

Implementasi Jaringan Syaraf Tiruan Pada Kendali Lampu Sorot Mobil Adaptif Berbasis Python

Farras Timorremboko*, Oki Teguh Karya

Teknik Elektro, Universitas Mercu Buana, Jakarta

*41418320045@student.mercubuana.ac.id

Abstrak— Fungsi utama dari lampu jalan untuk memastikan keamanan manusia. Penerangan lalu lintas diharuskan memberikan kondisi visibilitas yang baik dan mengurangi potensi bahaya dengan menerangi objek di sepanjang jalan. Jaringan Syaraf Tiruan diharapkan menghasilkan model terbaik untuk mengendalikan intensitas lampu sorot mobil adaptif pada kondisi yang sesuai dengan lapangan yaitu kondisi terang, mendung dan malam hari. Data diperoleh dari alat bantu yang terdiri dari 5 buah sensor cahaya dan 2 buah LED. Model terbaik didapat melalui *training* beberapa bentuk model Jaringan Syaraf Tiruan dan prediksi intensitas cahaya lampu sorot mobil berdasarkan *dataset training* dan *testing*. *Training* dilakukan pada 12 model berbeda dengan merubah banyak *neuron hidden layer* dan fungsi aktivasi pada program Jaringan Syaraf Tiruan. Model Jaringan Syaraf Tiruan terbaik memiliki parameter 20 *node hidden layer*, fungsi aktivasi Relu dan epoch 200 dengan error *training* sebesar 0,0038 dan hasil error prediksi sebesar 147,12.

Kata Kunci— Jaringan Syaraf Tiruan, Kendali, Komputer, LED, Sensor Cahaya.

DOI: 10.22441/jte.2020.v11i3.006

I. PENDAHULUAN

Fungsi utama dari lampu jalan untuk memastikan keamanan manusia. Penerangan lalu lintas diharuskan memberikan kondisi visibilitas yang baik dan mengurangi potensi bahaya dengan menerangi objek di sepanjang jalan. Berkendara di kegelapan sangat berbeda saat siang hari. Ruang penglihatan manusia lebih sempit tanpa bantuan pencahayaan alami sehingga diperlukan alat bantu dalam mengendarai seperti lampu sorot pada mobil. Namun, dengan banyaknya kondisi yang dinamis di jalanan, maka diperlukan sistem penerangan yang mumpuni. Kondisi pada dunia nyata tidak sebatas siang dan malam saja, namun ada kondisi hujan, kondisi malam hari di pedesaan, kondisi malam hari di perkotaan dan sebagainya. Sehingga, kasus ini diperlukan sebuah sistem pencahayaan sorot mobil yang lebih adaptif dan fleksibel dengan perubahan yang akan dihadapi oleh kendaraan tersebut dengan tujuan memudahkan pengemudi untuk berkendara di segala kondisi.

Jaringan Syaraf Tiruan adalah pengaturan yang dapat diterapkan untuk mensimulasikan perilaku objek tanpa algoritma namun memanfaatkan data eksperimen yang tersedia. Jaringan Syaraf Tiruan adalah sebuah teknik dalam meniru struktur sistem syaraf manusia dan mempelajari pola-pola dengan cara pengamatan. Hal tersebut dapat dilakukan oleh komputer dengan teknik Jaringan syaraf Tiruan melalui sebuah langkah yang disebut *training* atau pembelajaran pola data

historis sebuah kasus dan melakukan *testing* untuk menguji kehandalan model dalam memprediksi. Semakin banyak data atau pengalaman yang di-*input* kedalam komputer, maka semakin baik pula hasil prediksinya.

Salah satu teknik Jaringan Syaraf Tiruan yang paling populer adalah *backpropagation*. Secara singkat, Jaringan Syaraf Tiruan bekerja dengan melakukan komputasi penjumlahan nilai input dengan bobot menuju *hidden layer* dan menghasilkan *output*, dimana teknik ini disebut *feedforward*, maka langkah selanjutnya adalah *backpropagation*. Teknik ini akan melakukan optimasi dari nilai-nilai setiap bobot sehingga *output* prediksi dengan *output real* memiliki perbedaan atau *error* sekecil mungkin

Penelitian ini akan melakukan implementasi Jaringan Syaraf Tiruan dalam mencari model terbaik dalam sistem kontrol lampu sorot mobil adaptif. Penelitian dilaksanakan melalui pengambilan data dari lapangan yang didapat dari alat bantu berupa sensor dan LED.

II. PENELITIAN TERKAIT

Jaringan Syaraf Tiruan atau *Artificial Neural Network* merupakan teknik permodelan komputasi yang terinspirasi dari sistem syaraf biologis seperti otak manusia. Metode ini dapat digunakan untuk permodelan sistem fisik yang kompleks dan tidak memerlukan representasi matematika eksplisit. Jaringan syaraf Tiruan muncul sebagai teknologi untuk permodelan dan peramalan karena struktur matematika yang fleksibel mampu mendeskripsikan relasi non-linear yang kompleks antara data *input* dan *output*. Jaringan Syaraf Tiruan belajar dari suatu objek yang sedang dihadapinya karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data-data sebelumnya dan mengenal pola data yang selalu berubah. Jaringan Syaraf Tiruan juga merupakan sistem yang tak terprogram secara khusus, artinya hanya butuh satu program umum. Hal itu dapat membuat semua keluaran atau kesimpulan yang ditarik oleh jaringan berdasarkan pengalamannya selama mengikuti proses pembelajaran atau pelatihan.

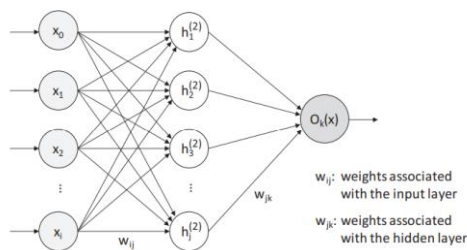
Pembelajaran atau pelatihan Jaringan syaraf Tiruan dibagi menjadi 2 berdasarkan tipe proses pembelajarannya yaitu:

1. *Supervised Learning* merupakan metode Jaringan syaraf Tiruan yang paling banyak digunakan. Metode dimulai apabila sebuah permasalahan memiliki *input* variabel (x) dan *output* variabel (y) dan data tersebut dipelajari polanya oleh Jaringan syaraf Tiruan menggunakan sebuah algoritma. Tujuan dari pembelajaran pola hubungan dari *input* dan *output* adalah apabila terdapat

nilai masukkan baru, maka sistem dapat memprediksi *output* yang akan keluar.

2. *Unsupervised Learning* adalah metode yang hanya memiliki *input* data (x) namun tidak memiliki nilai *output*. Tujuan dari *unsupervised learning* adalah untuk memodelkan struktur atau distribusi yang mendasari data dengan tujuan sebagai mempelajari lebih lanjut tentang data tersebut. Metode ini disebut *unsupervised learning* karena tidak seperti *supervised learning*, metode ini tidak memiliki jawaban yang benar.

Jaringan Syaraf Tiruan terdiri dari 3 lapisan: lapisan *input*, lapisan tersembunyi, lapisan *output*:



Gambar 1. Topologi Jaringan Syaraf Tiruan

- *Input Layer* menampilkan informasi dari dunia luar ke dalam jaringan dan beberapa variabel *input* bersatu disebut lapisan *input*.
- *Hidden Layer* atau lapisan tersembunyi tidak memiliki koneksi ke dunia luar. Lapisan tersebut melakukan komputasi dan melakukan pengiriman informasi dari lapisan *input* ke lapisan *output*.
- *Output Layer* adalah kumpulan dari beberapa *output nodes* dan lapisan ini memiliki tanggung jawab untuk proses komputasi dan pengiriman informasi dari jaringan ke dunia luar.

Proses *feed-forward* mengalirkan informasi *input* melalui fungsi aktivasi menuju lapisan *output*. Elemen proses mendasar dari Jaringan syaraf Tiruan adalah *neuron*, dimana setiap *neuron* akan tersambung dengan *neuron* lainnya melalui sinapsis. Setiap *neuron* menerima masukan nilai dari beberapa *node input* melalui sinapsis yang memiliki nilai bobot. Pada penelitian ini bobot awal menggunakan metode *random*. Formulasi dari perhitungan *input* berbobot adalah

$$n = \sum_{i=1}^R w_i p_i + b \tag{1}$$

dimana p_i dan w_i adalah nilai *input* dari *dataset* dan bobot sinapsis, sedangkan b adalah bias dari *neuron*.

Penjumlahan dari *input* berbobot dengan bias akan diaktivasi dengan fungsi aktivasi. Terdapat beberapa tipe fungsi aktivasi pada *neuron* yang digunakan pada Jaringan Syaraf Tiruan, termasuk fungsi *sigmoid*, *hyperbolic tangents* dan *Rectified Linear Units* (ReLU). *Sigmoid* akan menerima angka tunggal dan mengubah nilai x menjadi sebuah nilai yang memiliki *range* mulai dari 0 sampai 1. Rumus dari fungsi aktivasi *sigmoid* adalah

$$f(n) = \frac{1}{1 + e^{-n}} \tag{2}$$

hyperbolic tangents akan mengubah nilai *input* x menjadi sebuah nilai yang memiliki *range* mulai dari -1 sampai 1. Kelebihan *hyperbolic tangents* dari pada *sigmoid* adalah *output* yang dimiliki *hyperbolic tangents* merupakan *zero-centered*.

$$\tanh(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \tag{3}$$

ReLU atau *Rectified Linear Unit* adalah salah satu fungsi aktivasi yang paling mudah untuk dimengerti karena intinya ReLU hanya membuat pembatas pada bilangan nol. Apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$.

$$f(x) = \max(0, x) \tag{4}$$

BackPropagation secara efisien mengkalkulasi perubahan setiap sinapsis yang berasal dari penyimpangan jaringan dari tujuan (disebut juga dengan *error*). *Backpropagation* menggunakan hasil dari perhitungan untuk membuat sedikit modifikasi untuk mengurangi *error* pada jaringan. *Backpropagation* merupakan tahap kelanjutan dari proses *feed-forward*. Apabila lapisan *output* gagal mencapai nilai *output* yang diinginkan, *error* antara *output* aktual dan *output* yang diharapkan dihitung lapisan demi lapisan secara rekursif, dan sinyal *error* cenderung diminimalkan sesuai dengan *error*.

Pada tahap *backpropagation* alur perhitungan hampir sama dengan *feed-forward*. Nilai *error* antara *output* yang diinginkan dengan *output* aktual dikirim ke *hidden node* melalui fungsi aktivasi yang bersifat differensial. Mencari *error* menggunakan rumus

$$\delta k = (tk - yk)f'(yink) \tag{5}$$

Setelah nilai *error* dialirkan ke seluruh jaringan, maka akan didapatkan nilai selisih dari setiap bobot sinapsis menggunakan rumus

$$\Delta wjk = \alpha \delta k zj \tag{6}$$

Untuk mencari selisih bobot terdapat *hyper*-parameter yang bernama *learning rate*(alpha). Lalu setiap selisih bobot tersebut akan ditambah dengan nilai bobot awal sehingga menghasilkan nilai bobot yang baru.

$$wjk \text{ (baru)} = wjk \text{ (lama)} + \Delta wjk \tag{7}$$

III. METODOLOGI PENELITIAN

Pada implementasi Jaringan Syaraf Tiruan pada kendali lampu sorot mobil adaptif terdiri dari 3 tahap, yaitu tahap pengumpulan data, tahap *training model* dan tahap *testing model*. Data yang akan diolah oleh Jaringan Syaraf Tiruan perlu dikumpulkan terdahulu melalui sebuah alat bantu berupa kumpulan sensor-sensor. Apabila data yang didapat sudah tercukupi, maka tahap selanjutnya adalah *training* dan *testing* model Jaringan Syaraf Tiruan dengan beberapa variabel yang diubah-ubah seperti jumlah *neuron hidden layer* dan fungsi aktivasi.

Tahap pertama adalah penyiapan alat bantu untuk mengumpulkan data dari lapangan. Tahap kedua melakukan pengumpulan data menggunakan alat bantu yang sudah dibuat. Data yang sudah terkumpul akan dibagi menjadi *dataset training* dan *dataset testing*. Tahap *training* dan *testing* akan dilakukan beberapa kali dengan merubah parameter pada model jaringan. Apabila sudah mendapatkan hasil *training*, tahap selanjutnya adalah analisa setiap hasil dari semua model. Tahap terakhir adalah menentukan model apa yang terbaik dalam kendali lampu sorot mobil adaptif dan melakukan prediksi terhadap *dataset testing*.

A. Pengumpulan Data

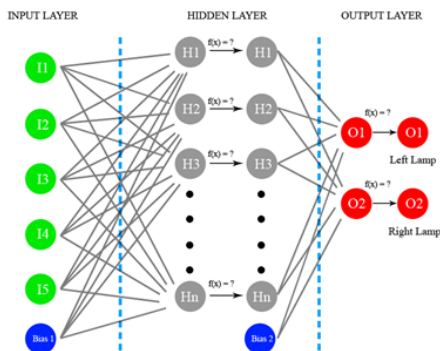
Tahap pengumpulan data membutuhkan alat bantu berupa kumpulan sensor. Alat bantu terdiri dari beberapa komponen elektronik yaitu Arduino, sensor LDR, LCD *display* dan lampu LED. Arduino sebagai otak dari alat bantu ini bertugas untuk mengumpulkan data dari pembacaan sensor LDR dimana data tersebut akan ditampilkan pada LCD *display*. Intensitas cahaya lampu LED akan dikendalikan pula oleh Arduino. Sensor LDR yang digunakan sebanyak 5 buah. Perbedaan antar sensor tersebut terletak pada sudut pengambilan data cahaya yaitu sejauh 45 derajat. Dengan pengambilan data dari segala arah tersebut, diharapkan penulis dapat mendapatkan data intensitas cahaya yang sebenar-benarnya pada kondisi di lapangan.



Gambar 2. Skema Alat Bantu

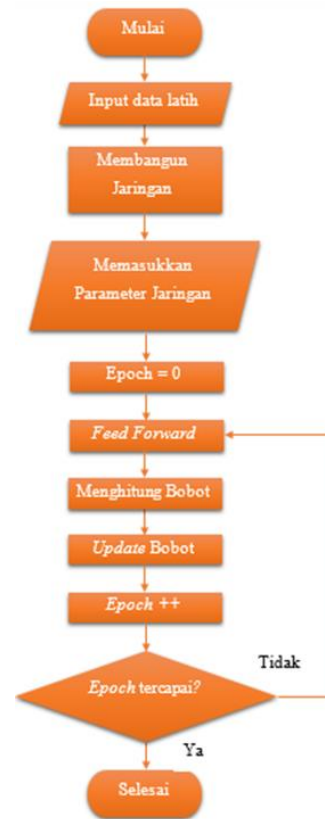
B. Training Model

Data yang sudah terkumpul akan dipakai sebagai bahan *training* dan *testing* model Jaringan Syaraf Tiruan. Data sebagai *training* dipergunakan untuk mendapatkan model Jaringan syaraf Tiruan yang menghasilkan *error* kecil melalui metode *BackPropagation*. Hasil dari *training* berupa bobot-bobot setiap sinapsis dan bobot bias.



Gambar 4. Konsep Model Jaringan Syaraf Tiruan

Secara konsep, model jaringan terdiri dari 1 *input layer*, 1 *hidden layer* dan 1 *output layer*. *Input layer* terdiri dari 5 sensor cahaya dan *output layer* terdiri dari 2 lampu, sedangkan *node hidden layer* diubah jumlahnya menyesuaikan model yang diinginkan.



Gambar 4. Flowchart Jaringan Syaraf Tiruan

Alur dari tahap *training* Jaringan Syaraf Tiruan tergambar pada *flowchart* diatas. Awal dari semua tahap adalah input *dataset training*. Lalu, jaringan dibangun dengan menentukan parameter seperti banyak *node hidden layer* dan *activation function*. Setelah parameter sudah ditentukan, *training* akan dimulai. *Training* dimulai pada perhitungan matematika *feedforward* dan dilanjutkan *backpropagation*, tahap ini dilakukan berulang-ulang sampai maksimum *epoch* atau batas maksimum pengulangan telah tercapai. Setelah maksimum *epoch* tercapai, program *training* akan menampilkan *error*, nilai bobot dan bias akhir.

Dataset Training

Array output			Array Input				
left_lamp	right_lamp		45deg_sensor	90deg_sensor	135deg_sensor	180deg_sensor	
0	50	50	72	47	39	34	39
1	50	50	52	48	45	53	57
2	80	80	96	92	82	76	80
3	200	200	841	882	909	926	940
4	50	50	21	26	23	38	52
...
795	50	50	80	33	27	26	65
796	250	250	919	924	922	925	919
797	80	80	84	83	76	77	86
798	200	200	856	869	892	898	926
799	50	50	56	52	51	56	44

800 rows x 7 columns

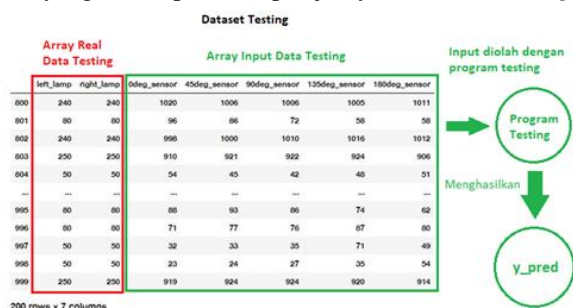
Gambar 5. Array Input Output Dataset Training

Program Jaringan Syaraf Tiruan untuk kendali lampu sorot mobil adaptif dibagi menjadi beberapa segmen.

- Segmen pertama adalah pemanggilan *tools* dan *dataset training*. *Tools* yang dipakai adalah *numpy*, *keras*, *matplotlib* dan *sklearn*.
- *Dataset training* terbagi menjadi 2 bagian yaitu bagian *array input* yang dilabel sebagai variabel *x* dan bagian *array output* yang dilabel variabel *y* lalu data tersebut dinormalisasi.
- Segmen ketiga penentuan banyak *layer* dan fungsi aktivasi. Banyaknya *layer* terdiri dari banyak *neuron input*, *neuron hidden layer* dan *neuron output*.
- *Training* pada model yang sudah dibangun.
- Menganalisa error setiap *epoch* dalam bentuk grafik

C. Testing Model

Tahap terakhir dari penelitian ini adalah *testing* model. Model Jaringan Syaraf Tiruan yang sudah ditetapkan sebagai model terbaik akan diuji kehandalannya dengan program *testing*. Program *testing* merupakan program Jaringan Syaraf Tiruan yang berfungsi untuk memprediksi nilai *output* berdasarkan *dataset* yang belum pernah dipelajari yaitu *dataset testing*.



Gambar 7. Array Input Output Dataset Testing

Dataset testing dibagi menjadi 2 bagian juga yaitu *array input* dan *array output* atau *array real data testing*. *Array input* diolah menggunakan program *testing* yaitu Jaringan Syaraf Tiruan dengan model terbaik tanpa *backpropagation* dan menghasilkan nilai prediksi. Segmen akhir dari program *testing* adalah mencari error antara nilai prediksi dengan nilai *real dataset testing*. Error setiap data akan dikalkulasikan rata-ratanya dengan metode *mean squared error*. Error tersebut merupakan error *real* dan berbeda dengan error saat *training* karena error *real* merupakan hasil simulasi seperti keadaan nyata

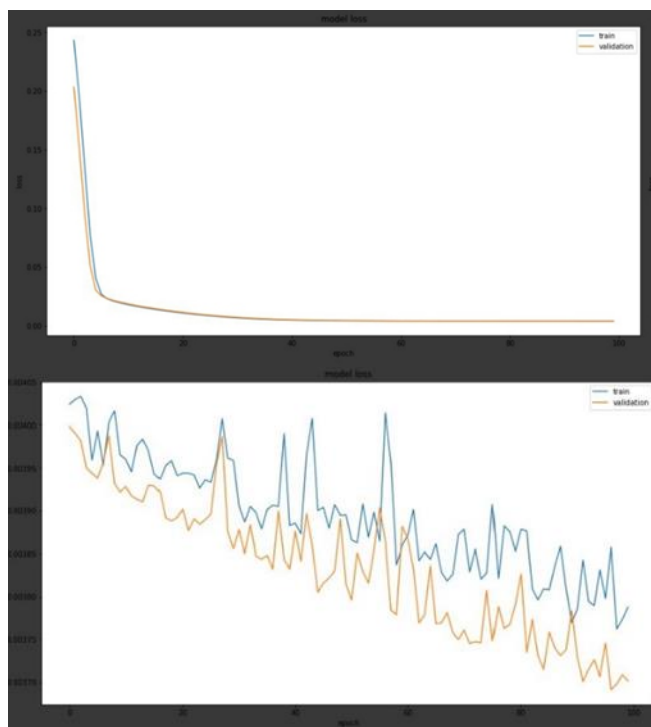
IV. HASIL DAN ANALISA

Pada penelitian ini, terdapat 2 parameter yang diuji dalam mempelajari *dataset training* yaitu jumlah *nodes* pada *hidden layer* dan fungsi aktivasi. *Hidden layer* yang diuji ada 4 variabel yaitu 10 *nodes*, 20 *nodes*, 30 *nodes* dan 40 *nodes*. Fungsi aktivasi terdapat 3 jenis yang akan digunakan yaitu Relu, Sigmoid dan Tanh. Oleh karena itu, terdapat 12 model yang akan *training* pada kasus lampu sorot mobil adaptif.

Tabel 1. Error *Training* Model Jaringan Syaraf Tiruan

No.	Fungsi Aktivasi	Jmlah Nodes	Epoch	Error Training	Error Validasi
1	Sigmoid	10	200	0.0105	0.0108
2	Sigmoid	20	200	0.0105	0.0109
3	Sigmoid	30	200	0.0107	0.0113
4	Sigmoid	40	200	0.0110	0.0117
5	Relu	10	200	0.0225	0.0236
6	Relu	20	200	0.0038	0.0037
7	Relu	30	200	0.0039	0.0038
8	Relu	40	200	0.0038	0.0036
9	Tanh	10	200	0.0070	0.0071
10	Tanh	20	200	0.0067	0.0068
11	Tanh	30	200	0.0065	0.0067
12	Tanh	40	200	0.0065	0.0066

Hasil error *training* disetiap jenis fungsi aktivasi menunjukkan nilai yang cukup berbeda. Error validasi setiap model menunjukkan nilai yang hampir sama dengan error *training*. Berdasarkan Tabel 1, fungsi aktivasi yang menghasilkan nilai error *training* terkecil adalah Relu yaitu antara 0.0038 hingga 0.0225. Sedangkan fungsi aktivasi yang menghasilkan nilai error terbesar adalah sigmoid yaitu antara 0.0105 hingga 0.0110. Tanh menghasilkan nilai error diantara kedua fungsi aktivasi lainnya yaitu antara 0.0065 hingga 0.0070. Data ini menunjukkan bahwa fungsi aktivasi yang terbaik adalah Relu



Gambar 7. Karakteristik *Training* Model Terbaik

Hasil dari *training* model berupa karakteristik error setiap *epoch* dan nilai error pada *epoch* terakhir. Harus diingat bahwa error tersebut adalah error dengan data yang ternormalisasi. Karakteristik error berupa grafik seperti Gambar 8. Semakin

naik *epoch*, maka semakin kecil errornya. Apapun parameter model Jaringan Syaraf Tiruan akan menghasilkan error yang kecil. Namun, model yang akan dipilih sebagai model terbaik adalah model yang dapat menghasilkan error terkecil. Error terkecil ini dapat dilihat dari nilai error pada *epoch* terakhir pada setiap model Jaringan Syaraf Tiruan.

Gambar 8 merupakan grafik karakteristik saat *training* untuk model terbaik. Model terbaik yang terpilih memiliki parameter fungsi aktivasi Relu dan *node hidden layer* sebanyak 20. Model terbaik terpilih berdasarkan error terkecil dan jumlah *node hidden layer* yang tergolong sedikit. Grafik atas gambar 8 adalah penurunan error *training* dari 0-100 epoch dan grafik bawahnya merupakan penurunan error dari epoch 101-200.

Setelah melakukan *training* di setiap model pada *dataset training* yang sama, tahap selanjutnya adalah menentukan model yang terbaik dan melakukan prediksi pada *dataset testing*. Selain karakteristik pada grafik, hasil *training* berupa kumpulan bobot dan nilai bias di setiap sinapsisnya, dimana bobot dan bias tersebut akan digunakan untuk prediksi *dataset testing*. *Dataset Testing* dibagi menjadi 2 bagian yaitu bagian input sensor dan output LED *real*. Prediksi dilakukan berdasarkan data input sensor untuk menghasilkan prediksi output LED dan prediksi tersebut akan dibandingkan dengan output LED *real*. Selain data prediksi, hasil dari prediksi adalah seberapa besar error *prediction* rata-rata antara *output prediction* dengan *output LED real*.

Tabel 2. Sampel Perbandingan Output Prediction dengan Output Real

No.	Output Predict		Output Real	
	LED Left	LED Right	LED Left	LED Right
1	210	210	200	200
2	230	230	240	240
3	230	230	240	240
4	231	231	24	24
5	68	68	80	80
6	50	50	50	50
7	50	50	50	50
8	50	50	50	50
9	165	165	160	160
10	171	171	160	160
11	229	229	240	240
12	76	76	80	80
13	152	152	180	180
14	229	229	240	240
15	223	223	200	200
16	254	254	180	180
17	67	67	80	80
18	50	50	50	50
19	50	50	50	50
20	210	210	200	200

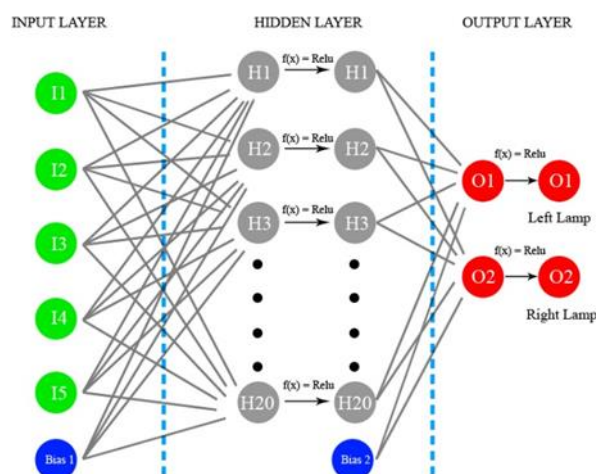
Tabel 2 merupakan sampel hasil prediksi model terbaik dari keseluruhan data *testing* yang berjumlah 200 data. Error *prediction* antara *output prediction* dan *output real* masih cukup besar, walaupun model yang dipilih merupakan model terbaik.

Beberapa data ada yang memiliki nilai prediksi yang melebihi nilai *real*, adapula yang kurang dari nilai *real*. Namun, ada beberapa data yang prediksinya sama dengan nilai *real*. Prediksi tersebut merupakan hasil terbaik karena apabila prediksi data menggunakan model yang lain, maka akan lebih besar lagi error *prediction* nya. Setiap baris data terdapat error, sehingga error *prediction* keseluruhan perlu dicari rata-rata. Oleh karena itu, error *prediction* rata-rata dicari menggunakan metode *mean square error*. Hasil error *prediction* rata-rata tersebut adalah 147.12.

Nilai error *prediction* yang dihasilkan pada model tersebut masih terbilang tinggi. Hal ini terjadi karena beberapa faktor, yaitu:

- *Dataset training* dan *testing* sulit untuk dipelajari dan diprediksi dengan Jaringan Syaraf Tiruan. *Dataset* cahaya *outdoor* dikumpulkan langsung dari jalan sehingga ada kemungkinan *noise* yang terekam. Selain itu, jumlah sensor mempengaruhi kesulitan *dataset*.
- Model yang didapat bukan paling sempurna. Ada kemungkinan lainnya untuk memperbaiki model yaitu menambah jumlah *hidden layer* dan menentukan jumlah *node* setiap *hidden layer*-nya. *Node* setiap *hidden layer* dapat berjumlah sama atau berbeda-beda.

Arsitektur model Jaringan Syaraf Tiruan yang terbaik memiliki beberapa parameter. Arsitektur terdiri dari 3 *layer* yaitu 1 *input layer*, 1 *hidden layer* dan 1 *output layer*. *Input layer* terdiri dari 5 input, *hidden layer* terdiri dari 20 *nodes* dan *output layer* terdiri dari 2 *nodes* output. Pada *hidden layer* dan *output layer* terdapat fungsi aktivasi yaitu Relu. Arsitektur dapat digambar seperti Gambar 8.



Gambar 8. Arsitektur Akhir Model Jaringan Syaraf Tiruan

V. KESIMPULAN

Model terbaik didapat melalui *training* beberapa bentuk model Jaringan Syaraf Tiruan dan prediksi intensitas cahaya lampu sorot mobil berdasarkan *dataset training* dan *testing*. *Training* dilakukan pada 12 model berbeda dengan merubah banyak *neuron hidden layer* dan fungsi aktivasi pada program Jaringan Syaraf Tiruan. Model Jaringan Syaraf Tiruan terbaik memiliki parameter 20 *node hidden layer*, fungsi aktivasi Relu

dan epoch 200 dengan error *training* sebesar 0,0038 dan hasil error prediksi sebesar 147,12.

UCAPAN TERIMA KASIH

Pada bagian ini dituliskan ucapan terima kasih terhadap pihak-pihak yang membantu terselesaikannya penelitian ini yaitu dosen pembimbing Universitas Mercu Buana dan teman-teman yang sudah membantu dan memberi semangat kepada penulis serta ucapan terima kasih terhadap tim editorial Jurnal Teknologi Elektro atas dipublikasikannya penelitian ini.

DAFTAR PUSTAKA

- [1] H. Alyuruk, "Introduction to R and Python. R and Python for Oceanographers", 1–21. 2019. <https://doi.org/10.1016/b978-0-12-813491-7.00001-4>
- [2] H. Bao, J. Wang, J. Li, H. Zhang, and F. Wu, "Effects of corn straw on dissipation of polycyclic aromatic hydrocarbons and potential application of backpropagation artificial neural network prediction model for PAHs bioremediation. *Ecotoxicology and Environmental Safety*", 186(September), 109745. 2019. <https://doi.org/10.1016/j.ecoenv.2019.109745>
- [3] A. Castañeda-Miranda and V. M. Castaño, "Smart frost control in greenhouses by neural networks models". *Computers and Electronics in Agriculture*, 137, 102–114. 2017 <https://doi.org/10.1016/j.compag.2017.03.024>
- [4] Y. S. Chen and S. C. Chiu, S. C., "New method of automatic control for vehicle headlights". *Optik*, 157, 718–723. 2018. <https://doi.org/10.1016/j.ijleo.2017.11.137>
- [5] A. Ekrias, M. Eloholma, L. Halonen, X. J. Song, X. Zhang, and Y. Wen, Road lighting and headlights: Luminance measurements and automobile lighting simulations. *Building and Environment*, 43(4), 530–536. 2008. <https://doi.org/10.1016/j.buildenv.2007.01.017>
- [6] T. Emoto, U. R. Abeyratne, K. Kawano, T. Okada, O. Jinnouchi, and I. Kawata, "Detection of sleep breathing sound based on artificial neural network analysis". *Biomedical Signal Processing and Control*, 41, 81–89. 2018. <https://doi.org/10.1016/j.bspc.2017.11.005>
- [7] F. Farhadi, and A. N. Lodi Vahid, "Learning activation functions in deep neural networks". *Département de Mathématiques et de Génie Industriel*, 151. 2017. <https://publications.polymtl.ca/2945/>
- [8] S. R. O. Y. Imantaka, "Universitas Indonesia Sistem Pengenal Wajah Berbasis Ensemble". 2010
- [9] E. Iyasele, "TO PROGRAMMING USING". February. 2020
- [10] R. E. Johnson, and J. W. Sensinger, "Actuator Technologies. In *Handbook of Biomechanics*". Elsevier Inc. 2019. <https://doi.org/10.1016/b978-0-12-812539-7.00002-7>
- [11] B. Karanayil, and M. F. Rahman, "Artificial Neural Network Applications in Power Electronics and Electric Drives. In *Power Electronics Handbook (4th ed.)*. Elsevier Inc. 2018. <https://doi.org/10.1016/b978-0-12-811407-0.00041-6>
- [12] Q. Khan, and Q. Bao, "Light-emitting devices. In *2D Materials for Photonic and Optoelectronic Applications*". Elsevier Ltd. 2019. <https://doi.org/10.1016/B978-0-08-102637-3.00007-3>
- [13] R. Kotschenreuther, R. "White LEDs for lighting applications. In *Nitride Semiconductor Light-Emitting Diodes (LEDs): Materials, Technologies, and Applications: Second Edition*. Elsevier Ltd. 2018. <https://doi.org/10.1016/B978-0-08-101942-9.00016-2>
- [14] T. P. Lillicrap and A. Santoro, "Backpropagation through time and the brain". *Current Opinion in Neurobiology*, 55, 82–89. 2019. <https://doi.org/10.1016/j.conb.2019.01.011>
- [15] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, S. "Activation Functions: Comparison of trends in Practice and Research for Deep Learning". 1–20. 2018. <http://arxiv.org/abs/1811.03378>
- [16] R. Prasetya, "Implementasi Jaringan Syaraf Tiruan Backpropagation untuk Memprediksi Ketinggian Air (Studi Kasus: Sungai Ciliwung)". *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 1(3), 297. 2018. <https://doi.org/10.30998/string.v1i3.1558>
- [17] B. A. Rizkin, K. Popovich, K and R. L. Hartman, "Artificial Neural Network control of thermoelectrically-cooled microfluidics using computer vision based on IR thermography". *Computers and Chemical Engineering*, 121, 584–593. 2019. <https://doi.org/10.1016/j.compchemeng.2018.11.016>
- [18] R. Sathya, and A. Abraham, "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification". *International Journal of Advanced Research in Artificial Intelligence*, 2(2). 2013. <https://doi.org/10.14569/ijarai.2013.020206>
- [19] M. H. SAZLI, "A brief review of feed-forward neural networks". *Communications, Faculty Of Science, University of Ankara, May*, 11–17. 2006. <https://doi.org/10.1501/0003168>
- [20] S. Sieniutycz, and Z. Szwast, "Neural Networks—A Review of Applications". *Optimizing Thermal, Chemical, and Environmental Systems*, 109–120. 2018. <https://doi.org/10.1016/b978-0-12-813582-2.00004-5>
- [21] A. K. Sreedhar, and B. L. Sharma, "Light Emitting Diodes". 1(March 2015), 1–13. 1973. https://doi.org/10.1007/978-1-349-16219-2_2
- [22] H. I. Suk, "An Introduction to Neural Networks and Deep Learning". In *Deep Learning for Medical Image Analysis (1st ed.)*. Elsevier Inc. 2017. <https://doi.org/10.1016/B978-0-12-810408-8.00002-X>
- [23] R. Toulson, and T. Wilmshurst, "Analog Input. Fast and Effective Embedded Systems Design", 91–109. 2017. <https://doi.org/10.1016/b978-0-08-100880-5.00005-0>