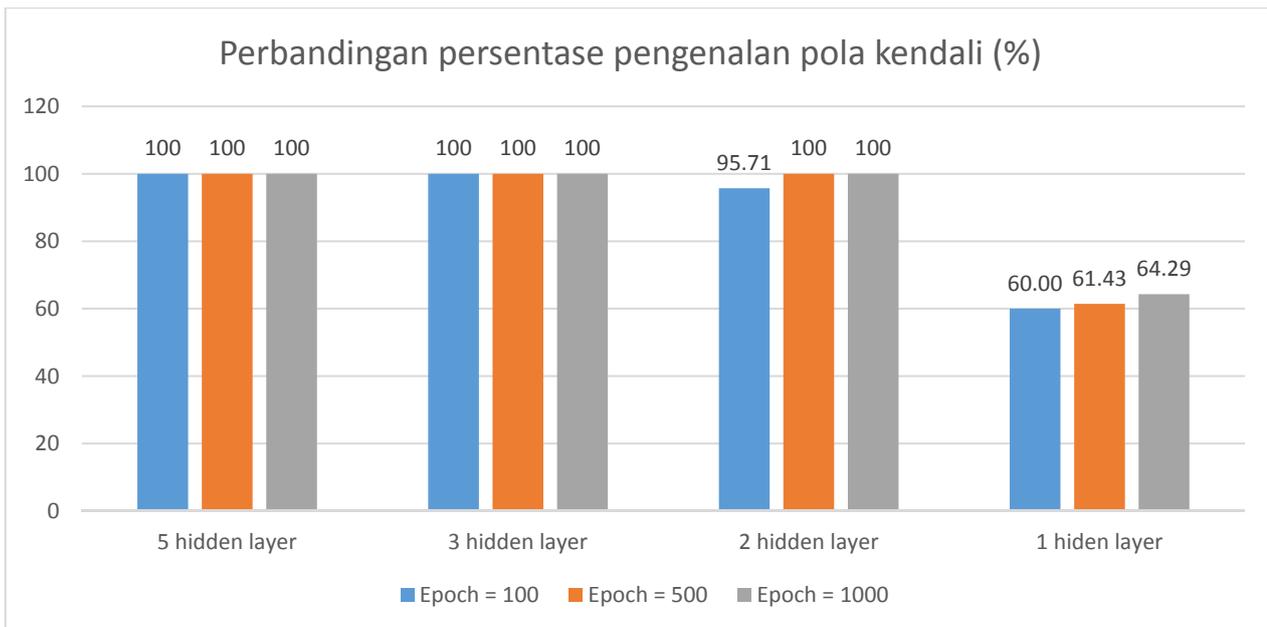
| Epoch = 500 Perulangan ke- | hidden layer : 5 | | hidden layer : 3 | | hidden layer : 2 | | hidden layer : 1 | |
|-------------------------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|
| | waktu (s) | % Pengenalan |
| 1 | 0.480 | 100 | 0.422 | 100 | 0.224 | 100 | 0.191 | 57.14 |
| 2 | 0.239 | 100 | 0.260 | 100 | 0.192 | 100 | 0.181 | 57.14 |
| 3 | 0.276 | 100 | 0.248 | 100 | 0.190 | 100 | 0.166 | 71.43 |
| 4 | 0.281 | 100 | 0.220 | 100 | 0.190 | 100 | 0.161 | 57.14 |
| 5 | 0.288 | 100 | 0.250 | 100 | 0.191 | 100 | 0.160 | 57.14 |
| 6 | 0.237 | 100 | 0.215 | 100 | 0.192 | 100 | 0.162 | 57.14 |
| 7 | 0.245 | 100 | 0.215 | 100 | 0.191 | 100 | 0.162 | 57.14 |
| 8 | 0.261 | 100 | 0.205 | 100 | 0.189 | 100 | 0.161 | 71.43 |
| 9 | 0.256 | 100 | 0.207 | 100 | 0.189 | 100 | 0.160 | 57.14 |
| 10 | 0.268 | 100 | 0.208 | 100 | 0.191 | 100 | 0.160 | 71.43 |
| Rerata | 0.283 | 100 | 0.245 | 100 | 0.194 | 100 | 0.166 | 61.43 |

Tabel 4. Hasil pengujian dengan Epoch 1000

| Epoch = 1000 Perulangan ke- | hidden layer : 5 | | hidden layer : 3 | | hidden layer : 2 | | hidden layer : 1 | |
|--------------------------------|------------------|--------------|------------------|--------------|------------------|--------------|------------------|--------------|
| | waktu (s) | % Pengenalan |
| 1 | 0.801 | 100 | 0.739 | 100 | 0.647 | 100 | 0.514 | 57.14 |
| 2 | 0.515 | 100 | 0.418 | 100 | 0.387 | 100 | 0.392 | 71.43 |
| 3 | 0.504 | 100 | 0.411 | 100 | 0.392 | 100 | 0.344 | 71.43 |
| 4 | 0.481 | 100 | 0.469 | 100 | 0.380 | 100 | 0.341 | 57.14 |
| 5 | 0.485 | 100 | 0.468 | 100 | 0.382 | 100 | 0.345 | 71.43 |
| 6 | 0.482 | 100 | 0.435 | 100 | 0.381 | 100 | 0.321 | 57.14 |
| 7 | 0.492 | 100 | 0.431 | 100 | 0.378 | 100 | 0.320 | 57.14 |
| 8 | 0.477 | 100 | 0.414 | 100 | 0.380 | 100 | 0.319 | 71.43 |
| 9 | 0.488 | 100 | 0.414 | 100 | 0.382 | 100 | 0.324 | 57.14 |
| 10 | 0.480 | 100 | 0.414 | 100 | 0.375 | 100 | 0.323 | 71.43 |
| Rerata | 0.521 | 100 | 0.461 | 100 | 0.408 | 100 | 0.354 | 64.29 |



Gambar 5. Grafik perbandingan waktu proses



Gambar 6. Grafik perbandingan persentase pengenalan pola kendali

B. Pembahasan

Dari tabel 2, 3 dan 4 dapat kita Analisa nilai rerata dengan menggunakan representasi grafik seperti pada Gambar 5 dan Gambar 6. Gambar 5 menunjukkan bahwa waktu proses paling cepat terdapat pada jumlah neuron 1 dengan epoch 100. Sedangkan waktu terlama didapatkan pada saat menggunakan jumlah neuron 5 dan epoch sebanyak 1000.

Fakta ini telah sesuai dengan apa yang terjadi di dalam eksekusi program, bahwasanya semakin banyak jumlah neuron artinya semakin banyak proses komputasinya, semakin banyak proses komputasi maka semakin banyak pula waktu yang dibutuhkan.

Selain itu, jumlah epoch juga mempengaruhi waktu proses. Hal ini disebabkan karena proses perulangan semakin banyak sehingga memerlukan waktu yang juga semakin lama.

Selanjutnya dari Gambar 6 diperoleh nilai pengenalan untuk jumlah neuron hidden layer = 1 tidak pernah mencapai 100%, untuk neuron hidden layer = 2 ada satu variasi yang dibawah 100% yaitu saat epoch = 100. Selebihnya untuk jumlah neuron 3, 4 dan 5 capaian pengenalan pola kendalinya semua mencapai angka 100%.

Dari terungkapnya fakta-fakta di atas maka untuk mendapatkan nilai pengenalan pola 100% kita tidak dapat menggunakan jaringan dengan 1 neuron di hidden layer meskipun secara komputasi waktu yang dibutuhkan paling cepat.

Pilihan paling baik adalah menggunakan jaringan dengan 2 neuron di hidden layer dengan minimal adalah 500 epoch. Karena pada saat ini jaringan sudah memiliki akurasi 100% dengan waktu proses rerata selama 0.194 detik saja. Artinya kita dapat menggunakan bobot-bobot dari jaringan dengan 2 neuron di hidden layer tersebut ke dalam sebuah mikrokontroler sebagai implementasi hardware yang dihubungkan dengan perangkat input dan output untuk mengendalikan lampu sein pada sepeda motor.

IV. KESIMPULAN

Dalam penelitian bahwa secara simulasi, jaringan syaraf tiruan backpropagation dapat digunakan untuk mendesain sebuah logika kontrol dalam mengendalikan lampu sein pada sepeda motor.

Dari penelitian juga terungkap bahwa untuk mendapatkan proses pengendalian yang sempurna (100%) dan waktu yang paling singkat maka setidaknya diperlukan 2 neuron pada lapisan hidden layer.

Saran penelitian berikutnya adalah untuk melakukan aplikasi hardware sistem pengendalian lampu sein sepeda motor ini dengan mengambil logika kontrol dan bobot-bobot yang diperoleh dari proses learning jaringan.

DAFTAR PUSTAKA

- [1] B. F. Rahmat, N. C. Basjaruddin, and D. Saefudin, "SISTEM PENGHINDAR TABRAKAN DEPAN-BELAKANG KOOPERATIF BERBASIS LOGIKA FUZZY," in *Prosiding Industrial Research Workshop and National Seminar*, 2019, vol. 10, pp. 155–161.
- [2] D. Nur, "Perancangan Sistem Peringatan Antar Kendaraan Untuk Peningkatan Keselamatan Berkendara di Jalan," 2019.
- [3] F. R. Adha, M. Yusro, and P. Yuliatmojo, "SISTEM LAMPU SEIN MATI OTOMATIS, DETEKSI TITIK BUTA PENGENDARA, DAN ENGINE STOP BERBASIS ARDUINO PADA SEPEDA MOTOR," *J. Pendidik. VOKASIONAL Tek. Elektron. JVoTE*, vol. 1, no. 1, pp. 18–21, 2018.
- [4] A. Natanael, "Implementasi Logika Fuzzy pada Sistem Kendali Lampu Sein Motor," PhD Thesis, Universitas Gadjah Mada, 2016.
- [5] Y. B. Praharto and C. Wicaksana, "Perancangan Alat Penetral Lampu Sein Sepeda Motor Otomatis Berbasis ARDUINO R3," *ITEKS*, vol. 8, no. 1, 2016.
- [6] Y. TAMAYA, "PENGATUR ON/OFF SEMI-OTOMATIS PADA LAMPU SEIN SEPEDA MOTOR DENGAN SENSOR LIMIT SWITCH BERBASIS ARDUINO UNO," PhD Thesis, Universitas Gadjah Mada, 2015.
- [7] F. A. Umaroh, "PORTAL BESI OTOMATIS LAMPU MERAH: PENGGUNA JALAN DIPAKSA UNTUK PATUH TERHADAP PERATURAN BERKENDARA," 2019.
- [8] H. M. Putri, A. S. Handayani, S. Soim, and M. I. Akbar, "Intelligent Transportation System dalam Sistem Monitoring Kecelakaan Lalu Lintas," in *Annual Research Seminar (ARS)*, 2019, vol. 4, pp. 283–287.
- [9] A. R. Budiman, D. W. Sudiharto, T. Brotoharsono, and E. Ariyanto, "DESAIN DAN PERANCANGAN HELM PINTAR DENGAN NOTIFIKASI KESELAMATAN BERKENDARA UNTUK PENGENDARA SEPEDA MOTOR," *Pros. SNST Fak. Tek.*, vol. 1, no. 1, 2018.
- [10] I. Dwisaputra, P. Silalahi, B. Cahyawan, and I. Akbar, "Lampu Sein Helm Sepeda Berbasis Voice Recognition," *Manutech J. Teknol. Manufaktur*, vol. 11, no. 1, pp. 20–25, 2019.
- [11] H. Fahrian, S. Munahar, and D. S. Putra, "Pengembangan Sirkuit Security System untuk Meningkatkan Driver Behaviour Control pada Kendaraan," *Automot. Exp.*, vol. 1, no. 1, pp. 13–19, 2018.
- [12] "Korlantas POLRI | Call Center : 1500669 | SMS Center : 9119." [Online]. Available: http://korlantas.polri.go.id/artikel/korlantas/113?Statistik_Laka. [Accessed: 17-Oct-2019].
- [13] J. Jama and W. Wagino, *Teknik Sepeda Motor*, vol. Jilid 1. Jakarta: Direktorat Pembinaan Sekolah Menengah Kejuruan.
- [14] Y. Tamaya and M. T. Ir. Sri Lestari, "PENGATUR ON/OFF SEMI-OTOMATIS PADA LAMPU SEIN SEPEDA MOTOR DENGAN SENSOR LIMIT SWITCH BERBASIS ARDUINO UNO," Universitas Gadjah Mada, 2015.
- [15] R. Wiryadinata and D. A. Ratnawati, "Simulasi Jaringan Syaraf Tiruan Berbasis Metode Backpropagation Sebagai Pengendali Kecepatan Motor DC," *Semin. Nas. Apl. Teknol. Inf. SNATI*, vol. 0, no. 0, 2005.
- [16] D. S. Putra, "Pemodelan Sistem dengan Metode Neural Network Back Propagation Modeling System Using Neural Network Backpropagation," *J. Ilm. Poli Rekayasa*, vol. 11, no. 2, pp. 22–31, 2016.
- [17] M. Azmi, "Komparasi metode jaringan syaraf tiruan SOM dan LVQ untuk mengidentifikasi data bunga iris," *J. TeknoIf*, vol. 2, no. 1, 2014.

Biodata Penulis

Meri Azmi, menyelesaikan master di UGM Jogjakarta saat ini aktif sebagai dosen jurusan Teknologi Informasi di Politeknik Negeri Padang sejak tahun 2006. Memiliki minat riset pada bidang kecerdasan buatan.

Dwi Sudarno Putra, adalah dosen jurusan Teknik Otomotif Universitas Negeri Padang dengan salah satu bidang riset pada sistem kendali dan sistem elektronika otomotif. Menyelesaikan master di Departemen Teknik Elektro Universitas Indonesia.

Wawan Purwanto, adalah dosen jurusan Teknik Otomotif Universitas Negeri Padang dengan salah satu bidang riset pada sistem kendali otomotif.

Menyelesaikan program doktoral di NKFUST (National Kaohsiung First University of Science and Technology), Kaohsiung, Taiwan.

Toto Sugiarto, adalah dosen jurusan Teknik Otomotif Universitas Negeri Padang dengan salah satu bidang riset pada sistem ototronik. Menyelesaikan program S2 di Universitas Andalas.

Donny Fernandez, adalah dosen jurusan Teknik Otomotif Universitas Negeri Padang. Mendapatkan gelar master dari Universitas Gadjah Mada Jogjakarta