

Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode *Multilayer Perceptron*

Muhammad Irfan*¹, Bakhtiar Alldino Ardi Sumbodo², Ika Candradewi³

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia

^{2,3}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *¹ipan.elins12@gmail.com, ²b.alldino.as@ugm.ac.id, ³ika.candradewi@gmail.com

Abstrak

Perkembangan video sensor dan hardware dapat digunakan dalam pengembangan sistem pemantauan lalu lintas berbasis vision. Karena dapat menyediakan informasi seputar kendaraan yang melintas agar pemantauan dengan memanfaatkan kamera dapat dilakukan secara otomatis. Dibutuhkan sistem pengolahan yang dapat memberikan beberapa informasi terkait kondisi lalu lintas. Salah satu pendekatannya adalah memanfaatkan pengolahan citra digital.

Penelitian ini terdiri dari 2 tahapan pengolahan citra digital yaitu proses deteksi dan klasifikasi. Proses deteksi menggunakan Haar Cascade classifier dengan data training berupa citra kendaraan dan data uji berupa citra keadaan di jalan tol yang diambil secara acak. Sementara, proses klasifikasi menggunakan Multilayer Perceptron dengan memanfaatkan hasil dari deteksi yang diproses. Klasifikasi kendaraannya di bagi menjadi 3 tipe kendaraan yaitu Mobil, Bus, dan Truk. Lalu klasifikasi dievaluasi dengan parameter accuracy.

Hasil pengujian dalam proses deteksi kendaraan menunjukkan accuracy sebesar 92,67%. Sementara itu, proses klasifikasi dilakukan dengan tahap trial and error untuk mengevaluasi parameter yang telah ditentukan. Hasil penelitian menunjukkan sistem klasifikasi memiliki rata-rata nilai accuracy sebesar 87,60%.

Kata kunci— Kendaraan, Deteksi, Haar, Cascade, Klasifikasi, MLP

Abstract

The evolution of video sensors and hardware can be used for developing traffic monitoring system vision based. It can provide information about vehicle passing by utilizing the camera, so that monitoring can be done automatically. It is needed for the processing systems to provide some information regarding traffic conditions. One such approach is to utilize digital image processing.

This research consisted of two phases image processing, namely the process of detection and classification. The process of detection using Haar Cascade Classifier with the training data image form the vehicle and data test form the image state of toll road drawn at random. While, Multilayer Perceptron classification process uses by utilizing the result of the detection process. Vehicle classification is divided into three types, namely car, bus and truck. Then the classification parameters were evaluated by accuracy.

The test results vehicle detection indicate the value of accuracy is 92.67. Meanwhile, the classification process is done with phase trial and error to evaluate the parameters that have been determined. Results of the study show the classification system has an average value of the accuracy is 87.60%.

Keywords— Vehicle, Detection, Haar, Cascade, Classification, MLP

1. PENDAHULUAN

Perkembangan video sensor dan hardware pada pemrosesan video membuka peluang besar bagi pengembangan teknologi sistem pemantauan lalu lintas berbasis vision, karena mampu menyediakan informasi yang lebih detail untuk kebutuhan analisis[1]. Informasi yang didapatkan bisa digunakan untuk memecahkan permasalahan yang biasa kita hadapi di jalan seperti macet atau penumpukan kendaraan. Berbagai cara telah dilakukan, contohnya dengan menggunakan *magnetic loop detector* dan radar sebagai alat pemantauan lalu lintas, akan tetapi alat ini memiliki keterbatasan [2].

Oleh karena itu dibutuhkan sistem yang mampu untuk mengatasi kondisi-kondisi tersebut salah satunya dengan mengembangkan teknologi *Area Traffic Control System (ATCS)*, tujuan dari penerapan ATCS ini adalah untuk mempermudah kerja manusia dalam mengatur lalu lintas pada setiap jalur yang ada dan dapat meminimalisir masalah *human error* dalam pengaturan lalu lintas [3]. ATCS yang disinkronkan dengan kamera yang ada di jalan-jalan dapat membantu pemantauan lalu lintas. Memanfaatkan kamera tersebut secara otomatis agar dapat memberikan beberapa informasi seputar kondisi lalu lintas di jalan [4]. Salah satu pendekatannya adalah memanfaatkan pengolahan citra digital. Pengolahan citra digital adalah ilmu pemrograman *computer* untuk memproses dan akhirnya memahami gambar dan video atau membuat *computer* dapat melihat [5].

Pemantauan lalu lintas yang serba otomatis harus melewati beberapa tahapan, seperti deteksi dan klasifikasi. Deteksi obyek dalam pengolahan citra digital adalah suatu proses yang digunakan untuk menentukan keberadaan obyek tertentu di dalam suatu citra digital [6]. Pendeteksian benda yang dikembangkan saat ini berdasarkan jenis suatu obyek [7]. Proses pendeteksian ini berfungsi untuk membedakan kendaraan dan non kendaraan dalam satu citra, yang hasilnya nanti akan digunakan untuk proses klasifikasi. Penelitian ini bertujuan untuk mengimplementasikan pengolahan citra digital untuk membuat sebuah sistem yang mampu mengklasifikasi kendaraan, berdasarkan jenisnya.

2. METODE PENELITIAN

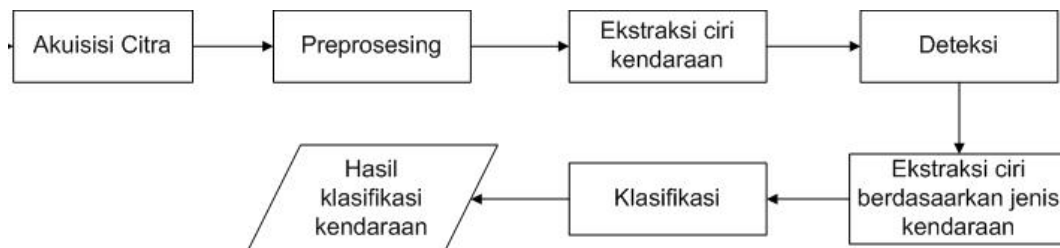
2.1 Analisis Sistem

Rancangan sistem menggunakan perangkat lunak (*software*) dengan pengolahan citra berdasarkan *library OpenCV*. Perancangan sistem ini menggunakan metode *Haar cascade* dalam mendeteksi obyek dan menggunakan *Multilayer Perceptron* untuk klasifikasinya. *Haar Cascade Classifier* ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah *pixel* dalam persegi dengan nilai dari sebuah gambar [8]. Metode ini merupakan metode yang menggunakan *statistical model (classifier)*. Ada beberapa langkah pendekatan untuk mendeteksi obyek, yaitu :

- a. *Training data*.
- b. Fitur segi empat sederhana yang disebut fitur *haar*.
- c. Integral *image* untuk pendeteksian fitur secara tepat.
- d. Pengklasifikasian bertingkat (*Cascade Classifier*).

Lalu dalam proses klasifikasinya, sistem menggunakan *Local Binary Pattern*. *Local Binary Pattern* atau LBP adalah teknik ekstraksi fitur berbasis pada analisis tekstur mentransformasikan sebuah citra kedalam *table statistic integer*, menggunakan komputasi *mean* efisien[9]. Dilanjutkan dengan proses pelatihan *Multilayer Perceptron*, *Multilayer Perceptron* adalah jaringan syaraf tiruan *feed-forward* yang terdiri dari sejumlah neuron yang dihubungkan oleh bobot-bobot penghubung *neuron-neuron* tersebut disusun dalam lapisan-lapisan yang terdiri dari satu lapisan *input (input layer)*, satu atau lebih lapisan tersembunyi (*hidden layer*), dan satu lapisan *output (output layer)* [10]. Lapisan *input* menerima sinyal dari luar, kemudian

melewatkannya ke lapisan tersembunyi pertama yang akan diteruskan sehingga akhirnya mencapai lapisan *output*. Penulis menggunakan metode tersebut dikarenakan ingin mencoba apakah *Haar Cascade* dan *Multilayer Perceptron* menjadi salah satu pilihan metode yang akurasiya baik untuk mendeteksi dan mengklasifikasi. Secara keseluruhan, blok diagram sistem dapat dilihat pada Gambar 1.



Gambar 1 Diagram blok sistem keseluruhan

Pada blok diagram sistem Gambar 1 didapatkan alur sistem yaitu pertama citra kendaraan diambil dari video lalu lintas yang ada di jalan tol, lalu diproses menjadi kumpulan-kumpulan gambar yang digunakan untuk *dataset* dan frame pengujian. Lalu citra akan melewati tahap *pre-processing* berupa perbaikan kualitas citra. Setelah itu citra akan masuk taha ekstraksi ciri untuk membedakan kendaran dan *non* kendaraan dengan menggunakan metode *Haar cascade clasiffier*. Dilanjutkan dengan tahap deteksi, sistem akan melakukan pelatihan dan pengujian untuk mendapatkan hasil deteksi berupa kendaraan. Setelah didapatkan hasil dari proses deteksi citra akan memasuki proses klasifikasi kendaraan, dimana kendaraan akan dibagi menjadi 3 jenis yaitu (mobil, bus, truk), karena sistem ini dirancang untuk mengelompokkan ketiga jenis kendaraan tersebut sesuai jenisnya.

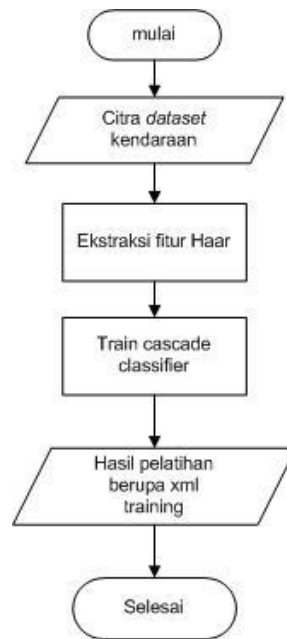
2. 2 Rancangan Arsitektur Sistem

Penulisan ini dirancang dengan usulan penulis menjadi 2 tahap yaitu proses pelatihan (*training*) dan pengujian (*testing*) untuk masing-masing tahap deteksi dan tahap klasifikasi.

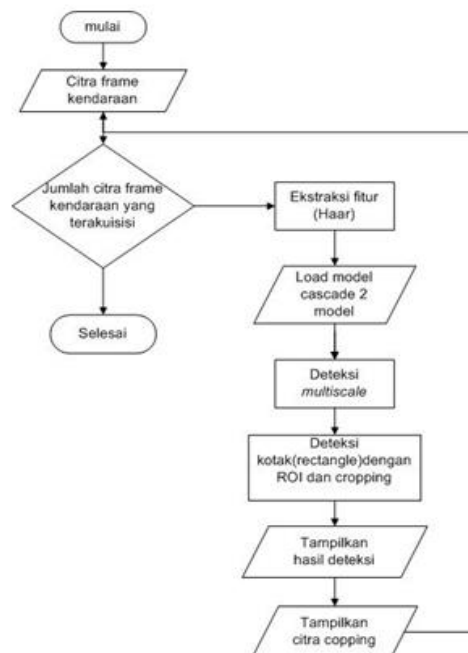
2. 2.1 Rancangan Sistem Deteksi

Proses deteksi menggunakan metode *Haar features* untuk ekstraksi ciri dan *Cascade Classifier* untuk membedakan kendaraan dan yang bukan kendaraan. Sebelum proses deteksi dapat dilakukan, terlebih dahulu dilakukan tahap pelatihan/*training* untuk membentuk pemodelan antara kendaraan dan yang bukan kendaraan. Proses pelatihan ini dimulai dari pengelompokan kendaraan dan yang bukan kendaraan. Jika sudah dikelompokkan maka ada dua kelompok besar, sebut saja positif (kendaraan) dan negatif (bukan kendaraan). Setelah itu proses berlanjut ke proses mendapatkan ekstraksi ciri dari citra positif dan negatif yang kemudian disimpan dalam format file XML. Proses deteksi akan ditunjukkan pada Gambar 2.

Lalu dilanjutkan dengan perancangan sistem selanjutnya, yaitu tahap deteksi pengujian. Pengujian ini dilakukan pada sistem dengan menggunakan citra frame kendaraan yang berada di jalan tol dengan tujuan mendeteksi kendaraan yang ada didalam frame tersebut. Berawal dari proses *pre-processing* yang kemudian citra frame tersebut di prediksi dengan membandingkan pemodelan yang sudah ada. Dengan menggunakan algoritma *detectmultiscale*, sistem akan mulai untuk mendeteksi kendaraan. Jika sistem mengenali obyek tersebut kendaraan, maka obyek akan di *rectangle*. Sistem akan memulai pengenalan pada area yang telah ditentukan. Citra yang terdeteksi, kemudian dicrop dan disimpan kedalam direktori file sebagai hasil dari sistem deteksi tahap pengujian. Tahap pengujian deteksi dapat dilakukan dengan diagram alir seperti pada Gambar 3.



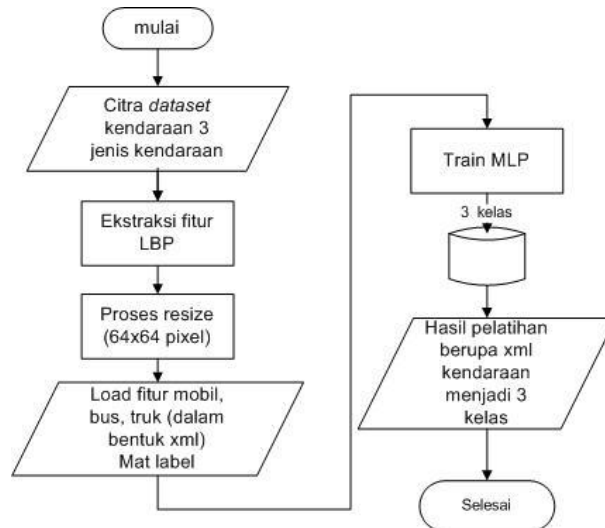
Gambar 2 Diagram alir deteksi tahap pelatihan



Gambar 3 Diagram alir sistem deteksi tahap pengujian

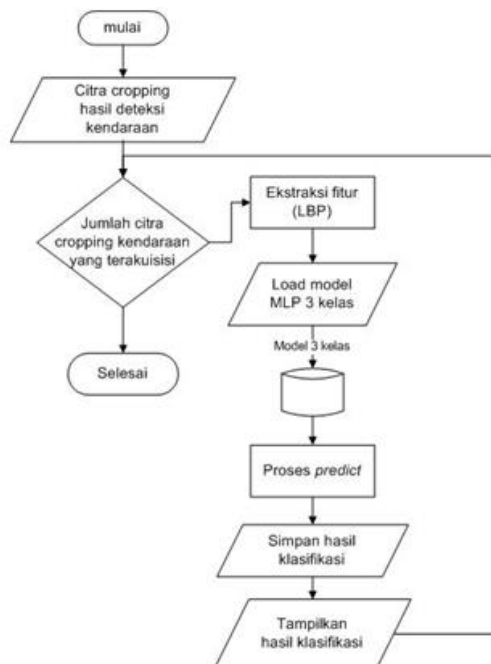
2. 2.2 Rancangan Sistem Klasifikasi

Klasifikasi dalam sistem ini menggunakan metode *Multilayer Perceptron* (MLP). Metode ini akan mengelompokkan tiga jenis kendaraan (mobil, bus, truk). Sebelum proses klasifikasi dapat dilakukan, sistem akan melewati proses ekstraksi ciri citra *dataset* (3 jenis kendaraan) menggunakan fitur *local binary pattern* (LBP) setelah didapatkan tiga ciri tersebut dilanjutkan dengan melabeli dan *training dataset* menggunakan MLP untuk membentuk pemodelan. Tahap pelatihan klasifikasi dapat dilakukan dengan tahapan seperti pada Gambar 4.



Gambar 4 Diagram alir pelatihan klasifikasi

Bentuk pemodelan tahap pelatihan klasifikasi berupa format XML file yang disimpan kedalam direktori untuk digunakan sebagai pembandingan pada tahap pengujian klasifikasi. Proses *training* dan pengujian MLP dilakukan dengan menggunakan citra yang terlebih dahulu dipisahkan, dimana proses pengujian pada MLP dilakukan dengan citra yang sebelumnya tidak dijumpai oleh MLP pada proses *training* untuk melihat tingkat kemampuan sistem klasifikasi. Sistem pengujian memproses citra *cropping* untuk mendapatkan ciri-ciri dari citra dan kemudian dibandingkan dengan pemodelan MLP yang sudah tersimpan dari tahap pelatihan klasifikasi. Setelah itu sistem memprediksi setiap citra *cropping* untuk dikelompokkan ke dalam salah satu kelas kendaraan (mobil, bus, truk). Tahap pengujian menggunakan MLP dapat dilakukan dengan tahapan seperti pada Gambar 5.



Gambar 5 Diagram alir sistem klasifikasi tahap pengujian

3. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari pengujian sistem serta pembahasannya. Secara garis besar dibagi menjadi dua bagian yaitu pengujian pendeteksian dan pengujian klasifikasi. Dalam pengujian deteksi terdapat 1 pengujian, yaitu pengujian *min_neighbours*, sedangkan pengujian klasifikasi terdapat 3 pengujian, meliputi perbandingan data, *hidden layer*, *epsilon*.

3.1 Pengujian pendeteksian kendaraan

Tahapan deteksi ini berfungsi untuk membedakan antara obyek yang dideteksi dengan obyek yang lain misalnya saja jalan, lampu jalan, pohon dll. Pada tahapan deteksi sistem mencoba mendeteksi keberadaan kendaraan tanpa memperhatikan jenisnya. Seperti yang sudah peneliti jelaskan sebelum memasuki sub-bab ini bahwa metode yang digunakan untuk deteksi berbeda dengan metode yang digunakan pada klasifikasi. Pada tahap deteksi peneliti menggunakan *haar cascade classifier*. Data pelatihan terbagi menjadi dua bagian pada setiap sistem yaitu 1667.xml(1667 *positive* dan 5000 *negative*) dan 2500.xml(2500 *positive* dan 5000 *negative*). Contoh frame pengujian yang akan dideteksi akan ditunjukkan pada Gambar 6.



Gambar 6 Frame pengujian deteksi

3.1.1 Pengujian menggunakan parameter *minimal neighbours*

Pengujian pendeteksian kendaraan dengan menggunakan parameter *min_neighbours* untuk mencari nilai parameter *min_neighbours* yang terbaik untuk sistem ini. Pada pengujian dilakukan dengan konfigurasi dimana parameter *min_neighbours* divariasikan nilainya, yaitu 3, 4, dan 5. Sedangkan parameter yang lain tetap dengan nilai *default*. Pengujian dilakukan dengan cara mendeteksi kendaraan dengan variasi nilai *min_neighbours* yang diubah. Setelah didapatkan 3 nilai dengan tingkat keberhasilan tertinggi dengan variasi *min_neighbours* sebagai parameter yang diubah. Hasil Pengujian pendeteksian kendaraan akan ditunjukkan pada Tabel 1.

Tabel 1 Hasil pengujian deteksi dengan parameter *minimal neighbours*

jumlah data minimal neighbours	1667 <i>positive</i> dan 5000 <i>negative</i>	2500 <i>positive</i> dan 5000 <i>negative</i>
3	92.31%	92.67%
4	90.94%	92.28%
5	90.91%	91.95%

Berdasarkan data yang disajikan, dapat dilihat bahwa *min_neighbours* terbaik dari variasi ketiga nilai tersebut adalah 3. Hal ini bisa terjadi karena saat klasifikasi fitur jika persegi lebih dari jumlah persegi (*min_neighbours*) maka kendaraan tersebut terdeteksi. Hal ini juga dibantu dengan adanya *Adaboost machine learning* dengan menggunakan algoritma ini setiap fitur yang terdeteksi bukan kendaraan maka akan diabaikan oleh sistem, sedangkan pada fitur yang masih terdeteksi sebagai fitur kendaraan maka akan dilanjutkan sampai selesai pengecekan dan jika sudah selesai dilakukan pengecekan dan dinyatakan sebagai kendaraan maka akan ditandai dengan *rectangle* (kotak berwarna hijau). Dari hasil pengujian yang memvariasikan

banyaknya *dataset* yang digunakan untuk *training Haar cascade* tidak mempengaruhi tingkat keberhasilan dalam pendeteksian. Hal tersebut dapat dilihat pada Tabel 1 Sehingga digunakan variasi data *training* 2500 *positive* dan 5000 *negative* yang memberikan tingkat *accuracy* yang lebih baik yaitu 92.67%. Data yang dihasilkan dari proses deteksi akan digunakan untuk proses klasifikasi berupa citra *cropping* yang disimpan dalam folder tertentu.

3. 2 Pengujian klasifikasi kendaraan

Tahap klasifikasi ini berfungsi untuk membagi kendaraan menjadi tiga jenis yaitu mobil, bus, dan truk. Sebelum dilakukan proses pengujian. Dilakukan proses pelatihan untuk menentukan parameter *default* yang akan digunakan. Tujuan dari pelatihan jaringan adalah keseimbangan antara kemampuan dan generalisasi. *Multilayer perceptron* diharapkan tidak hanya baik dalam memprediksi dengan masukan set *training*, akan tetapi juga baik dalam memprediksi set testing. Citra-citra testing merupakan hasil *cropping* dari hasil pendeteksian yang dilakukan oleh sistem. Proses *trial and error* mendapatkan parameter *default* diantaranya *Hidden layer*: 50, *Epsilon*: 0.001, Iterasi: 1000, *Bp_scale*: 0.1, *Bp_moment_scale*: 0.1. Contoh frame pengujian yang didapatkan dari hasil *cropping* berupa citra mobil yang ditunjukkan pada Gambar 7.



Gambar 7 Citra *cropping* mobil

Citra tersebut didapatkan saat proses deteksi, dengan menggunakan fungsi *detectmultiscale* sistem akan menandai kendaraan dengan *rectangle* berwarna hijau. Setelah ditandai dengan *rectangle* sistem akan melakukan proses *cropping*. Dengan contoh citra *cropping* bus yang akan ditunjukkan pada Gambar 8.



Gambar 8 Citra *cropping* bus

Citra *cropping* berukuran 64 x 64 pixel. Dikarenakan pelatihan sistem menggunakan citra 64x64 *pixel*, maka sistem akan diuji dengan citra yang sama ukurannya. Contoh citra *cropping* truk yang akan ditunjukkan pada Gambar 9.



Gambar 9 Citra *cropping* truk

3. 2.1 Pengujian parameter perbandingan data

Tahap pengujian pertama ialah mencari perbandingan jumlah data dengan *accuracy* yang paling baik. Perbandingan jumlah data disini dibagi menjadi 3 berdasarkan jumlah data

training, yaitu rendah (jumlah data *training* < 1000), sedang ($1000 < \text{jumlah data } training < 2000$), dan tinggi (jumlah data *training* > 2000). *Balance data* merupakan pembagian kelas yang jumlah tiap jenisnya sama dengan yang lain, sedangkan *imbalance data* merupakan pembagian kelas yang jumlah tiap jenisnya berbeda satu dengan yang lainnya. Sistem pelatihan pertama-tama akan dilatih dengan ke enam data *training* dengan menggunakan parameter-parameter *default* yang sudah ditentukan didapatkan grafik *accuracy* dari ke enam data *training*. Dapat dilihat hasil dari pengujian ini pada Tabel 2.

Tabel 2 Hasil *accuracy* dari parameter perbandingan data

Jenis kendaraan	Mobil	Bus	Truk	Rata-Rata
Jumlah perbandingan data				
perbandingan data 1	78.07%	86.78%	83.00%	82.62%
perbandingan data 2	72.01%	77.20%	84.28%	77.83%
perbandingan data 3	53.76%	63.84%	69.08%	62.23%
perbandingan data 4	67.78%	76.57%	80.26%	74.87%
perbandingan data 5	79.85%	85.60%	85.60%	83.68%
perbandingan data 6	78.15%	86.48%	84.06%	82.90%

Berdasarkan data yang sudah ditampilkan pada Tabel 2, masing-masing mempunyai tingkat *accuracy* berbeda-beda. Hal ini terjadi karena sistem belum mampu mengklasifikasi secara benar. Banyaknya data yang dilatih juga bisa membebani komputasi karena data yang di *training* semakin banyak.

3. 2.2 Pengujian parameter hidden layer

Tahap pengujian kedua ialah menguji sistem dengan merubah parameter jumlah *neuron* pada *hidden layer*, apakah sistem akan merespon dengan baik sehingga didapatkan nilai *accuracy* yang lebih baik dari pengujian sebelumnya. Lapisan tersembunyi atau *hidden layer* berfungsi meningkatkan kemampuan jaringan dalam memecahkan masalah. Dalam pengujian, *hidden layer* yang digunakan hanya berjumlah satu buah dan variasi *neuron* yang digunakan meliputi nilai 10-100. Sedangkan parameter yang lain tetap dalam keadaan *default*.

Berdasarkan data yang sudah ditampilkan pada Tabel 3, jika dilihat dari pengujian sebelumnya sistem mengalami perbaikan nilai *accuracy*. Dalam pengujian kali ini juga didapatkan bahwa jumlah *neuron* untuk proses pelatihan bisa mempengaruhi hasil dari klasifikasi dan semakin banyak jumlah *neuron* yang digunakan akan semakin membebani komputasi, karena konsekuensi dari penggunaan *hidden layer* adalah pelatihan menjadi makin sulit atau lama. Oleh karena itu dibutuhkan proses *trial and error* agar bisa didapatkan hasil yang paling baik.

Tabel 3 Hasil *accuracy* dari parameter jumlah *neuron* pada *hidden layer*

Jenis kendaraan	Mobil	Bus	Truk	Rata-Rata
Jumlah neuron				
10	75.46%	82.07%	85.83%	81.12%
20	80.66%	92.47%	84.35%	85.83%
30	66.42%	78.07%	75.42%	73.30%
40	63.33%	78.80%	71.85%	71.33%
50	79.85%	85.60%	85.60%	83.68%
60	77.86%	85.77%	85.08%	82.90%
70	72.22%	79.92%	82.98%	78.37%
80	71.00%	80.59%	79.92%	77.17%
90	75.46%	82.40%	86.92%	81.59%
100	63.57%	71.55%	78.80%	71.31%

3. 2.3 Pengujian parameter epsilon

Tahap pengujian ketiga ialah menguji sistem dengan merubah parameter epsilon, apakah sistem akan merespon dengan baik sehingga didapatkan nilai *accuracy*. Dengan menggunakan perbandingan data dan *hidden layer* yang sudah ditentukan. Epsilon atau yang

biasa disebut dengan error minimum merupakan salah satu kondisi pemberhentian training. Variasi epsilon yang digunakan meliputi nilai 0.1-0.00001, sistem akan melakukan pelatihan dengan merubah parameter epsilon. Proses penghentian *training* dalam pengenalan pola mempunyai 2 cara. Cara pertama dengan menggunakan *epsilon* dan cara kedua menggunakan *iterasi*. Jika salah satu sudah dianggap melewati batas yang diberikan sistem proses *training* akan berhenti. Jaringan dapat diuji secara terus menerus hingga semua pola pelatihan dapat dikenali. Akan tetapi itu tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga kesalahan atau *error* pelatihan bernilai 0. Sedangkan parameter yang lain tetap dalam keadaan default. Hasil pengujian akan diperlihatkan pada Tabel 4.

Tabel 4 Hasil *accuracy* dari parameter *epsilon*

<i>epsilon</i>	Jenis kendaraan	Mobil	Bus	Truk	Rata-Rata
0.1		80.57%	90.74%	84.16%	85.16%
0.01		80.78%	88.15%	85.82%	84.92%
0.001		82.92%	91.51%	84.69%	86.37%
0.0001		83.20%	93.56%	85.22%	87.33%
0.00001		81.58%	93.56%	83.98%	86.37%

Berdasarkan data yang disajikan dari ketiga pengujian, masing-masing mempunyai nilai *accuracy* rata-rata yang berbeda-beda. Pada pengujian ini juga membuktikan bahwa semakin kecil nilai *error* maksimum atau *epsilon* tidak menjamin sistem akan mengklasifikasi dengan baik contohnya hasil *accuracy* yang paling baik adalah ketika *epsilon* bernilai 0.01. Setelah melihat hasil yang didapatkan, sistem belum mampu mengklasifikasi kendaraan dengan baik. Hal ini terjadi karena sistem belum menemukan pemodelan yang pas untuk mengklasifikasikan kendaraan sesuai jenisnya. Hal ini disebabkan sistem masih banyak melakukan kesalahan klasifikasi jenis kendaraan, yang seharusnya pada data *real* adalah bus tetapi dikenali oleh sistem sebagai mobil.

4. KESIMPULAN

Berdasarkan hasil penelitian pemrosesan citra klasifikasi jenis kendaraan dan tahap pengujian sekaligus pembahasan yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Kombinasi ekstraksi ciri *Local Binary Pattern* dan *Multilayer Perceptron* telah berhasil mengklasifikasikan kendaraan berdasarkan 3 jenis pengelompokkan yaitu mobil, bus, dan truk.
2. Sistem pendeteksian dengan menggunakan metode *Haar cascade classifier* memiliki tingkat *accuracy* yang baik dalam mendeteksi kendaraan dengan variasi terbaik adalah nilai *min_neighbour* 3 dan perbandingan data 2500 *positive* x 5000 *negative*.
3. Penggunaan jumlah data yang akan ditraining pada proses klasifikasi akan sangat mempengaruhi kinerja dari sistem.
4. Pada kasus ini dibutuhkan *trial and error* untuk mendapatkan hasil yang baik pada beberapa parameter, seperti pada parameter *hidden layer* dan *epsilon*
5. Presentase keberhasilan proses klasifikasi jenis kendaraan pada setiap frame, tergantung pada hasil deteksi dan pemilihan fitur yang dapat mewakili tiap jenis kendaraan yang akan digunakan pada proses pengklasifikasian serta jumlah sampel data pada proses pelatihan.

5. SARAN

Beberapa hal yang perlu disempurnakan sehingga perlu dilakukan penelitian lebih lanjut untuk memperbaiki kualitas sistem klasifikasi jenis kendaraan dalam penelitian ini :

1. Pengujian harus memperbanyak proses *trial and error* untuk mendapatkan hasil yang lebih optimal pada proses pelatihan.
2. Menambah pengujian berupa seberapa besar pengaruh intensitas cahaya pada frame citra terhadap hasil deteksi.
3. Menambah variasi data *training* yang lebih banyak dan berbagai parameter agar bisa mendapatkan proses klasifikasi yang lebih akurat.
4. Menggunakan algoritma atau metode yang lain untuk ekstraksi fitur setiap kendaraan sehingga dapat membedakan satu jenis kendaraan dengan jenis kendaraan lainnya secara lebih jelas.
5. Menggunakan perangkat pengolah data (*computer / laptop*) yang mempunyai performa lebih dari perangkat yang digunakan dalam penelitian ini agar proses olah data bisa lebih baik dan lancar.
6. Menerapkan sistem ini kedalam video, karena dalam penelitian ini hanya menggunakan frame citra keadaan jalan tol.

DAFTAR PUSTAKA

- [1] Candradewi, I., 2015, Pemrosesan Video untuk Klasifikasi Kendaraan Berbasis Support Vector Machine, *Tesis*, Pasca Sarjana Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.
- [2] Wang, G., Xiao, D., dan Gu, J., 2008, Review on Vehicle Detection Based on Video for Traffic Surveillance, *Proceedings of the IEEE International Conference on Automation and Logistics*, Qingdao, China, 2961 –2966.
- [3] Syndhuwardhana, F., 2010, Perancangan Pengaturan Sistem Traffic Light dengan Webcam Dinamis, *Skripsi*, Universitas Katolik Soegijarpranata, Semarang.
- [4] Handayani, A.M., 2014, Sistem Penghitung Jumlah Kendaraan Ringan Roda Empat Pada Jalan Raya dengan Metode Haar Cascade Classifier dan Camshift, *Tesis*, Universitas Gadjah Mada, Yogyakarta.
- [5] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. dan Cifrek, M., 2012, A brief introduction to OpenCV, *MIPRO, 2012 Proceedings of the 35th International Convention*, Croatia.
- [6] Nagataries, D., Hardiristanto, S. dan Purnomo, M.H., 2012, Deteksi Obyek pada Citra Digital Menggunakan Algoritma Genetika untuk Studi Kasus Sel Sabit, <http://digilib.its.ac.id/ITS-paper-22021120001182/21993>, diakses tanggal 23 Oktober 2016.
- [7] Latifaf, D.A., Bambang, H. dan Wibowo, T.A., 2011, Klasifikasi Jenis Mobil Menggunakan Metode Backpropagation dan Deteksi Tepi Canny, *Skripsi*, Universitas Telkom, Bandung.
- [8] Viola, P. dan Michael, J.J., 2004, Robust Real-Time Face Detection, *Internasional journal Of Computer Vision*, 57(2), 137-154.
- [9] Trefny, J. dan Matas, J., 2010, Extebded Set of Local Binary Patterns for rapid Object Detection, *Computer Vision Winter Workshop Februari 3-5*, Libor Špacek and Vojtech Fran.
- [10] Puspitaningrum, D., 2006, *Pengantar Jaringan Syaraf Tiruan*, Edisi 1, Yogyakarta, ANDI.