

PERAMALAN JUMLAH KENDARAAN DI DKI JAKARTA DENGAN JARINGAN BACKPROPAGATION

Alwi Smith¹, Zumrotus Sya'diyah²

¹Fakultas Keguruan dan Ilmu Pendidikan Universitas Pattimura

¹Fakultas Keguruan dan Ilmu Pendidikan Universitas Darussalam Ambon

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Indonesia

e-mail: ¹alwi.smith1960@gmail.com, ²zuma.yakuza@gmail.com

Abstrak

Kemacetan di ibukota DKI Jakarta tidak dapat dihindari, terutama pada titik-titik persimpangan baik di jalan-jalan protokol hingga di jalan lingkungan. Semakin hari, kemacetan di Jakarta semakin parah. Menurut sebuah penelitian, kemacetan tersebut membuat masyarakat Jakarta mengalami kerugian hingga Rp 48 triliun per tahun [1]. Dalam makalah ini akan dibahas mengenai prediksi jumlah kendaraan pada tahun 2017. Prediksi ini akan dilakukan dengan menggunakan jaringan syaraf tiruan, yaitu metode backpropagation. Metode ini digunakan karena keunggulannya dalam learning rate. Learning rate sangat berguna dalam menentukan prediksi dengan eror yang kecil. Prediksi jumlah kendaraan ini akan dilakukan pada kendaraan bermotor, mobil pribadi dan kendaraan umum. Sehingga dari hasil prediksi ini akan dapat ditentukan langkah-langkah yang tepat untuk menekan laju pertumbuhan jumlah kendaraan. Dalam pembahasannya nanti akan digunakan Matlab 2009a.

Kata Kunci : jaringan syaraf tiruan, metode backpropagation..

FORECASTING NUMBER OF VEHICLES IN DKI JAKARTA WITH BACKPROPAGATION NETWORK

Abstract

Congestion in the capital city of Jakarta can not be spared, imagine at the intersection points both in the streets protocol up on the road environment. The more the day, the congestion in Jakarta getting worse. According to a study, the congestion that caused the people of Jakarta to lose up to Rp 48 trillion per year [1]. In this paper will discuss about the prediction of the number of vehicles in 2017. This prediction will be done by using artificial neural network, the method of backpropagation. This method is used because of its superiority in learning. Level of learning is very useful in determining predictions with small errors. Predicted number of these vehicles will be done on motor vehicles, private cars and public transport. From the results of this preview will be determined the appropriate steps to suppress the growth rate of the number of vehicles. In the discussion will be used Matlab 2009a.

Keywords: artificial neural network, backpropagation method.

1. Pendahuluan

Kemacetan lalu lintas Jakarta yang semakin akut ini membuat pemerintah Provinsi DKI Jakarta melakukan berbagai upaya untuk mengatasinya. Namun upaya ini seakan sia-sia jika melihat data terbaru jumlah kendaraan di Ibukota. Oleh karena itu, dibutuhkan prediksi yang akurat untuk mengetahui jumlah kendaraan tiap tahun agar dapat dilakukan tindakan preventif untuk menyesuaikan antara kebutuhan masyarakat atas kendaraan dengan kendaraan umum yang ada. Selain itu, dapat dilakukan juga pembatasan pada kepemilikan kendaraan. Hal ini diharapkan dapat membantu pemerintah dalam melakukan upaya penanganan terhadap kemacetan yang selalu terjadi dan bahkan bertambah parah tiap tahunnya.

Dalam makalah ini akan dibahas mengenai prediksi jumlah kendaraan pada tahun 2017. Prediksi ini akan dilakukan dengan menggunakan jaringan syaraf tiruan, yaitu metode *backpropagation*. Metode ini digunakan karena keunggulannya dalam *learning rate*. *Learning rate* sangat berguna dalam menentukan prediksi dengan eror yang kecil. Prediksi jumlah kendaraan ini akan dilakukan pada kendaraan bermotor, mobil pribadi dan kendaraan umum. Sehingga dari hasil prediksi ini akan dapat ditentukan langkah-langkah yang tepat untuk menekan laju pertumbuhan jumlah kendaraan. Dalam pembahasannya nanti akan digunakan Matlab 2009a.

1.1. Jaringan Saraf Tiruan [2]

Jaringan syaraf tiruan (*artificial neural network*) adalah pemrosesan sistem informasi pada karakteristik tertentu dalam keadaan yang berhubungan dengan jaringan syaraf biologi. Jaringan syaraf tiruan dibangun berdasarkan generalisasi dari model matematika pada manusia atau syaraf biologi, didasarkan pada asumsi:

- 1) Pemrosesan informasi terjadi pada banyak element yang disebut *neuron*.
- 2) Sinyal berjalan diantara neuron yang terkoneksi jaringan.
- 3) Masing masing jaringan koneksi dihubungkan dengan bobot, yang mana didalam jaringan syaraf khusus, melipatgandakan transmisi sinyal.
- 4) Masing masing neuron mempergunakan fungsi aktifasi (biasanya nonlinier) ke jaringan inputnya untuk menentukan sinyal output.

Jaringan syaraf dikarakteristikan oleh :

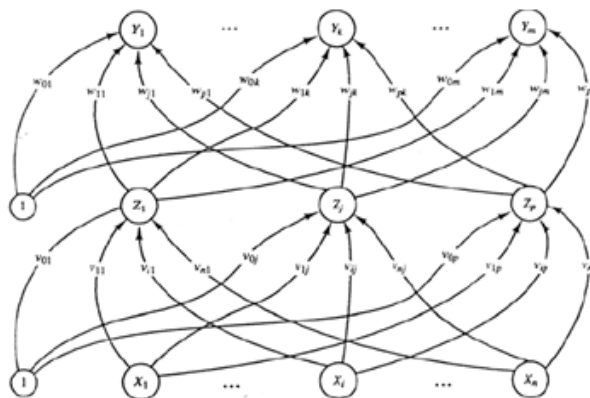
- 1) Polanya terkoneksi diantara *neuron* (disebut arsitektur)
- 2) Metode untuk menentukan bobot dari koneksi (disebut training atau learning, algoritma)
- 3) Fungsi aktifasi.

1.2. Backpropagation [2]

Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama proses training serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola inputan yang serupa (tapi tak sama) dengan pola yang dipakai selama *training*.

1.2.1 Arsitektur

Backpropagation memiliki beberapa unit yang ada di dalam satu atau lebih hidden layer.



Gambar 1. Arsitektur backpropagation

Gambar 1 merupakan arsitektur *backpropagation* dengan n buah inputan (ditambah sebuah bias), sebuah hidden layer yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit output. Dimana v_{ji} merupakan bobot garis dari unit input x_i ke unit hidden layer z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit inputan ke unit *hidden layer* z_j). w_{ji} merupakan bobot dari unit hidden layer z_j ke unit output y_k (w_{k0} merupakan bobot garis yang menghubungkan bias di unit *hidden layer* ke unit output z_k).

1.2.2 Fungsi Aktivasi

Fungsi aktivasi untuk jaringan *backpropagation* memiliki beberapa karakteristik yang penting. Fungsi aktivasi tersebut kontinu, dapat diturunkan dan tidak turun secara monoton. Untuk efisiensi perhitungan, turunan dari fungsi aktivasi tersebut juga mudah dihitung. Fungsi aktivasi yang paling umum digunakan adalah nilai dari turunannya (pada nilai tertentu dari variabel independen) dapat diekspresikan dalam bentuk nilai dari fungsi aktivasi tersebut (pada nilai variabel independen). Fungsi aktivasi yang paling sering digunakan adalah fungsi sigmoid biner. Sedangkan fungsi lain yang juga cukup sering dipakai adalah fungsi sigmoid bipolar. Penjelasan dari kedua fungsi tersebut diberikan sebagai berikut:

1. Fungsi sigmoid biner untuk interval (0,1)

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = f(x)[1 - f_1(x)]$$

2. Fungsi sigmoid bipolar dengan interval (-1,1)

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

$$f'(x) = \frac{1}{2}[1 + f_2(x)][1 - f_2(x)]$$

1.2.3 Algoritma *Backpropagation*

Step 0 : Inisialisasi bobot semua bobot dengan bilangan acak kecil

Bobot bias input (v_{0j}) = bilangan acak dari $-\beta$ dan β , dengan

Bobot input (v_{ij}) = bilangan acak dari -0.5 dan 0.5

Bobot bias hidden (w_{0k}) dan bobot hidden (w_{jk}) = bilangan acak dari -1 dan 1.

Step 1 : Jika kondisi penghentian belum terpenuhi, lakukan Step 2-9

Step 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Step 3 : Tiap unit input ($X_i, i=1, \dots, n$) menerima sinyal dan meneruskannya ke unit hidden

Step 4 : Hitung semua output di unit hidden ($Z_j, j=1, \dots, p$)

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z_in_j)$$

Step 5 : Hitung semua output di unit output ($Y_k, k=1, \dots, m$)

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

$$y_k = f(y_in_k)$$

Step 6 : Hitung δ unit output berdasarkan error di setiap unit output Y_k

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Hitung suku perubahan bobot hidden dan bobot bias *hidden* dengan *learning rate* α

$$\Delta w_{jk} = \alpha \delta_k z_j$$

$$\Delta w_{0k} = \alpha \delta_k$$

Step 7 : Hitung δ unit *hidden* berdasarkan *error* di setiap unit *hidden* Z_j

Hitung suku perubahan bobot input dan bobot bias input dengan *learning rate* α

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$$\Delta v_{0j} = \alpha \delta_j$$

Step 8 : Hitung semua perubahan bobot

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Step 9 : Test kondisi penghentian

2. Hasil dan Pembahasan

2.1. Analisa Data

Sebagaimana telah disebutkan sebelumnya bahwa data yang akan digunakan dalam proses peramalan atau prediksi ini adalah data jumlah kendaraan di Jakarta selama sepuluh tahun terakhir. Data ini akan digunakan untuk mengetahui kisaran yang akurat tentang jumlah kendaraan Data yang digunakan dalam proses prediksi adalah sebagai berikut:

Tabel 1. Data Jumlah Kendaraan Polda Metro Jaya [3]

Tahun	Jenis Kendaraan Bermotor				Jumlah
	Sepeda Motor	Mobil Penumpang	Mobil Beban	Mobil Bis Penumpang	
2007	1.813.136	1.130.496	347.443	253.648	3.544.723
2008	2.257.194	1.195.871	366.221	254.849	4.074.135
2009	3.316.900	1.529.824	464.748	315.652	5.627.124
2010	3.940.700	1.645.306	488.517	316.396	6.390.919
2011	4.647.435	1.766.801	499.581	316.502	7.230.319
2012	5.310.068	1.835.653	504.727	317.050	7.967.498
2013	5.947.173	1.916.469	518.991	318.332	8.727.965
2014	6.765.723	2.034.943	538.731	308.528	9.647.925
2015	7.084.753	2.054.254	507.410	308.941	9.955.358
2016	8.484.384	2.285.802	564.776	322.440	11.657.402

(Sumber: www.komisikepolisianindonesia.com, 2016)

Tampak pada tabel bahwa kenaikan jumlah kendaraan di Jakarta sangat signifikan. Tapi, tiap tahun mempunyai tingkat kenaikan yang berbeda-beda. Hal inilah yang membuat prediksi atas data selanjutnya yang akurat menjadi sulit didapatkan. Oleh karena itu, akan digunakan metode backpropagation untuk meramalkan data berikutnya. Program akan belajar untuk mengetahui model kenaikan jumlah kendaraan yang berubah dan berbeda tiap tahunnya.

2.2. Hasil dan Representasi Prediksi

Pada metode *backpropagation*, langkah yang pertama dilakukan adalah menentukan bobot awal. Pemilihan bobot awal akan mempengaruhi apakah jaringan mencapai minimum global (atau hanya lokal) dari *error*, dan jika itu terjadi, seberapa cepat akan konvergen. *Update* bobot antara dua unit bergantung pada turunan fungsi aktivasi dari unit batas atas dan unit batas bawah. Untuk alasan inilah, sangat penting untuk menghindari pemilihan bobot awal yang akan membuat aktivasi atau turunan dari aktivasi tersebut bernilai nol. Nilai dari bobot awal tidak harus terlalu besar, atau sinyal *input* awal ke tiap unit *hidden* atau *output* akan jatuh pada daerah dimana turunan dari fungsi sigmoid memiliki nilai yang sangat kecil (disebut *area saturation*). Selain itu, apabila bobot awal terlalu kecil, *input* jaringan untuk unit *hidden* atau *output* akan mendekati nol, yang akan menyebabkan proses *learning* yang sangat lambat.

Prosedur umum adalah untuk memberi bobot awal (dan bias) dengan nilai acak antara -0.5 dan 0.5 (atau antara -1 dan 1 atau interval lain yang sesuai). Nilai tersebut bisa negatif maupun positif karena nilai akhir dari bobot setelah *training* mungkin sesuai dengan tanda tersebut.

Pada metode ini, proses training tidak dilakukan pada seluruh data, melainkan hanya sebagian saja (biasanya setengah dari data keseluruhan). Sedangkan sisa data yang lain digunakan dalam proses testing. Dalam training, bobot akan diperbarui di setiap iterasi sedemikian hingga diperoleh bobot yang sesuai dengan model data. Sedangkan proses *testing* akan menentukan seberapa akurat prediksi yang telah dilakukan. Hal inilah yang membuat metode ini unggul dari metode yang lain.

Bobot input yang dipakai adalah data yang bernilai (-0.5, 0.5). Bobot ini akan diinputkan ke program dalam bentuk matriks yang ukurannya sesuai dengan jumlah data yang akan diproses dan jumlah hidden layer yang diinginkan. Untuk jaringan syaraf tiruan dengan lebih dari 1 unit *hidden*, hanya modifikasi kecil dari algoritma *backpropagation* pada bab 2.2.2. Perhitungan dari δ diulang untuk setiap *hidden layer*. Hasil teoritis yang telah banyak ditunjukkan pada menyimpulkan bahwa satu *hidden layer* cukup bagi sebuah jaringan *backpropagation* untuk menetapkan setiap pemetaan kontinu dari pola *input* ke pola *output* pada sebarang derajat ketepatan. Bagaimanapun juga, 2 *hidden layer* mungkin dapat membuat *training* lebih mudah untuk beberapa situasi. Sedangkan bobot awal untuk bobot input diambil nilai $[-\beta, \beta]$, dengan $\beta = 0.7(p)^{1/n} = 0.7\sqrt[n]{p}$. Dan bobot awal untuk bobot hidden diambil nilai (-0.5,0.5).

Dari penjelasan tersebut, maka dalam makalah ini akan digunakan bobot input data yang bernilai (0.5,0.5), jumlah data adalah 10 dan banyak *hidden layer* adalah 1. Oleh karena itu, dapat diambil suatu inisialisasi bobot berupa matriks berukuran 5×1 yang dinyatakan dalam tabel berikut:

Tabel 2. Bobot Input

Bobot input
0.2
0.2
0.2
0.2
0.2

Karena data yang dimiliki bernilai sangat besar, maka perlu dilakukan konversi dengan rumus berikut:

$$x_baru = \frac{x_lama - x_min}{x_maks - x_min}$$

Konversi ini pada akhir peramalan atau prediksi akan dikembalikan lagi dengan invers dari persamaan tersebut sebagai berikut:

$$y_pred = y_lama * (y_maks - y_min) + y_min$$

Sehingga, konversi ini tidak akan mengubah data yang dihasilkan. Dari proses konversi pada data input diperoleh:

Tabel 3. Hasil Konversi data Inputan

Tahun	Jenis Kendaraan Bermotor				Jumlah
	Sepeda Motor	Mobil Penumpang	Mobil Beban	Mobil Bis	
2007	0	0	0	0	0
2008	0.066563	0.056587	0.086402	0.017458	0.065257
2009	0.22541	0.345647	0.539748	0.901326	0.256685
2010	0.318915	0.445605	0.649114	0.912141	0.350833
2011	0.424853	0.550768	0.700023	0.913682	0.454301
2012	0.52418	0.610364	0.7237	0.921648	0.545168
2013	0.61968	0.680316	0.789332	0.940284	0.638906
2014	0.742378	0.782864	0.880161	0.797767	0.752304
2015	0.7902	0.799579	0.723765	0.803771	0.7902
2016	1	1	1	1	1

Data pada Tabel 3 adalah data inputan untuk program. Setelah running, yaitu dengan jumlah hidden layer adalah 1, *learning rate* sebesar 0.5, banyak iterasi adalah 3 (telah mencapai kekonvergenan, artinya bobot tidak terbaru lagi), bobot bias input adalah 0.7, bobot bias *hidden* adalah 0.5, dan bobot *hidden layer*

(berukuran 1x1) sebesar 0.2, diperoleh data hasil *running* program sebagaimana dijelaskan dalam sub bab berikut ini.

2.2.1 Prediksi Jumlah Kendaraan Bermotor

Setelah dilakukan *running* program terhadap data jumlah kendaraan bermotor selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

Tabel 4. Bobot Awal dan Akhir layer

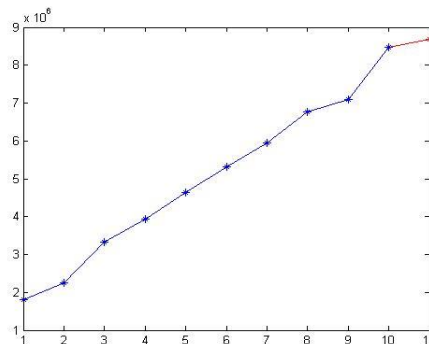
Bobot Awal Input	Bobot akhir Input ($\times 10^5$)
0.7	1.0879
0.2	0.0000
0.2	0.0724
0.2	0.2452
0.2	0.3470
0.2	0.4622

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ($\times 10^7$)
0.5	5.0388
0.2	4.9625

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah kendaraan bermotor pada tahun 2017 adalah sebesar 8.6895×10^6 . Hal ini digambarkan dengan garis merah pada grafik berikut:



Grafik 1. Hasil Prediksi Jumlah Kendaraan Bermotor

2.2.2 Prediksi Jumlah Mobil Penumpang

Setelah dilakukan *running* program terhadap data jumlah mobil penumpang selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

Tabel 5. Bobot Awal dan Akhir layer

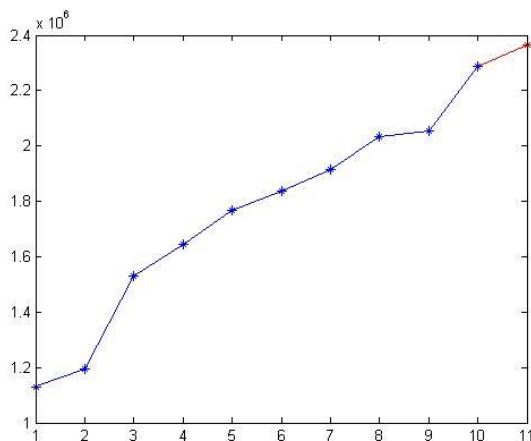
Bobot Awal Input	Bobot akhir Input ($\times 10^4$)
0.7	3.6429
0.2	0.0000
0.2	0.2062
0.2	1.2591
0.2	1.6233
0.2	2.0064

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ($\times 10^7$)
0.5	1.5191
0.2	1.4940

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah mobil penumpang pada tahun 2017 adalah sebesar 2.3665×10^6 . Hal ini digambarkan dengan garis merah pada grafik berikut:



Grafik 2. Hasil Prediksi Jumlah Mobil Penumpang

2.2.3 Prediksi Jumlah Mobil Beban

Setelah dilakukan *running* program terhadap data jumlah mobil beban selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

Tabel 6. Bobot Awal dan Akhir layer

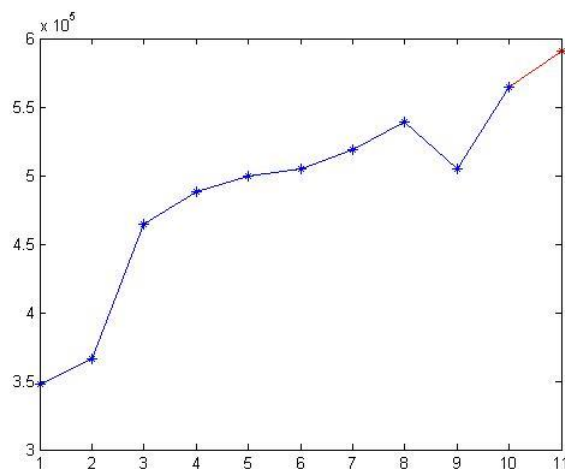
Bobot Awal Input	Bobot akhir Input ($\times 10^3$)
0.7	9.4811
0.2	0.0002
0.2	0.8193
0.2	5.1172
0.2	6.1541
0.2	6.6367

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ($\times 10^6$)
0.5	3.9479
0.2	3.8847

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah mobil beban pada tahun 2017 adalah sebesar 5.9050×10^5 . Hal ini digambarkan dengan garis merah pada grafik berikut:



Grafik 3. Hasil Prediksi Jumlah Mobil Beban

2.2.4 Prediksi Jumlah Mobil Bis

Setelah dilakukan running program terhadap data jumlah mobil bis selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

Tabel 7. Bobot Awal dan Akhir layer

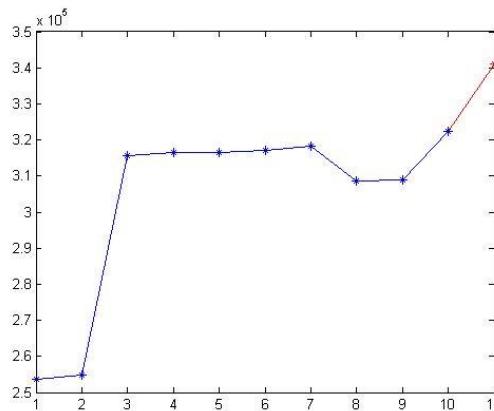
Bobot Awal Input	Bobot akhir Input ($\times 10^3$)
0.7	5.4923
0.2	0.0002
0.2	0.0961
0.2	4.9499
0.2	5.0093
0.2	5.0177

a) Bobot Awal dan Akhir layer input

Bobot Awal Hidden	Bobot Akhir Hidden ($\times 10^6$)
0.5	2.3629
0.2	2.3276

b) Bobot Awal dan Akhir hiddenlayer

Sehingga diperoleh data prediksi jumlah mobil beban pada tahun 2017 adalah sebesar 3.4102×10^5 . Hal ini digambarkan dengan garis merah pada grafik berikut:



Grafik 4. Hasil Prediksi Jumlah Mobil Bis

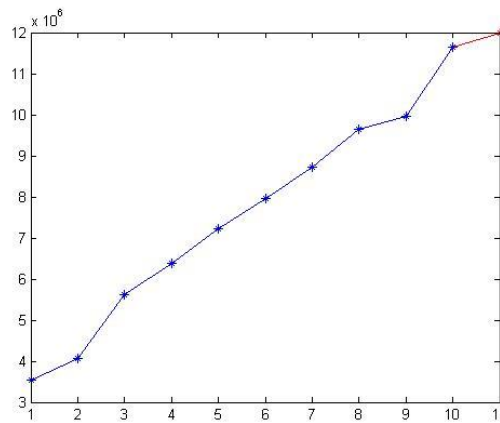
2.2.5 Prediksi Jumlah Total Seluruh Kendaraan

Setelah dilakukan running program terhadap data jumlah total seluruh kendaraan selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

Tabel 8. Prediksi jumlah Total kendaraan tahun 2017

	Prediksi kenaikan jumlah tiap jenis kendaraan
Sepeda Motor	8.689.500
Mobil Penumpang	2.366.500
Mobil Beban	590.500
Mobil Bis	341.020
Jumlah Total	11.987.520

Sehingga diperoleh data prediksi jumlah total seluruh kendaraan pada tahun 2017 adalah sebesar 11.987.520. Hal ini digambarkan dengan garis merah pada grafik berikut:



Grafik 5. Hasil Prediksi Jumlah Total Kendaraan

3. Kesimpulan

Dari penggunaan program menggunakan metode *backpropagation*, diperoleh hasil prediksi sebagai berikut:

	Prediksi kenaikan jumlah tiap jenis kendaraan
Sepeda Motor	8.689.500
Mobil Penumpang	2.366.500
Mobil Beban	590.500
Mobil Bis	341.020
Jumlah Total	11.987.520

Tabel menunjukkan bahwa jumlah sepeda motor naik sebanyak 205.116 unit dari data september 2016, mobil penumpang naik 80.698 unit, mobil beban naik sebanyak 25.724 unit, mobil bis naik 18.580 unit, sedangkan total jumlah kendaraan akan mengalami kenaikan pula sebesar 330.118 unit.

Kenaikan ini harus diatasi dengan pengurangan jumlah kendaraan. Hal ini dapat dilakukan dengan mengadakan uji kelayakan kendaraan dimana kendaraan yang sudah tak layak operasi harus segera diberhentikan ijin beroperasinya. Selain itu, pengurangan tingkat kenaikan jumlah kendaraan dapat dilakukan juga dengan mengurangi pembelian kendaraan dengan cara kredit. Hal ini dikarenakan oleh pemberian kredit dengan bunga yang terlalu rendah meningkatkan keinginan dan daya beli masyarakat atas kendaraan bermotor. Dan kasus ini menyumbangkan pengaruh besar bagi meroketnya jumlah kendaraan. Cara lain yang juga dapat dilakukan adalah membatasi kepemilikan kendaraan bagi tiap keluarga. Hal ini disebabkan oleh tingginya tingkat konsumsi masyarakat Indonesia atas barang mewah, terutama mobil.

Daftar Pustaka

1. Fausett, Laurence. *Fundamentals of Neural Network: Architectures, Fundamentals, and Applications*. Newe Jersey : Prentice-Hall, Inc., 1994.
2. Susanto. *www.Detik News.com*. [Online] 26 November 2008. [Dikutip: 23 Juni 2015.] www.Detik News.com//kemacetan_susanto_26NOV2008.
3. Kompolnas. *www.komisikepolisianindonesia.com*. [Online] Juni 2016. www.komisikepolisianindonesia.com/data_kendaraan_2016.