



## Desain dan simulasi *adaptive high power LED driver* menggunakan *feed-forward backpropagation neural network*

Muhammad Miftahuddin <sup>a,1</sup>, Eka Prasetyono <sup>a</sup>, Diah Septi Yanaratri <sup>a</sup>

<sup>a</sup>Program Studi Teknik Elektro Industri, Politeknik Elektronika Negeri Surabaya, Kampus ITS, Jl. Raya ITS, Keputih, Kec. Sukolilo Surabaya, 60111, Indonesia

<sup>1</sup>E-mail: [miftahuddin1244@gmail.com](mailto:miftahuddin1244@gmail.com)

### INFO ARTIKEL

#### Riwayat artikel:

Diajukan pada 25 Juni 2020

Direvisi pada 04 Juli 2020

Disetujui pada 15 Oktober 2020

Tersedia daring pada 15 November 2020

#### Kata kunci:

*Adaptive LED driver*, jaringan saraf tiruan, *light emitting diode*.

#### Keywords:

*Adaptive LED driver*, *feed-forward backpropagation neural network*, *light emitting diode*.

### ABSTRAK

Lampu LED (*light emitting diode*) merupakan salah satu jenis lampu hemat energi yang mempunyai *lifetime* yang panjang. Lampu LED memiliki dua bagian utama, yaitu LED modul dan LED *driver* modul. LED modul dan LED *driver* keduanya harus memiliki spesifikasi yang sama, apabila tidak sesuai maka LED modul tidak bisa menyala karena tegangan atau arus yang kurang dan bisa mengalami kerusakan karena melebihi batas arus dan tegangan. LED *driver* yang umum dipasaran hanya dapat digunakan untuk satu jenis LED modul saja, sehingga setiap LED modul dengan daya yang berbeda memerlukan LED *driver* yang berbeda-beda. Penelitian ini membahas tentang *adaptive LED driver* yang mampu menyesuaikan dengan kebutuhan LED modul. Metode yang digunakan adalah *feed-forward backpropagation neural network* (FF-BPNN) yang diadopsi dari cara kerja sistem saraf biologis. FF-BPNN terdiri dari *layer input*, *hidden layer*, dan *layer output*. Penggunaan metode ini berfungsi sebagai kontrol *driver* LED agar didapatkan daya yang sesuai dengan kebutuhan tiap lampu LED modul sehingga tidak terjadi *over current* dan *over voltage*. Pengujian simulasi adaptif LED *driver* dilakukan dengan 3 variasi daya LED modul yaitu sebesar 50 watt, 70 watt dan 100 watt. Hasil simulasi menunjukkan bahwa *driver* LED mampu menyesuaikan rating dari daya *led* yaitu sebesar 49.89 watt, 69.94 watt dan 99.42 watt.

### ABSTRACT

LED (light-emitting diode) lamps are one type of energy-saving lamp that has a long lifetime. The LED lamp has two main parts, i.e the LED module and the LED driver module. Both module LEDs and driver LEDs must have the same specifications, if they do not match, the module LEDs cannot turn on due to insufficient voltage or current and can be damaged because they exceed the current and voltage limits. The general LED driver in the market can only be used for one type of LED module, so each LED module with different power requires a different LED driver. This research discusses the adaptive LED driver that can suit the needs of the LED module. The method used is a feed-forward backpropagation neural network (FF-BPNN) which is adopted from the workings of the biological nervous system. FF-BPNN consists of an input layer, a hidden layer, and an output layer. The use of this method functions as an LED driver control to obtain power that is under the needs of each LED module lamp so that over current and over voltage do not occur. Adaptive LED driver simulation testing is done with 3 variations of LED module power, i.e 50 watts, 70 watts, and 100 watts. The simulation results show that the LED driver can adjust the rating of the led power which is 49.89 watts, 69.94 watts, and 99.42 watts.

Tersedia pada: <http://dx.doi.org/10.36055/tjst.v16i2.8411>.

## 1. Pendahuluan

Sejak ditemukan lampu pijar oleh Thomas Alfa Edison, kemudian ditemukan lampu jenis SL, TL dan kini lampu LED, lampu penerangan berkembang dengan pesat. LED merupakan perangkat keras dan padat (*solid-state component*) sejenis semikonduktor yang dapat mengubah energi listrik menjadi cahaya, sehingga lebih unggul dalam ketahanan (*durability*). Kelebihan LED adalah usia relatif panjang dibandingkan lampu jenis lainnya. LED memiliki umur



penggunaan hingga 30.000 jam dan dapat menghemat energi 80% sampai 90%. Dengan kelebihan tersebut, LED mempunyai kelemahan pada harga per lumen (satuan cahaya) lebih mahal dibandingkan dengan lampu jenis pijar, TL dan SL [1-2].

*Driver* LED adalah catu daya mandiri yang mengatur daya yang diperlukan untuk LED atau *array* LED. Tingkat daya yang dibutuhkan berubah sepanjang kenaikan dan penurunan suhu LED. Apabila menggunakan *driver* LED yang tidak benar, LED akan mengalami kenaikan suhu dan nilai arus yang tidak stabil sehingga mengakibatkan kegagalan kinerja yang buruk pada LED tersebut. *Driver* LED dapat diklasifikasikan menjadi dua tipe, yaitu *constant current* dan *constant voltage*. *Driver* LED arus konstan dirancang untuk rentang tegangan *output* yang ditentukan dan arus *output* tetap (mA). LED yang beroperasi pada *driver* arus konstan membutuhkan arus dalam satuan mili amps (mA) atau amp (A). Peningkatan arus yang lebih tinggi dapat membuat LED lebih terang, namun dapat menyebabkan kenaikan suhu yang berakibat mengurangi umur LED. *Driver* arus konstan adalah cara terbaik untuk mendorong LED daya tinggi karena mempertahankan kecerahan yang konsisten pada semua LED secara seri [3]. *Driver* tegangan konstan dirancang untuk LED yang memerlukan tegangan sesuai dengan tegangan kerja dari LED. *Driver* akan selalu menjaga tegangan tetap konstan meskipun terjadi penurunan tegangan arus bolak-balik (AC).

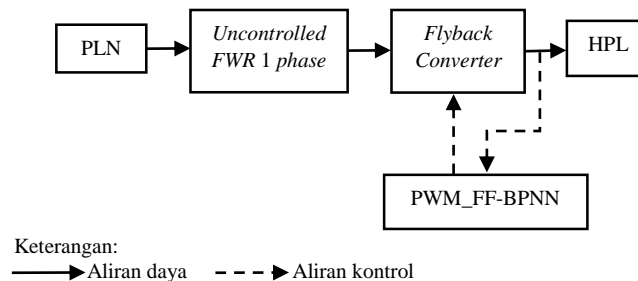
Penggunaan *constant current* dan *constant voltage* dapat dipengaruhi dari variasi jumlah LED yang tersusun secara seri ataupun paralel. Susunan LED seri dapat ditingkatkan untuk mengurangi jumlah *channel* pada *driver* LED [4-5], namun membutuhkan tegangan yang lebih tinggi. Jika tegangan lebih tinggi dari yang dibutuhkan LED, *driver* akan menghilangkan lebih banyak daya dan menjadi kurang efisien bahkan dapat mengurangi masa pakai LED [6]. Menggunakan sistem *adaptive LED driver* dapat menentukan tegangan *output* yang optimal berdasarkan *pulse with modulation* (PWM) dengan cara melakukan sistem pelacakan tegangan otomatis untuk berbagai LED secara seri atau secara paralel [7].

*Feed-forward backpropagation neural network* (FF-BPNN) adalah teknik pengolahan data yang diadopsi dari cara kerja sistem saraf biologis. FF-BPNN ini terdiri dari layer *input*, *hidden layer*, dan layer *output* serta bekerja secara propagasi maju (*feed-forward*) dan propagasi balik (*backpropagation*) [8]. *Feed-forward* hanya akan mengumpukan nilai *input* ke depan tanpa kembali lagi. Setelah nilai diumpan ke depan, selanjutnya *backpropagation* akan menghitung *error* dan berbalik menuju lapisan-lapisan neuron pada *hidden layer* untuk menyebarkan nilai tersebut. Pada penelitian ini, digunakan FF-BPNN sebagai optimasi pada tegangan *output* dari *flyback converter*, sehingga FF-BPNN dapat menyuplai tegangan pada HPL secara otomatis. Pemilihan metode *feed-forward backpropagation neural network* (FF-BPNN) karena sistem LED *driver* yang dikerjakan bersifat adaptif (*driver* dapat menyesuaikan daya dari modul LED), sehingga membutuhkan metode yang dapat melakukan pelacakan arus dan tegangan yang sesuai dengan *rating* dari modul LED dan sesuai dengan hasil *learning* yang diberikan.

## 2. Metodologi Penelitian

### 2.1. Blok Diagram Sistem

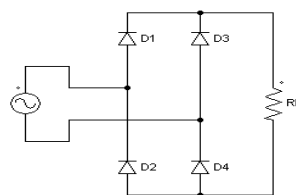
*Adaptive LED driver* digunakan dalam penelitian ini karena arus LED rentan rusak maupun berkurang umur pakainya apabila dialiri arus yang berlebih. Dari permasalahan tersebut dibutuhkan rangkaian *driver* yang mampu mengalirkan arus sesuai dengan *rating* LED itu sendiri. Gambar 1 menunjukkan sistem LED *driver* yang terdiri dari *rectifier* untuk mengubah dari sumber arus bolak-balik (AC) menjadi sumber arus searah (DC).



Gambar 1. Blok diagram sistem.

*Flyback converter* berfungsi sebagai *driver* LED dengan menggunakan metode *feed-forward backpropagation neural network* (FF-BPNN) untuk optimasi tegangan sebagai suplai LED. Pada umumnya rangkaian *driver* HPL yang digunakan saat ini telah ditentukan berapa besar *rating* arus dan tegangannya, sehingga pengguna *high power* LED harus selalu memastikan apakah *rating* arus dan tegangan *driver* yang akan digunakan sudah sesuai dengan beban LED-nya. Untuk memudahkan pengguna dalam pemilihan *driver* serta efisiensi agar tidak perlu mengganti *driver* LED apabila menggunakan LED dengan *rating* daya yang berbeda-beda, maka diperlukan *driver* LED yang dapat menyesuaikan *output* tegangan serta arusnya secara otomatis. Penelitian ini membahas tentang simulasi *adaptive high power LED driver*, dimana alat ini dapat melakukan pendeteksian tegangan kerja beban secara otomatis menggunakan metode FF-BPNN untuk menjaga LED tetap bekerja pada tegangan yang sesuai dengan *rating* tegangannya.

### 2.2. Perancangan Rectifier



Gambar 2. Rangkaian *rectifier*.

Gambar 2 merupakan rangkaian *rectifier* penyearah gelombang penuh sistem jembatan. Prinsip kerja dari rangkaian ini dapat dijelaskan sebagai berikut.

- Pada saat rangkaian jembatan mendapatkan siklus positif dari sinyal arus bolak-balik (AC), maka

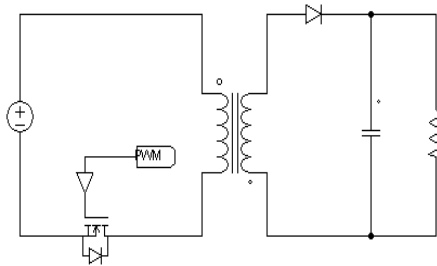
- a. Diode 1 (D1) dan diode 3 (D3) hidup (*on*), karena mendapat bias maju.
  - b. Diode 2 (D2) dan diode 4 (D4) mati (*off*), karena mendapat bias mundur.
- Sehingga arus  $i_2$  mengalir melalui D1, D3 dan RL.
- Apabila jembatan memperoleh siklus negatif, maka
    - a. Diode 2 (D2) dan diode 4 (D4) hidup (*on*), karena mendapat bias maju.
    - b. Diode 1 (D1) dan diode 3 (D3) mati (*off*), karena mendapat bias mundur.

Sehingga arus  $i_2$  mengalir melalui D2, D4 dan RL.

Dengan demikian, arus yang mengalir ke beban ( $i_L$ ) merupakan penjumlahan dari dua arus  $i_1$  dan  $i_2$  [9].

### 2.3. Perancangan Flyback Converter

*Flyback converter* merupakan konverter yang dapat bekerja berdasarkan prinsip kerja dari *buckboost converter* yang terdiri dari transformator termasuk induktor magnetisasi ( $L_m$ ), sebuah *switch (mosfet)* yang terhubung ke sisi *ground primer* pada transformator. Transformator yang digunakan berbeda dengan transformator ideal yang sering digunakan, karena arus tidak dapat mengalir secara simultan pada sisi sekunder sehingga polaritas transformator saling terbalik [10]. Pada penelitian ini *flyback converter* digunakan yaitu untuk menurunkan tegangan 311 volt menjadi 66 volt dengan frekuensi *switching* 40 KHz.



Gambar 3. Rangkaian ekivalen *flyback converter*.

Berikut adalah persamaan-persamaan yang digunakan untuk mendapatkan parameter-parameter dari *flyback converter* [9].

$$V_o = V_s \times \left( \frac{D}{1-D} \right) \times \left( \frac{N_2}{N_1} \right) \quad (1)$$

$$L_m = \frac{(V_s \times D)^2}{2 \times P_{in} \times f_s \times K_{rf}} \quad (2)$$

$$C = \frac{D}{R \times \left( \frac{\Delta V_o}{V_o} \right) \times f} \quad (3)$$

dengan:

$V_s$  = Tegangan masukan (volt)

$V_o$  = Tegangan keluaran (volt)

$D$  = *Duty cycle* (%)

$\left( \frac{N_2}{N_1} \right)$  = Rasio belitan (*turn ratio*)

$L_m$  = Induktansi transformator ( $\mu$ H)

$f_s$  = *frequency switching* (Hz)

$P_{in}$  = Daya masukan (watt)

$C$  = Kapasitansi kapasitor ( $\mu$ F)

$R$  = Resistansi/hambatan (ohm)

$\Delta V_o$  = *Ripple* (riak) tegangan keluaran (volt)

$K_{rf}$  = Konstanta *ripple factor*

Sehingga pada sistem ini diperoleh parameter *flyback converter* seperti pada Tabel 1.

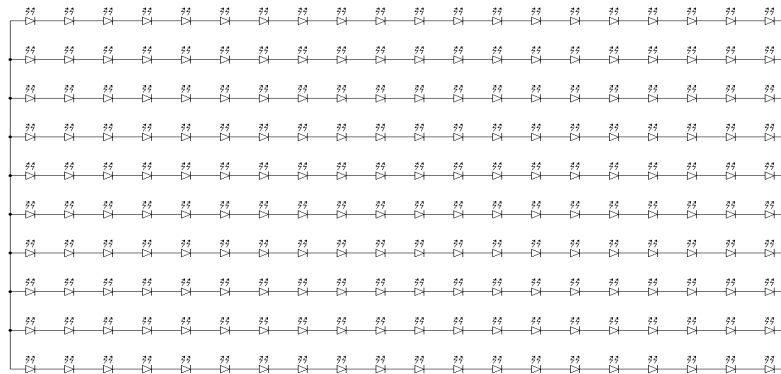
Tabel 1. Parameter *flyback converter*.

Parameter	Nilai
Tegangan masukan ( $V_s$ )	311 Volt
Tegangan keluaran ( $V_o$ )	66 Volt
<i>Frequency switching</i> ( $f_s$ )	40 KHz
Induktor	711.91 $\mu$ H
Kapasitor	330 $\mu$ F

**2.4. Perancangan LED**

Pada penelitian ini beban yang digunakan yaitu jenis *high power* LED dengan tipe LED 5730 yang mempunyai daya sebesar 0.5 watt tiap LED. Dari tipe LED tersebut dibuat LED modul dengan daya LED sebesar 50 watt, 70 watt dan 100 watt yang disusun secara seri dan paralel. Pada LED dengan daya 50 watt, LED disusun secara seri (*string*) sebanyak 10 buah kemudian diparalel (*merge*) sebanyak 10 kali, sehingga LED yang digunakan sebanyak 100 buah. Pada LED dengan daya 70 watt, LED disusun secara seri (*string*) sebanyak 14 buah kemudian diparalel (*merge*) sebanyak 10 kali, sehingga LED yang digunakan sebanyak 140 buah. Sedangkan pada LED dengan daya 100 watt, LED disusun secara seri (*string*) sebanyak 20 buah kemudian diparalel (*merge*) sebanyak 10 kali, sehingga LED yang digunakan sebanyak 200 buah. Untuk mengurangi disipasi pada LED, diberikan jarak 5 mm tiap LED satu dengan LED yang lain.

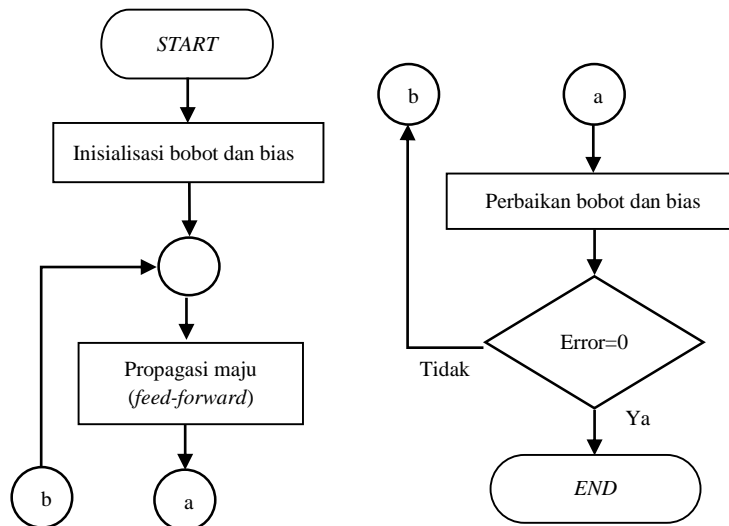
Pada *datasheet* LED 5730 tegangan *forward* dari LED 5730 berkisar mulai dari 3 V - 3.3 V tiap *led*, yang berarti nilai tegangan *forward* apabila LED disusun secara seri dan paralel sebesar 60 V - 66 V untuk LED dengan daya 100 watt. Dengan demikian, *driver* LED harus bisa menyesuaikan dengan tegangan *forward* LED. Desain LED dengan daya 100 watt ditunjukkan pada Gambar 4.



**Gambar 4. Beban *high power* LED.**

**2.5. Pemodelan dan Training FF-BPNN**

Proses *training* menggunakan data pengujian yang dilakukan secara *open-loop* dengan pengambilan data tegangan, arus dan *duty cycle*. Metode ini merupakan milik dari kategori *learning* yang terkontrol, dimana merupakan pendekatan yang sudah berisi data yang terlatih, dan ada variabel yang ditargetkan. Sehingga tujuan dari pendekatan ini adalah untuk mengelompokkan data ke dalam data target yang ada [11, 12].



**Gambar 5. Flow chart learning FF-BPNN.**

Gambar 5 memperlihatkan bahwa proses awal dari *learning* FF-BPNN adalah menginisialisasi nilai bobot dan bias (nilai bobot dan bias sebelum dilakukan *learning*), dimana nilai *input* serta nilai bobot dan bias memasuki tahap propagasi maju (*feed-forward*). Kemudian tahap selanjutnya melalui propagasi mundur (*backpropagation*) untuk melakukan tahap perubahan nilai bobot dan bias. Apabila nilai *error* yang dilakukan 3 tahap tersebut sudah mendekati 0 (nol) dan regresi sama dengan 1 atau iterasi sudah mencapai batas maksimum yang telah ditentukan, maka iterasi akan berhenti dan proses *learning* akan berhenti.

**2.6. Proses Pengambilan Data Learning FF-BPNN**

Pengaplikasian FF-BPNN sebagai *adaptive LED driver* diperlukan data *learning* yang diambil dari simulasi *open-loop* agar didapatkan daya maksimum pada LED. Pengambilan data dilakukan dengan 3 variasi daya pada beban LED dengan daya sebesar 50 watt, 70 watt dan 100 watt. Sinyal PWM yang diberikan ke konverter *flyback* menyesuaikan dengan nilai daya pada beban LED. Data dari variasi daya LED ditunjukkan pada Tabel 2.

Tabel 2. Data simulasi *open-loop*.

Variasi beban (Watt)	V <sub>ac</sub> (volt)	V <sub>in</sub> (volt)	I <sub>in</sub> (amper)	P <sub>in</sub> (watt)	V <sub>o</sub> (volt)	I <sub>o</sub> (amper)	P <sub>o</sub> (watt)	Duty (%)
50	222	311.88	0.262	81.805	33.016	1.51	50.05	29.7
	218	306.19	0.270	82.926	33.005	1.51	50.02	30.09
	214	300.49	0.279	84.067	33.008	1.51	50.02	30.5
	210	294.82	0.288	85.141	33.011	1.51	50.04	30.92
	206	289.10	0.297	86.091	33.007	1.51	50.02	31.35
	202	283.41	0.306	86.959	33.002	1.51	50.01	31.79
	198	277.71	0.316	87.821	33.007	1.51	50.02	32.25
	194	272.01	0.325	88.574	33.007	1.51	50.02	32.72
70	222	311.109	0.322	100.862	46.212	1.52	70.04	37.07
	218	305.412	0.333	101.748	46.202	1.52	70.01	37.51
	214	299.711	0.342	102.648	46.210	1.52	70.03	37.97
	210	294.009	0.351	103.453	46.215	1.52	70.05	38.44
	206	288.307	0.361	104.159	46.217	1.52	70.05	38.92
	202	282.603	0.370	104.762	46.214	1.52	70.05	39.41
	198	276.886	0.380	105.250	46.205	1.52	70.02	39.91
	194	271.260	0.407	110.468	46.201	1.51	70	40.45
100	222	309.911	0.437	135.570	66.002	1.52	100.04	45.73
	218	304.175	0.448	136.439	66	1.52	100	46.21
	214	298.433	0.460	137.328	66.021	1.52	100.06	46.71
	210	292.693	0.471	137.994	66.015	1.52	100.05	47.21
	206	286.95	0.482	138.550	66.007	1.52	100.02	47.72
	202	281.202	0.494	139.106	66.019	1.52	100.06	48.25
	198	275.455	0.506	139.429	66.002	1.51	100.01	48.78
	194	269.701	0.518	139.736	66.003	1.51	100.01	49.3

Keterangan:

V<sub>ac</sub> = Tegangan jala-jala PLNV<sub>in</sub> = Tegangan *input flyback converter*I<sub>in</sub> = Arus *input flyback converter*P<sub>in</sub> = Daya *input flyback converter*V<sub>o</sub> = Tegangan *output flyback converter*I<sub>o</sub> = Arus *output flyback converter*P<sub>o</sub> = Daya *output flyback converter*

Data pada Tabel 2 dapat dilihat bahwa nilai yang akan digunakan sebagai parameter *input* FF-BPNN yaitu nilai daya *input flyback converter* dan nilai *output* FF-BPNN yaitu *duty cycle*. Berikut data *learning* FF-BPNN yang ditunjukkan pada Tabel 3.

Tabel 3. Data *learning* FF-BPNN.

Nomor	Data input (daya (watt))	Data target (duty (%))	Nomor	Data input (daya (watt))	Data target (duty (%))
1	81.805	29.7	13	104.159	38.92
2	82.926	30.09	14	104.762	39.41
3	84.067	30.5	15	105.250	39.91
4	85.141	30.92	16	110.468	40.45
5	86.091	31.35	17	135.570	45.73
6	86.959	31.79	18	136.439	46.21
7	87.821	32.25	19	137.328	46.71
8	88.574	32.72	20	137.994	47.21
9	100.862	37.07	21	138.550	47.72
10	101.748	37.51	22	139.106	48.25
11	102.648	37.97	23	139.429	48.78
12	103.453	38.44	24	139.736	49.3

Arsitektur *artificial neural network* (ANN) terdiri dari dua buah layer, yaitu satu buah layer tersembunyi dan satu buah layer *output*. Berikut perhitungan jumlah neuron yang digunakan.

$$\text{Input layer (in)} = 1 \text{ (Daya input)}$$

$$\text{Output layer (on)} = 1 \text{ (Duty cycle)}$$

$$\text{Hidden layer (hn)} = \sqrt{\text{in} + \text{on} + \sigma}$$

(4)

$$\text{Hidden layer (hn)} = \sqrt{\text{in} + \text{on} + 10} = 11,44$$

Keterangan :

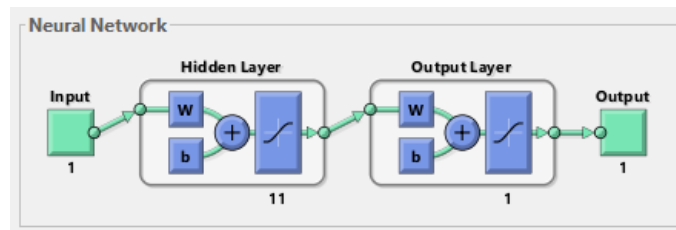
hn = jumlah *neuron* pada *hidden layer*

in = jumlah *neuron* pada *input layer*

on = jumlah *neuron* pada *output layer*

$\sigma$  = konstanta dengan kisaran 1 hingga 10

Dari hasil perhitungan sesuai dengan persamaan (4) maka jumlah layer tersembunyi terdapat 11 buah *neuron* dengan fungsi aktivasi tansig dan pada layer *output* terdapat 1 buah *neuron*.



Gambar 6. Desain arsitektur *perceptron neural network*.

Gambar 6 merupakan arsitektur *perceptron* yang digunakan dalam penelitian ini, dengan 3 jenis variasi beban LED. *Neural network* yang digunakan, memiliki spesifikasi sebagai berikut:

1. *Input ANN* = 1 *input*
2. Jumlah layer = 1 *hidden Layer*
3. Neuron pada *hidden layer* = 11 neuron
4. *Output ANN* = 1 *output*

Dari empat bagian tersebut dapat dilakukan analisis perhitungan secara matematis dalam setiap bagiannya. Secara lebih detail dijelaskan sebagai berikut:

#### 1. Proses *input* 1

Pada proses *input* yaitu normalisasi, artinya data *input* diubah ke dalam bentuk bilangan standar ANN dengan *range* data -1 sampai dengan 1. Blok fungsi  $\text{Map}_{\text{minmax}}$  berfungsi untuk menstandarkan nilai *input* data menjadi bilangan standar ANN (persamaan (5)). Variabel  $x_{\text{min}}$  dan  $x_{\text{max}}$  merupakan nilai minimal dan nilai maksimal dari data *input*, sedangkan  $y_{\text{max}}$  bernilai 1 dan  $y_{\text{min}}$  bernilai -1.

$$\text{Map}_{\text{minmax}} = \frac{(x - x_{\text{min}})(y_{\text{max}} - y_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})} + y_{\text{min}} \quad (5)$$

#### 2. Layer 1 (*hidden layer*)

Pada *hidden layer* terdapat blok fungsi utama yaitu *weight* (W), bias (B), netsum dan fungsi aktivasi tansig. Perhitungan dilakukan dengan perkalian *input* dengan bobot (*weight*) dan ditambahkan dengan biasnya persamaan (6). Setelah itu diaktivasi oleh fungsi aktivasinya.

$$\text{Layer1}(n) = (X_1 \times W_1) + \dots + (X_9 \times W_9) + B_n \quad (6)$$

Selanjutnya, hasil perhitungan dari persamaan (6) dilanjutkan dengan perhitungan rumus fungsi aktivasi logsig. Fungsi logsig (log-sigmoid) adalah fungsi transfer yang membawa *input* ke *output* dengan penghitungan log-sigmoid. Persamaan (7) merupakan perhitungan matematis dari fungsi aktivasi logsig yang digunakan untuk menyederhanakan hasil perhitungan menjadi angka dengan *range* (nilai *output*) 0 hingga 1.

$$\text{log sig}(n) = \frac{1}{1 + \exp(-n)} \quad (7)$$

#### 3. Layer 2 (*output layer*)

Pada *output layer* terdapat blok fungsi utama yaitu *weight* (W), bias (B), netsum, dan fungsi aktivasi tansig. Perhitungan dilakukan dengan perkalian *input* dengan bobot (*weight*) dan ditambahkan dengan biasnya (persamaan (8)). *Input* dalam proses merupakan data hasil perhitungan dari layer 1. Setelah itu diaktivasi oleh fungsi aktivasinya.

$$\text{Layer2}(n) = (X_1 \times W_1) + \dots + (X_9 \times W_9) + B_n \quad (8)$$

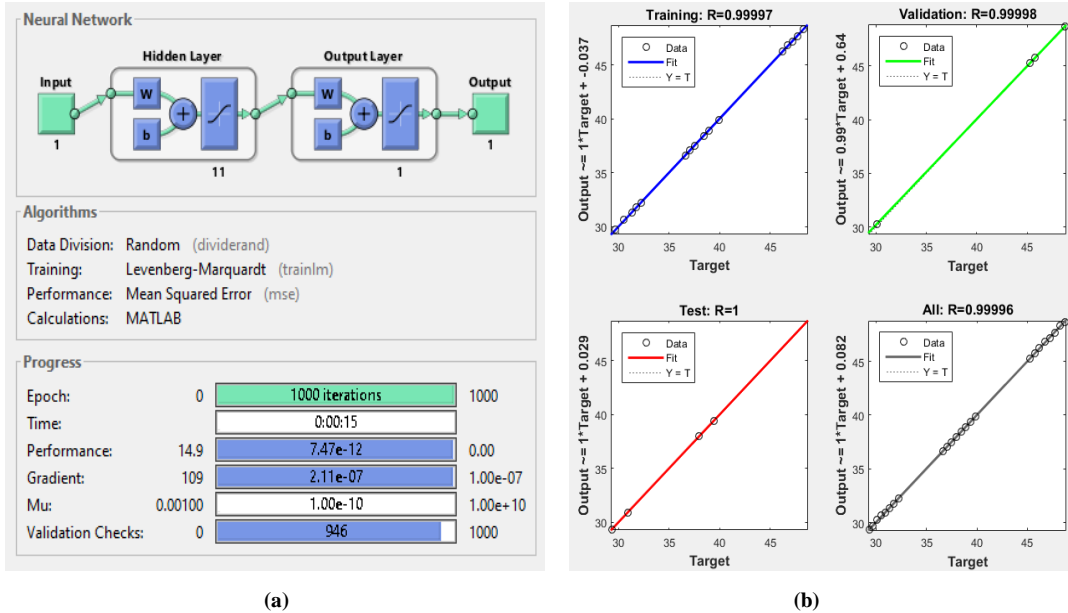
Selanjutnya, hasil perhitungan dari persamaan (8) dilanjutkan dengan perhitungan rumus fungsi aktivasi tansig [13]. Tansig adalah fungsi sigmoid tangen yang digunakan sebagai fungsi aktivasi. Fungsi ini akan membawa nilai *input* pada *output* dengan menggunakan rumus hiperbolik tangen sigmoid. Persamaan (9) merupakan perhitungan matematis dari fungsi aktivasi tansig yang digunakan untuk menyederhanakan hasil perhitungan menjadi angka dengan *range* (nilai *output*) -1 hingga 1.

$$\text{tansig}(n) = \frac{2}{(1 + \exp(-2n))} - 1. \tag{9}$$

4. Proses *output* 1

Proses *output* merupakan proses terakhir dalam perhitungan jaringan arsitektur ANN, yakni blok fungsi  $\text{Map}_{\min\max}$ . Perbedaan yang mendasar terletak pada variabel yang digunakan dibalik, yang sebelumnya  $x_{\min}$  menjadi  $y_{\min}$ . Sesuai dengan nama blok fungsinya yakni  $\text{Map}_{\min\max}$  *reverse*, yang berarti membalik dari bilangan standar ANN menjadi bilangan asli atau data *output* asli (logika 1 atau 0) dengan cara perhitungan matematis. Persamaan (10) adalah perhitungan terakhir dari serangkaian proses pada jaringan ANN yang merubah bilangan standar ANN menjadi bilangan target yang sebenarnya.

$$\text{Map}_{\min\max}\text{Denorm} = \frac{(y - y_{\min})(x_{\max} - x_{\min})}{(y_{\max} - y_{\min})} + x_{\min}. \tag{10}$$



Gambar 7. - (a) Proses pembelajaran FF-BPNN; (b) Plot regresi FF-BPNN

Gambar 7(a) merupakan proses *learning* FF-BPNN, dimana proses *learning* ini berlangsung selama 1000 iterasi dengan menggunakan metode *training* Levenberg-Marquadt [11-12]. Sedangkan Gambar 7(b) merupakan hasil dari *learning* yang telah dilakukan. Dapat dilihat bahwa nilai plot mendekati 1 yang berarti FF-BPNN yang dibuat cukup identik dengan data yang telah dilakukan *learning*, sehingga FF-BPNN ini layak digunakan sebagai kontrol driver LED. Dari hasil tersebut terlihat bahwa hasil *learning* untuk data ini sudah bagus dengan nilai sebesar 0.99996. Dari proses *learning* FF-BPNN didapatkan nilai bobot dan bias yang ditunjukkan pada Tabel 4.

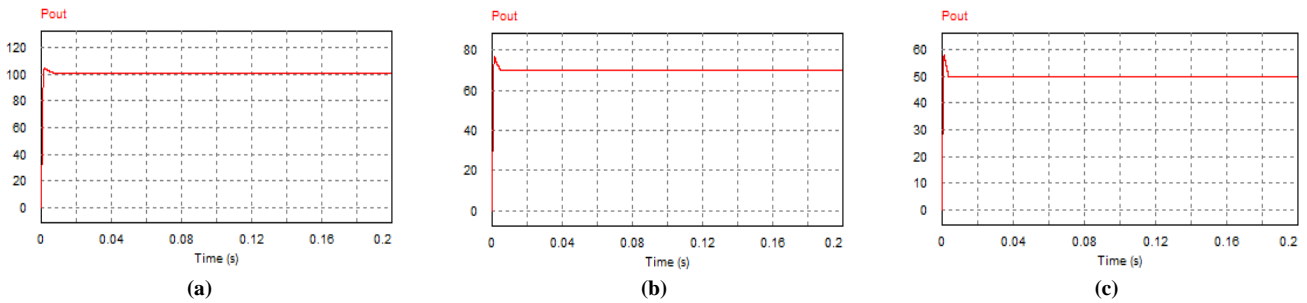
Tabel 4. Nilai bobot dan bias dari hasil *learning* FF-BPNN

Neuron	Layer tesembunyi		Neuron	Output layer	
	Bobot layer	Bias layer		Bobot layer	Bias layer
	Input	Input		Output	Output
1	30.4102	31.8401		11.8224	
2	14.6745	-13.2614		0.28248	
3	-15.397	9.2439		0.4704	
4	15.4	-6.16		-0.12836	
5	15.4007	-3.0853		-0.91299	
6	-15.3221	-0.6122	1	-1.8399	4.6757
7	-14.0116	-2.9069		-0.082543	
8	13.4558	4.1857		0.089199	
9	15.484	9.0874		0.22733	
10	15.6642	12.4754		0.26169	
11	-16.2097	-17.0012		-7.1344	

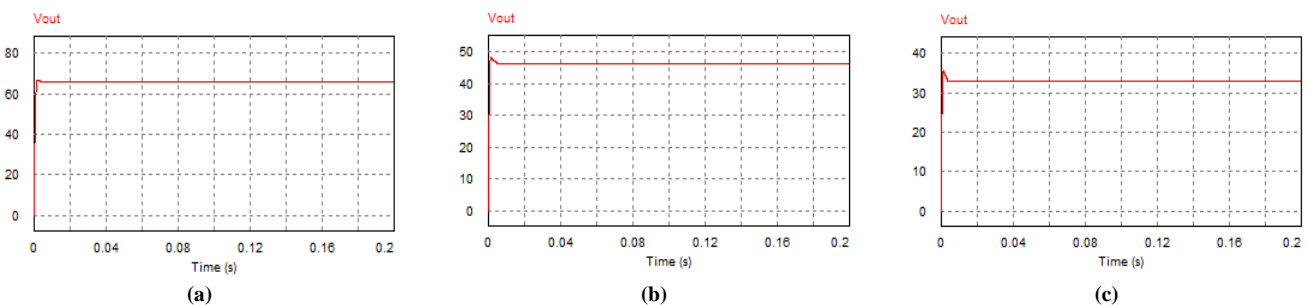
Bobot dan bias hasil *learning* ini kemudian diekspor ke dalam program sebagai inisialisasi bobot dan bias untuk proses propagasi maju algoritma ANN. Nilai bobot dan bias ini harus dipastikan urutan matriksnya agar tidak terjadi kesalahan pada pengolahan data yang diberikan.

### 3. Hasil dan Pembahasan

Simulasi *close-loop* dilakukan dengan integrasi sistem secara simulasi yang melibatkan seluruh elemen system, dimana pada penelitian ini dimulai dari *control*, konverter, hingga beban pada sistem. Pengujian dilakukan dengan tegangan jala-jala PLN pada saat 194 volt menggunakan 3 variasi daya LED yaitu sebesar 100 watt, 70 watt dan 50 watt. Hasil pengujian daya dan tegangan *output driver* LED menggunakan variasi 100 watt, 70 watt dan 50 watt ditunjukkan berturut-turut pada Gambar 8 dan Gambar 9. Sedangkan data hasil simulasi ditunjukkan pada Tabel 5.



Gambar 8. - (a) Daya output driver LED 100 watt; (b) Daya output driver LED 70 watt; (c) Daya output driver LED 50 watt.



Gambar 9. - (a) Tegangan output driver LED 100 watt; (b) Tegangan output driver LED 70 watt; (c) Tegangan output driver LED 50 watt.

Tabel 5. Hasil simulasi.

V <sub>ac</sub> (volt)	V <sub>in</sub> (volt)	I <sub>in</sub> (amper)	Daya (watt)	Duty cycle (%)
194	272.01	0.33302	49.89	32.71
194	271.26	0.42433	69.94	40.47
194	269.70	0.55769	99.42	49.32

Keterangan:

V<sub>ac</sub> = Tegangan jala-jala PLN

I<sub>in</sub> = Arus input flyback converter

V<sub>in</sub> = Tegangan input flyback converter

Daya = Daya output flyback converter

Data Tabel 5 menunjukkan bahwa *driver* LED dapat menyesuaikan apabila nilai daya LED berubah. Apabila menggunakan beban LED dengan daya 100 watt maka *driver* akan menyesuaikan dengan *rating* LED yaitu sebesar 99.42 watt dengan *duty cycle* sebesar 49.32%. Ketika daya LED sebesar 70 watt, *driver* LED dapat menyesuaikan dengan daya 69.94 watt dengan *duty cycle* sebesar 40.47%. Sedangkan ketika daya LED sebesar 50 watt, *driver* LED dapat menyesuaikan dengan daya 49.89 watt dengan *duty cycle* sebesar 32.71%. Contoh perhitungan FF-BPNN berdasarkan persamaan (5) sebagai adaptif LED *driver* pada Tabel 5 nomor 1 sebagai berikut:

$$\text{Map}_{\min\max} V_{in} (\text{normalisasi}) = \frac{(272.01 - 263.94)(1 - (-1))}{317.57 - 263.94} + (-1) = -0.699.$$

Selanjutnya, perhitungan dengan menggunakan persamaan (6) pada layer 1 dengan bobot ke-1 (W<sub>1</sub>) dan bias ke-1 (B<sub>1</sub>) sebagai berikut:

$$\begin{aligned} \text{Layer1 (I)} &= (\text{Map}_{\min\max} V \times W_1) + B_1 \\ &= (-0.699 \times (-85.0297)) + 83.6432 = 143.078. \end{aligned}$$

Layer 1 terdapat 11 neuron yang dihitung berdasarkan persamaan (5) dengan nilai bobot dan bias yang berbeda-beda sesuai hasil *learning* FF-BPNN. Setelah menghitung semua neuron pada layer 1, selanjutnya menghitung aktivasi sesuai persamaan (7) dengan n adalah hasil perhitungan pada layer 1. Adapun perhitungannya adalah sebagai berikut:

$$\begin{aligned} \log \text{sig}(n) &= \frac{1}{1 + \exp(-n)} \\ &= \frac{1}{1 + \exp(-143.078)} = 1 \end{aligned}$$

Selanjutnya dilakukan perhitungan pada layer 2 dengan menggunakan persamaan (5) dan perhitungan seperti pada perhitungan layer 1. Fungsi aktivasi pada layer 2 berbeda dengan fungsi aktivasi layer 1. Fungsi aktivasi layer 2 menggunakan aktivasi tansig pada persamaan (9) dengan n adalah hasil perhitungan pada layer 2. Adapun perhitungannya adalah sebagai berikut:



$$\begin{aligned}\tan \operatorname{sig}(n) &= \frac{2}{(1 + \exp(-2n))} - 1 \\ &= \frac{2}{(1 + \exp(-2 \times (-0.81086478)))} - 1 = -0.670067.\end{aligned}$$

Setelah didapatkan nilai aktivasi pada layer 2, kemudian dilakukan denormalisasi sesuai rumus pada persamaan (10). Adapun perhitungannya sebagai berikut:

$$\begin{aligned}\operatorname{Map}_{\min \max} \operatorname{Denorm} &= \frac{(y - y_{\min})(x_{\max} - x_{\min})}{(y_{\max} - y_{\min})} + x_{\min} \\ &= \frac{(0.670067 - (-1))(49.9 - 29.31)}{(1 - (-1))} + 29.31 = 32.7066.\end{aligned}$$

Berdasarkan perhitungan dari persamaan (10), maka nilai *duty cycle* yang didapatkan sebesar 32.7066% dan nilai secara simulasi *close-loop* sebesar 32.71%. Dengan demikian, nilai antara perhitungan secara matematik dan secara simulasi *close-loop* hasilnya mendekati dengan *error* sebesar 0.028%. Pada hasil simulasi secara *open-loop* dengan simulasi secara *close-loop* memiliki *error* sebesar 0.059%. Dengan demikian, bahwasanya dengan menggunakan sistem *adaptive LED driver* dapat menentukan daya LED yang optimal berdasarkan *pulse with modulation* (PWM) dengan cara melakukan sistem *tracking* tegangan sesuai dengan data *learning* FF-BPNN untuk berbagai LED secara seri atau paralel, sehingga tidak terjadi kenaikan tegangan yang lebih tinggi dari *rating* LED. Apabila tegangan lebih tinggi dari yang dibutuhkan LED, *driver* akan menghilangkan lebih banyak daya dan menjadi kurang efisien bahkan dapat memperpendek masa pakai LED.

#### 4. Kesimpulan

Penelitian ini penggunaan *flyback converter* sebagai *adaptive LED driver* dengan metode *feed-forward backpropagation neural network* bekerja dengan baik dan sesuai perancangan. Daya LED yang digunakan pada penelitian ini yaitu sebesar 50 watt, 70 watt dan 100 watt. Berdasarkan hasil simulasi *open-loop* dan hasil simulasi *close-loop* pada beban LED 50 watt didapatkan nilai dari PWM mendekati dengan *error* sebesar 0.059%. *Driver* mampu menyesuaikan untuk menyuplai daya LED sebesar 49.89 watt, 69.94 watt dan 99.42 watt, sehingga tidak melebihi dari batas kemampuan LED tersebut. Penggunaan *adaptive LED driver* sangat penting karena *user* tidak perlu mengganti *driver* LED apabila ingin mengganti LED dengan daya yang berbeda. Untuk meningkatkan efisiensi dari *driver* LED ini, dilakukan dengan memperbanyak data dan parameter sebagai data *learning* FF-BPNN.

#### Ucapan terima kasih

Ucapan terimakasih saya kepada Kementerian Riset dan Teknologi - Badan Riset dan Inovasi Nasional untuk mendanai penelitian di Pusat Penelitian Politeknik Elektronika Negeri Surabaya.

#### DAFTAR PUSTAKA

- [1] Suhardi, D. (2014). *Prototipe controller* lampu penerangan LED (*light emitting diode*) *independent* bertenaga surya. *Jurnal Gamma*, no. 10, vol. 1, pp. 116-122.
- [2] Bakin, N. N., Tuyev, V. I., & Yauk, E. F. (2011). LED lighting. *2011 International Conference and Seminar on Micro/Nanotechnologies and Electron Devices Proceedings, Erlagol, Altai, IEEE proceeding*, pp. 346-348. doi: 10.1109/EDM.2011.6006944.
- [3] Yudong, Y., & Liqun, H. (2012). Design of high power LED driver power supply. *2012 Fifth International Conference on Information and Computing Science, Liverpool, IEEE proceeding*, vol. 1, pp. 99-102. doi: 10.1109/ICIC.2012.19.
- [4] Lun, W. K., Loo, K. H., Tan, S. C., Lai, Y. M., & Tse, C. K. (2009). Bilevel current driving technique for LEDs. *IEEE Trans. Power Electron*, vol. 24, no. 12, pp. 2920-2932. doi: 10.1109/TPEL.2009.2021687.
- [5] Ng, S. K., Loo, K. H., Ip, S. K., Lai, Y. M., Tse, C. K., & Mok, K. T. (2012). Sequential variable bilevel driving approach suitable for use in high-color-precision LED display panels. *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4637-4645. doi: 10.1109/TIE.2011.2180490.
- [6] Hsia, S., Sheu, M., & Lai, S. (2015). Chip implementation of high-efficient light-emitting diode dimming driver for high-power light-emitting diode lighting system. *IET Power Electronics*, vol. 8, no. 6, pp. 1043-1051. doi: 10.1049/iet-pel.2013.0805.
- [7] Hsia, S., Sheu, M., Hsiao, P., & Chen, K. (2017). Auto tracking DC/DC converter for adaptive LED driving system. *2017 IEEE Wireless Power Transfer Conference (WPTC)*, Taipei, pp. 1-4. doi: 10.1109/WPT.2017.7953807.
- [8] Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of Artificial Neural Networks*. Cambridge: MIT Press.
- [9] Hart, D. W. (2002). *Power Electronic*. New York: McGraw - Hill Companies, Inc.
- [10] Salam, Z. (2003). Power electronics and Drives (version 3). UTMJB. <https://documents.pub/document/power-electronics-and-drives-version-3-2003-dr-zainal-salam-utmjb-1-dc.html>. Accessed on 15th June 2020.
- [11] Anggriawan, D. O., Satriawan, A. L., Sudiharto, I., Wahjono, E., Prasetyono, E., & Tjahjono, A. (2018). Levenberg Marquardt backpropagation neural network for harmonic detection. *International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, Bali, pp. 129-132. doi: 10.1109/ELECSYM.2018.8615531.
- [12] Tjahjono, A., Anggriawan, D. O., Priyadi, A., Pujiantara, M., & Purnomo, M. H. (2015). Digital overcurrent relay with conventional curve modeling using Levenberg-Marquardt backpropagation. *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, pp. 171-176. doi: 10.1109/ISITIA.2015.7219974.
- [13] Zhan, X., Wang, W., & Chung, H. S. (2017). Application of artificial neural-network to control the light of multi-color LED system. *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, Cincinnati, OH, pp. 3669-3675. doi: 10.1109/ECCE.2017.8096650.