

IMPLEMENTASI SISTEM KLASIFIKASI MOBIL PADA SISTEM PENGATURAN LAMPU LALU LINTAS TERDISTRIBUSI BERBASIS JARINGAN SYARAF TIRUAN

Big Zaman, Muhammad Fajar, M. Eka Suryana, dan Arief Ramadhan

Fakultas Ilmu Komputer, Universitas Indonesia, Kampus Baru UI Depok, Jawa Barat, 16424, Indonesia

E-mail:big.zaman@ui.ac.id

Abstrak

Sistem Pengaturan Lampu Lalu Lintas Terdistribusi adalah sebuah sistem lampu lalu lintas yang ditujukan untuk memenuhi kebutuhan akan kinerja pengaturan lampu lalu lintas yang cerdas dengan pengambilan data secara *real-time*. Sistem ini dapat melakukan penjadwalan dan pengaturan jaringan banyak persimpangan secara *real-time* yang tidak bisa dilakukan oleh sistem pengaturan lampu lalu lintas konvensional. Penerapan klasifikasi di dalam sistem ini digunakan untuk meningkatkan akurasi dari pengenalan mobil. Proses klasifikasi diimplementasikan menggunakan tiga algoritma Jaringan Syaraf Tiruan, yakni *Backpropagation*, FLVQ, dan FLVQ-PSO. Berdasarkan hasil uji coba, dapat ditunjukkan bahwa algoritma *Backpropagation* memiliki performa akurasi yang lebih baik dibandingkan dua algoritma JST yang lainnya.

Kata Kunci: *backpropagation, FLVQ, FLVQ-PSO, jaringan syaraf tiruan, lampu lalu lintas*

Abstract

Distributed Traffic Light Control System is a traffic light system intended to meet the need for setting the performance of intelligent traffic lights with real-time data capturing. The system can perform scheduling and network settings of multi-junction in real time that can not be done by a conventional traffic light settings system. Application of classification within this system is used to improve the accuracy of the car recognition. Classification process is implemented using three neural network algorithms, namely *Backpropagation*, FLVQ, and FLVQ-PSO. Based on the test results, it can be shown that the *Backpropagation* algorithm performs better accuracy than the other two algorithms.

Keywords: *artificial neural networks, backpropagation, FLVQ, FLVQ-PSO, traffic lights*

1. Pendahuluan

Lalu lintas merupakan salah satu elemen vital dalam kehidupan masyarakat. Semua permasalahan di dunia ini akan kembali ke jalan. Ada orang-orang yang pulang bekerja, anak-anak yang pergi bersekolah, dan berbagai aktivitas lainnya, sehingga tidak berlebihan ketika lalu lintas disebut sebagai jantung stabilitas kegiatan rakyat.

Sayangnya, hari ini, lalu lintas kota masih dirundung permasalahan kemacetan yang semakin memprihatinkan. Hal ini dapat diukur berdasarkan data dari Direktorat Samapta, Polri dan Biro Pusat Statistik bahwa pada tahun 2003 populasi kendaraan di Jakarta sudah mencapai 4,159 juta. Sebagian besar populasi kendaraan tersebut merupakan sepeda motor (66.15%) diikuti mobil pribadi (22.01%), kendaraan umum bis (3.57%), dan kendaraan barang atau truk (8.25%).

Selain itu, berdasarkan data dari Komisi Kepolisian Indonesia, jumlah kendaraan bermotor yang terdaftar di DKI Jakarta (kecuali milik TNI/Polri) pada bulan Juni 2009 adalah sebanyak

9,993,867 kendaraan (9,739,633 kendaraan pribadi dan 254,234 kendaraan umum), sedangkan jumlah penduduk DKI Jakarta pada bulan Maret 2009 sebanyak 8,513,385 jiwa [1]. Dengan membandingkan antara banyaknya kendaraan dengan jumlah penduduk, maka dapat disimpulkan bahwa dalam satu keluarga di DKI Jakarta, terdapat sekitar tiga kendaraan bermotor. Dapat diprediksi bahwa pada tahun 2014, DKI Jakarta dapat mengalami macet total, karena pertumbuhan jumlah kendaraan tidak diimbangi dengan pertumbuhan pelebaran atau pembuatan jalan.

Kondisi di atas menjadi motivasi utama diadakannya penelitian mengenai Sistem Pengaturan Lampu Lalu Lintas Terdistribusi (SPLLLT). Lampu lalu lintas konvensional yang ada pada saat ini, masih sekedar berorientasi pada waktu, bersifat *standalone*, dan tidak adaptif terhadap kondisi jalan. SPLLLT diharapkan dapat menjadi bentuk evolusi dari lampu lalu lintas konvensional. Sistem ini dapat menjadi sebuah sistem lampu lalu lintas yang mampu melakukan

penjadwalan dan pengaturan jaringan multi persimpangan berbasis data *real-time*.

SPLLLT memiliki fungsi untuk memahami kondisi jalan dengan menggunakan metode *Principal Component Analysis* (PCA). Metode ini mampu mengenali kendaraan dari gambar yang ditangkap oleh kamera video yang diarahkan ke jalan raya untuk memantau keadaan jalan secara *real-time*. Sistem ini juga akan mampu menghubungkan lampu di masing-masing persimpangan jalan menggunakan metode *Distributed Constraint Satisfaction Problem* (DCSP). Metode ini akan membuat persimpangan jalan yang ada dapat saling berkomunikasi, berbagi data, dan bekerja sama.

Penelitian ini merupakan salah satu bagian penelitian dari kerangka riset besar Sistem Pengaturan Lampu Lalu Lintas Terdistribusi. Penelitian ini bertujuan untuk meningkatkan akurasi proses pengenalan dan penghitungan jumlah mobil dengan melakukan proses klasifikasi mobil. Dalam hal ini, digunakan pula informasi jenis mobil dan jumlahnya sehingga kesimpulan kepadatan jalan yang dapat ditarik akan semakin baik.

Proses klasifikasi akan diimplementasikan menggunakan tiga algoritma Jaringan Syaraf Tiruan (JST) yaitu *Backpropagation*, *Fuzzy Learning Vector Quantization* (FLVQ), dan *Fuzzy Learning Vector Quantization Particle Swarm Optimization* (FLVQ-PSO). Semua algoritma tersebut akan diujidengan cara memasukan citra mobil yang sama, kemudian dievaluasi tingkat presisinya.

2. Metodologi

Penelitian induk dari penelitian ini telah dilakukan dalam kurun dua tahun terakhir, yang dipimpin oleh Wisnu Jatmiko dalam [2]; Abdul Arfan dalam [3]; dan Adhitya Novian Raidy dalam [4] telah mencoba melakukan simulasi lampu lalu lintas dengan mengimplementasikan model Kuramoto. Abdul Arfan mengompilasikan model Kuramoto dengan penelitian Kosuke Sekiyama dalam sebuah simulasi dua dimensi menggunakan bahasa pemrograman Java. Sedangkan Adhitya Novian Raidy mengimplementasikan model Kuramoto dalam sebuah simulasi tiga dimensi menggunakan bahasa pemrograman C++ dengan menggunakan *library* ODE.

Adi Wibowo dalam [5] juga telah mencoba mengembangkan sebuah sistem besar yang terdiri dari sebuah *engine* untuk melakukan perhitungan terhadap data-data yang diberikan oleh sistem sebelumnya yang telah dikembangkan oleh Syarif A.R dalam [6] dan Hendra Marihot dalam [7].

Proses visualisasi untuk menggambarkan komponen-komponen lalu lintas seperti jalan, mobil, dan lampu lalu lintas juga pernah dikembangkan oleh Michael Wicaksana dalam [8]. Kemudian, *car sensor* yang bertugas untuk menghitung berapa jumlah kendaraan yang ada di sebuah jalan dan menghitung kecepatan rata-rata dari seluruh kendaraan yang melewati jalan tersebut telah dikembangkan oleh Manggala Eka Adideswar dalam [9], dan Faris Al Afif dalam [10].

Principal Component Analysis (PCA) atau analisis komponen utama adalah suatu prosedur matematika yang menganalisis data secara statistik dengan tujuan untuk mengekstraksi beberapa variabel yang memiliki kemungkinan korelasi satu sama lain yang disebut *principal component* (komponen utama). Dalam beberapa penerapannya, komponen utama ini biasa disebut sebagai fitur. PCA berfungsi menghitung tingkat variabilitas dalam data yang ditemukan, dan menghitung setiap komponen yang memeruskannya. Selain itu, PCA juga menghitung tingkat variasi yang tersisa dan mungkin berkorelasi.

Secara teori, PCA biasa dikenal sebagai sebuah model komputasi yang paling tepat dalam merepresentasikan suatu objek dengan mengekstraksi informasi yang paling relevan yang terkandung dalam objek tersebut. Tidak semua data relevan untuk diekstraksi menggunakan PCA, salah satu jenis data yang sering diolah menggunakan PCA adalah data citra *digital*.

Ekstraksi PCA pada gambar mengenalkan kita pada konsep selanjutnya yakni *eigen vector* (vektor eigen). Pendekatan *eigen vector* pada metode PCA merupakan suatu pendekatan di mana sekelompok kecil karakteristik gambar digunakan untuk mendapatkan variasi antara gambar objek tertentu, misalnya mobil sedan, mobil mini-bus, truk, bis, dan jenis mobil lainnya. Tujuannya adalah untuk menemukan *eigen vector* dari *matriks kovarian* dari distribusi objek tersebut yang dibentangkan dengan pelatihan menggunakan serangkaian gambar objek. Setiap gambar objek selanjutnya diwakili oleh kombinasi linear dari *eigen vector* ini. Pengenalan dilakukan dengan memproyeksikan citra atau gambar baru ke *subspace* yang membentang oleh *eigen vector* dan kemudian mengelompokkan mobil dengan membandingkan posisinya dalam ruang objek dengan posisi tertentu. Misalkan untuk sebuah wajah, nilai *eigen*-nya diwakilkan oleh bentuk dan posisi kedua mata, posisi hidung, dan bibir. Pada data citra mobil, nilai *eigen* yang dianggap merepresentasikan adalah letak kaca depan, posisi dua lampu, yang dianggap merupakan komponen utama dari sebuah mobil.

Jaringan Syaraf Tiruan (JST) adalah sebuah konsep yang mengadopsi keterkaitan antar neuron pada jaringan syaraf manusia. JST pun dikembangkan untuk mencoba membuat suatu sistem yang dapat melakukan proses pembelajaran sendiri layaknya manusia tanpa harus setiap saat mengubah kondisi masukan atau sistem dan memiliki kecenderungan semakin pandai dalam melakukan keputusan selanjutnya.

JST terdiri atas neuron-neuron yang saling terhubung satu sama lain. Setiap neuron dapat terhubung dengan sejumlah neuron, keterhubungan ini memiliki karakteristik yang sama dengan *sinapsis* pada sistem jaringan syaraf manusia yang menghubungkan akson dengan *dendrit*. Sistem JST dibangun berdasarkan cara kerja jaringan syaraf manusia. Jaringan syaraf manusia terdiri dari sel-sel syaraf yang disebut neuron yang tersusun dari *dendrit* yang menerima masukan dari neuron lain. Kemudian akson yang berfungsi meneruskan masukan tersebut ke neuron yang lain.

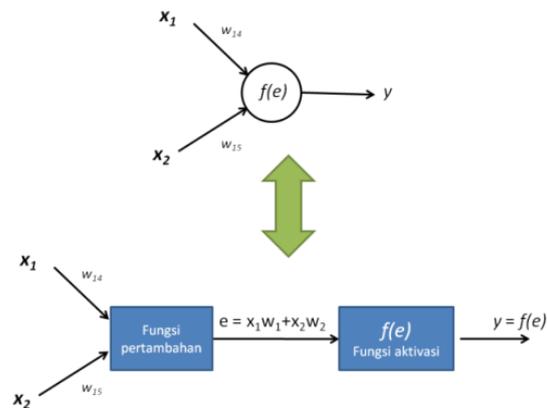
Paradigma pembelajaran JST dikategorikan menjadi dua yakni pembelajaran dengan pengarah (*supervised*) dan pembelajaran tidak dengan pengarah (*unsupervised*).

Backpropagation adalah salah satu algoritma klasifikasi JST (*neural network*). Algoritma *Backpropagation* adalah metode pembelajaran yang dikembangkan dari aturan *multilayer perceptron*. Algoritma *Backpropagation* ini pertama kali diperkenalkan oleh Arthur E. Bryson dan Yu-Chi Ho di tahun 1969, kemudian dikembangkan oleh Werbos, Rumelhart, Hinton, dan Williams di tahun 1986.

Backpropagation terkenal sebagai salah satu algoritma JST yang paling sederhana dan mudah untuk diimplementasikan. *Backpropagation* terdiri dari tiga *layer* yakni *input layer*, *hidden layer*, dan *output layer*. Setiap *layer* memiliki neuron masing-masing dengan jumlah yang bervariasi berdasarkan jenis *input*, jumlah fitur, dan pertimbangan lainnya. Setiap neuron di dalam *hidden layer* dan *output layer* memiliki fungsi yang melakukan fungsi perhitungan terhadap masukan dan bobotnya, fungsi ini disebut sebagai fungsi aktivasi. Hasil perhitungan fungsi tersebut akan diakumulasikan ke setiap neuron selanjutnya, hingga memunculkan nilai klasifikasi dari *input* tersebut (lihat gambar 1).

Fuzzy Learning Vector Quantization (FLVQ) adalah salah satu algoritma klasifikasi JST yang bersifat *supervised*. FLVQ merupakan modifikasi dari algoritma LVQ yang dipadukan

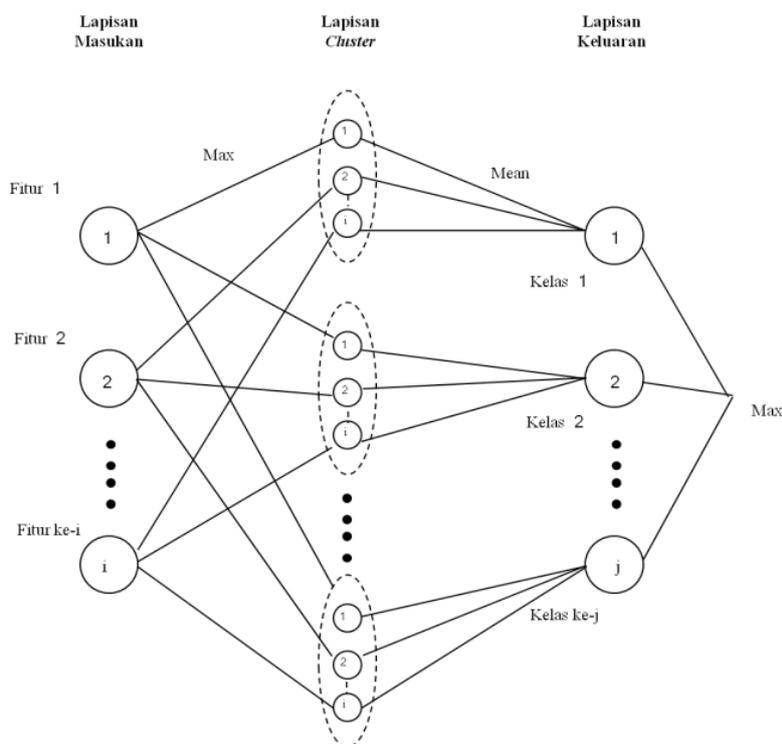
dengan konsep *fuzzy* pada inisialisasi masukannya.



Gambar 1. Isi sebuah neuron dalam *backpropagation*.

Seperti halnya LVQ, FLVQ memiliki prinsip *winner take all*, yaitu untuk setiap data yang direpresentasikan dalam vektor, vektor yang paling dekat dengan masukanlah yang menjadi vektor pemenang dibandingkan vektor-vektor lainnya. Dalam FLVQ juga terjadi pergeseran posisi vektor-vektor sesuai dengan kondisi keluarannya, yaitu bila vektor tersebut menjadi vektor pemenang maka akan didekatkan, namun sebaliknya bila kalah akan dijauhkan.

Dengan digabungkannya konsep *fuzzy* dengan LVQ membuat FLVQ memiliki kelebihan dalam hal keunggulan waktu komputasi yang cepat layaknya LVQ serta tingkat pengenalan yang lebih tinggi daripada algoritma *Backpropagation* atau jaringan syaraf probabilistik. Selain itu FLVQ juga sangat baik dalam hal pengklasifikasian data yang memiliki tingkat kemiripan (*similaritas*) yang tinggi. Arsitektur jaringan FLVQ dapat digambarkan seperti pada gambar 2. Secara umum, FLVQ mirip dengan algoritma JST lainnya, yaitu terdiri dari tiga lapisan yakni lapisan masukan (*input layer*), lapisan keluaran (*output layer*), serta lapisan *cluster*. *Input layer* diisi oleh neuron masukan sesuai dengan data yang akan diklasifikasi. *Output layer* diisi oleh neuron keluaran yang direncanakan, sedangkan lapisan *cluster* atau lapisan tersembunyi adalah lapisan yang berfungsi untuk menghitung nilai kesamaan setiap vektor masukan terhadap vektor perwakilan. Banyaknya neuron pada lapisan *cluster* adalah hasil perkalian antara banyaknya neuron pada lapisan masukan dengan banyaknya neuron pada lapisan keluaran. Neuron-neuron ini yang disebut dengan istilah vektor perwakilan.



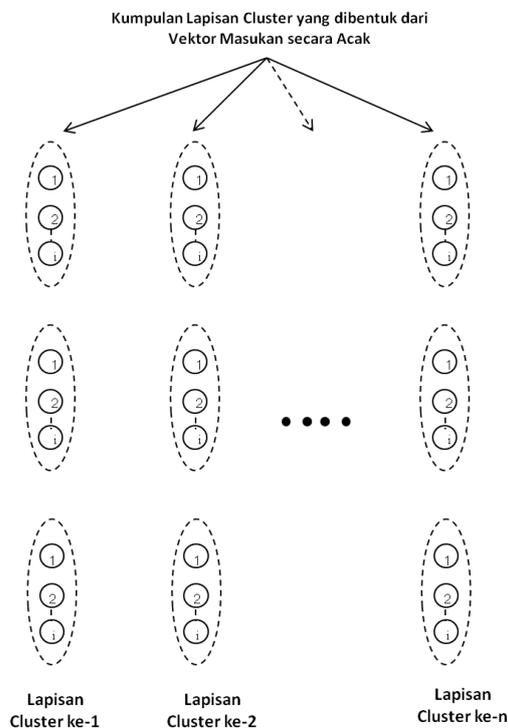
Gambar 2. Arsitektur jaringan FLVQ.

Hal yang membedakan FLVQ dengan algoritma JST lain adalah bahwa pada FLVQ bobot dari jaringan direpresentasikan dengan fungsi keanggotaan dari vektor perwakilan dalam lapisan *cluster*, sedangkan fungsi aktivasinya adalah merupakan hasil irisan fungsi keanggotaan antara vektor masukan dengan vektor perwakilan.

Fuzzy Learning Vector Quantization Particle Swarm Optimization (FLVQ-PSO) adalah bentuk improvisasi dari metode FLVQ yang telah dijelaskan sebelumnya. FLVQ-PSO dikembangkan pertama kali oleh Benyamin Kusumoputro dan rekan-rekan di dalam [11]. Konsep FLVQ yang merupakan bentuk modifikasi dari algoritma LVQ masih dianggap belum cukup karena FLVQ memiliki ketergantungan yang besar terhadap inisialisasi vektor awal, jika tidak sesuai maka nilai akurasi klasifikasi akan jauh dari harapan. Hal ini diatasi di dalam FLVQ-PSO dengan cara membentuk vektor perwakilan yang lebih dari satu, dan dipilih secara acak sebanyak partikel yang dikehendaki (lihat gambar 3).

Konsep FLVQ-PSO merupakan gabungan dari konsep FLVQ, *Matrix Similarity Analysis (MSA)*, dan *Particle Swarm Optimization (PSO)*. FLVQ digunakan karena kelebihan kecepatannya dalam melakukan proses pelatihan, MSA digunakan untuk menentukan nilai *fitness*, sedangkan PSO diadopsi pula untuk melakukan

perubahan posisi vektor *fuzzy* agar menghasilkan posisi yang lebih optimal.



Gambar 3. Ilustrasi kumpulan vektor perwakilan yang dibentuk dari vektor masukan.



Gambar 7. Proses *Scaling*.

Pada penelitian ini, penulis menggunakan percobaan terhadap 120 gambar yang terdiri dari dua kelas, yakni kelas mobil kecil (kelas yang mewakili mobil-mobil ukuran kecil seperti *sedan*, *jeep*, angkot, dan sebagainya) serta kelas mobil besar (kelas yang mewakili mobil-mobil berukuran besar seperti bus, truk, dan sejenisnya). Dari masing-masing kelas dipilih 60 gambar.

Kodifikasi masing-masing masukan akan disimpan dalam bentuk *file* teks (.txt), dengan informasi nama *file* beserta kelas targetnya. Contoh bentuk *file* teks dapat dilihat dalam gambar 8.

new 5	main.cpp	new 5	Target TA-11 Juni 2011.txt	fileEigen-4train.txt		
1	-1.309	0.2274	0.4048	-1.139	1	-1
2	0.7759	-1.895	-0.5834	-1.036	-1	1
3	-1.201	0.4334	-0.3507	-1.117	1	-1
4	1.02	0.9086	0.08465	0.02719	-1	1
5	-0.1696	-0.9747	1.116	1.91	1	-1
6	1.017	0.8688	0.1601	0.01266	-1	1
7	-0.7957	-0.02455	2.112	-0.4221	1	-1
8	0.7759	-1.895	-0.5834	-1.036	-1	1
9	-1.225	0.7109	-1.344	0.08994	1	-1
10	0.7759	-1.895	-0.5834	-1.036	-1	1
11	-0.6809	-0.1268	-3.033	2.32	1	-1
12	1.02	0.9086	0.08465	0.02719	-1	1
13	-0.8507	0.1106	1.874	-0.2175	1	-1
14	1.02	0.9086	0.08465	0.02719	-1	1

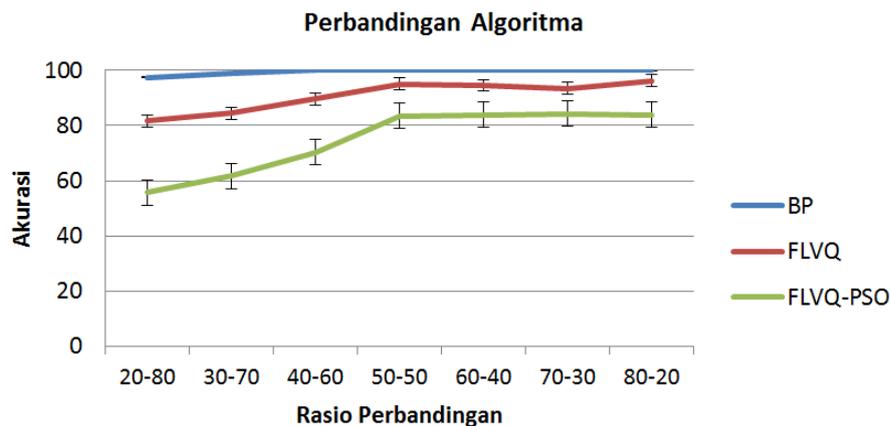
Gambar 8. Contoh nilai *Eigen* hasil konversi dari gambar masukan.

Klasifikasi dengan JST. Ini adalah proses utama dari penelitian ini. Proses klasifikasi dilakukan terhadap data masukan *fileEigen.txt* yang telah dibuat sebelumnya. Seperti yang telah dipaparkan sebelumnya, proses pengenalan mobil dalam penelitian ini dilakukan menggunakan tiga algoritma JST yakni *Backpropagation*, FLVQ, dan FLVQ-PSO.

Eksperimen pengujian dilakukan dengan desain eksperimen sebagai berikut. Pertama, data masukan. Data masukan adalah daftar nilai-nilai *eigen* yang merepresentasikan masing-masing gambar mobil yang akan diklasifikasikan. Daftar nilai *eigen* ini disimpan dalam sebuah *file* dengan tipe .txt dengan jumlah 120 data gambar. Metode *cross validation* dengan *10-fold* digunakan untuk menentukan rasio perbandingan data pelatihan dan pengujian. Terdapat tujuh kategori rasio perbandingan data yakni 20:80, 30:70, 40:60, 50:50, 60:40, 70:30, dan 80:20. Kedua, kelas target. Kelas target yang diujikan terdiri dari dua kelas, mobil kecil dan besar di mana mobil kecil direpresentasikan dengan nilai target [1 -1], sedangkan mobil besar dengan nilai target [-1 1]. Ketiga, jumlah fitur. Fitur yang digunakan untuk merepresentasikan sebuah gambar terdiri dari beberapa kombinasi fitur. Penulis memutuskan untuk menggunakan empat fitur yang didapatkan dengan metode PCA karena dengan empat fitur sudah mewakili sekitar 85% representasi gambar yang sebenarnya. Terakhir, nilai laju pembelajaran dan jumlah *epoch*. Nilai laju pembelajaran dan jumlah *epoch* yang digunakan untuk melakukan uji komparasi adalah nilai terbaik dari masing-masing algoritma yang telah diteliti sebelumnya. Adapun hasil percobaannya penggunaan berbagai nilai laju pembelajaran dan jumlah *epoch* dari masing-masing metode dapat dilihat pada tabel I.

TABEL I
LAJU PEMBELAJARAN (ALPHA) DAN EPOH YANG DIGUNAKAN

Algoritma	Alpha	Epoch
FLVQ	0.02	100
PSO	0.02	100
BP	0.045	45



Gambar 9. Grafik perbandingan masing-masing algoritma.

Percobaan juga dilakukan sebanyak sepuluh kali untuk masing-masing skenario dan dirata-rata untuk mendapatkan kesimpulan yang lebih objektif. Hasil ujicoba untuk masing-masing algoritma adalah dapat dilihat pada tabel II. Grafik perbandingannya juga dapat dilihat pada gambar 9.

TABEL II
AKURASI HASIL UJICoba DARI TIGA ALGORITMA JST (DALAM %)

Rasio	BP	FLVQ	PSO
20-80	97.5	81.69	55.81
30-70	99.05	84.43	61.71
40-60	100	89.58	70.39
50-50	100	95	83.43
60-40	100	94.5	83.92
70-30	100	93.5	84.22
80-20	100	96.25	83.83
Mean	99.51	90.71	74.76
Std Dev	0.953	5.677	12.11

Berdasarkan hasil uji coba, algoritma *Backpropagation* tampak mengungguli kedua algoritma JST lainnya. *Backpropagation* memiliki akurasi yang baik dalam rasio perbandingan manapun, dan dapat mempertahankan akurasinya diatas 97% dengan rata-rata 99.51%. FLVQ memiliki performa yang cukup baik, di mana akurasi terendahnya hanya 81.69% namun memiliki akurasi rata-rata 90.71%. FLVQ-PSO yang masih kurang baik dan jauh dari ekspektasi yakni memiliki rata-rata 74.76% dengan akurasi terburuk hingga mencapai 55.81%.

4. Kesimpulan

Klasifikasi mobil pada SPLLLT dapat diimplementasikan menggunakan beberapa algoritma JST, yaitu *Backpropagation*, FLVQ, dan FLVQ-PSO. Diantara ketiga metode tersebut, *Backpropagation* menunjukkan performa akurasi terbaik dengan akurasi rata-rata 99.51%, disusul

oleh FLVQ dengan 90.71%, serta FLVQ-PSO dengan akurasi rata-rata 74.76%.

Beberapapenelitian-penelitian selanjutnya yang dapat dilakukan untuk mendukung realisasi penerapan sistem klasifikasi mobil ini diantaranya metode klasifikasi dikembangkan untuk kelas target yang lebih banyak dengan berbagai macam kondisi kendaraan dalam lalu lintas. Selain itu, dapat juga mengeksplorasi metode penjejakkan mobil, fase sebelum proses klasifikasi mobil agar dapat mengidentifikasi objek-objek mobil dari video streaming menjadi objek-objek mobil yang siap untuk diklasifikasi.

Penelitian-penelitian diatas diharapkan membuat SPLLT ini selangkah lebih siap untuk diluncurkan di lapangan sehingga dapat segera menjadi salah satu solusi kemacetan di kota-kota besar di Indonesia.

Referensi

- [1] Komisi Kepolisian Indonesia, Jumlah Kendaraan Bermotor Juni 2009, <http://www.komisikepolisianindonesia.com/main.php?page=artikel&id=1187>, 2009, retrieved June 3, 2011.
- [2] W. Jatmiko, et. al., "Self-Organizing Urban Traffic Control Architecture With Swarm-Self Organizing Map In Jakarta: Signal Control System And Simulator," *International Journal on Smart Sensing and Intelligent System*, vol. 3, pp 443 – 465, 2010.
- [3] A. Arfan, "Sistem Pengaturan Lalu Lintas Terdistribusi dengan Menggunakan Nonlinear Coupled Oscillator," B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [4] A.N. Raidy, "Pengembangan Simulasi Pengaturan Lampu Lalu Lintas Secara Terdistribusi dengan Menggunakan Open

- Dynamics Engine,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [5] A. Wibowo, “Optimasi Pengaturan Lampu Lalu Lintas Secara Terkoordinasi Berbasis Self Organizing Control,” Ph.D Thesis. Faculty of Computer Science, Universitas Indonesia, Indonesia, 2011.
- [6] S. Abdurrahman, “Usulan Pengembangan Model Matematika Pengaturan Lampu Lalu Lintas Simpang-N Secara Terdistribusi Berbasis Nonlinear Coupled Oscillator dan Sinkronisasi Kuramoto,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [7] H. Marihot, “Desain dan Implementasi Engine Simulasi Pengendalian Lampu Lalu Lintas secara Terdistribusi Menggunakan Model Kuramoto,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [8] M. Wicaksana, “Penerapan Model Component Based Software Engineering untuk Implementasi Visualisasi Simulasi Keadaan Lalu Lintas,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [9] M.E. Adideswar, “Implementasi Pendeteksian dan Penghitungan Mobil dengan Pemrosesan Video untuk Sistem Pengendalian Lampu Lalu Lintas Terdistribusi,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2010.
- [10] F.A. Afif, “Implementasi Prototipe Sistem Pengaturan Lampu Lalu Lintas Terdistribusi dengan Optimasi Pengenalan dan Penjejakan Kendaraan Berbasis Pemrosesan Video,” B.S Thesis, Faculty of Computer Science, Universitas Indonesia, Indonesia, 2011.
- [11] B. Kusumoputro, H. Budiarto, & W. Jatmiko, “Fuzzy-neuro LVQ and its comparison with fuzzy algorithm LVQ in artificial odor discrimination system,” *ISA Transactions*, vol. 41, pp 395-407, 2002.