

DISERTASI

**PEMODELAN EKSTRAKSI CIRI OBJEK LALU LINTAS PADA
*SMART VIDEO SENSOR***

***FEATURE EXTRACTION MODELING FOR TRAFFIC OBJECTS
ON SMART VIDEO SENSOR***



IMELDA
11/326055/SPA/00409

**PROGRAM STUDI S3 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GAD JAH MADA
YOGYAKARTA**

2018

DISERTASI

**PEMODELAN EKSTRAKSI CIRI OBJEK LALU LINTAS PADA
*SMART VIDEO SENSOR***

***FEATURE EXTRACTION MODELING FOR TRAFFIC OBJECTS
ON SMART VIDEO SENSOR***

Disertasi untuk memperoleh derajat
Doktor dalam Ilmu Komputer pada
Universitas Gadjah Mada



IMELDA
11/326055/SPA/00409

**PROGRAM STUDI S3 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GAD JAH MADA
YOGYAKARTA**

2018

HALAMAN PENGESAHAN

DISERTASI

PEMODELAN EKSTRAKSI CIRI OBJEK LALU LINTAS PADA SMART VIDEO SENSOR

IMELDA
11/326055/SPA/00409


Dipertahankan di hadapan Dewan Penguji Program Studi S3 Ilmu Komputer
Departemen Ilmu Komputer dan Elektronika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Gadjah Mada
Pada tanggal 28 Mei 2018




Dr. Ing. MHD. Reza M.L. Pulungan, S.Si., M.Sc.
Ketua Tim Penguji




Drs. Agus Harjoko, M.Sc., Ph.D.
Promotor



Prof. Dra. Sri Hartati, M.Sc., Ph.D.
Penguji



Drs. Pekik Nurwantoro, M.S., Ph.D.
Ko-Promotor



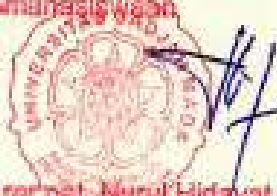
Dr. Tri Kuntoro Priyambodo, M.Sc.
Penguji



Dr. Subarto

Penguji

Mengetahui
a.n. Dekan FMIPA-UGM
Wakil Dekan Bidang Akademik dan
Kemahasiswaan



Dr. rer.nat. Nurul Hidayat Aprilia, M.Si.
NIP. 197304071998031002



Dr. Ir. Albertus Joko Santoso, M.T.

Penguji




Wahyono, S.Kom, Ph.D.

Penguji

PERNYATAAN

Dengan ini saya menyatakan bahwa Disertasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Doktor di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 28 Mei 2018

A handwritten signature in blue ink, appearing to read 'Imelda', written over a horizontal line.

Imelda

Percayalah kepada TUHAN dengan segenap hatimu,
dan janganlah bersandar kepada pengertianmu sendiri.
(Amsal 3:5)

Janganlah hendaknya kamu kuatir tentang apapun juga,
tetapi nyatakanlah dalam segala hal keinginanmu kepada Allah
dalam doa dan permohonan dengan ucapan syukur.
(Filipi 4:6)

Sebab di dalam Dia kamu telah menjadi kaya dalam segala hal:
dalam segala macam perkataan dan segala macam pengetahuan,
(1 Korintus 1:5)

oleh karna kemurahan Tuhan
ku ada sampai hari ini
oleh karna kebaikan Tuhan
janjiMu terjadi bagiku

ku sembah Kau Tuhan
ku mengangkat tanganku
slamanya Kau ajaib bagiku
ku sembah Kau Tuhan
ku mengangkat wajahku
slamanya Kau hebat bagiku

PRAKATA

Puji syukur pada TUHAN atas Kasih dan Kuasa-Nya yang ajaib yang memberi kekuatan dan hikmat dari TUHAN yang memperlengkapi setiap hal. Sungguh besar karya TUHAN yang bekerja sehingga penulisan disertasi dapat terselesaikan.

Dalam penyusunan disertasi ini penulis telah banyak mendapatkan arahan, bantuan, dan doa serta dukungan dari berbagai pihak. Disertasi yang terselesaikan ini kiranya menjadi ungkapan terima kasih yang terdalam atas doa, dukungan, dan semangat dari kedua orang tua, suami, anakku Yohanes Bryan dan adik-adikku.

Ungkapan terima kasih yang mendalam juga ditujukan kepada :

1. Bapak Drs. Agus Harjoko, M.Sc, Ph.D. selaku Promotor sekaligus Ketua Departemen Ilmu Komputer dan Elektronika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada yang telah menyediakan waktu, tenaga dan pikiran dalam memberi petunjuk, inspirasi, mengarahkan dan memotivasi pada saat bimbingan hingga laporan disertasi ini terselesaikan.
2. Bapak Drs. Pekik Nurwantoro, M.S., Ph.D. selaku Ko-Promotor yang telah menyediakan waktu, tenaga dan pikiran dalam memberikan arahan, inspirasi, motivasi dalam menyelesaikan laporan disertasi ini.
3. Bapak Drs. Edi Winarko, M.Sc., Ph.D. selaku Ketua Program Studi Monodisiplin S3 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada sekaligus Dosen saya atas motivasi, doa dan dukungan untuk fokus, gigih dan disiplin dalam penyelesaian studi.
4. Bapak Prof. Dr. Triyono, S. U. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
5. Segenap Dosen dan civitas akademika di lingkungan Program Studi Ilmu Komputer, Departemen Ilmu Komputer dan Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada. Terima kasih untuk semua ilmu, tuntunan, arahan, semangat dan motivasi yang terus diberikan untuk segera menyelesaikan disertasi sampai tuntas.
6. Bapak Drs. Djaetun H.S. selaku Ketua Yayasan Pendidikan Budi Luhur Cakti atas dukungan, semangat, motivasi dan doa dalam studi lanjut penulis.
7. Bapak Kasih Hanggoro, MBA selaku Ketua Badan Pelaksana Harian Yayasan Pendidikan Budi Luhur Cakti atas izin, kesempatan, kepercayaan, dukungan, semangat, motivasi dan doa dalam studi lanjut penulis.
8. Bapak Prof. Dr. sc. agr. Ir. Didik Sulistyanto selaku Rektor Universitas Budi Luhur masa bakti 2017 sampai saat ini atas dukungan, semangat dan motivasi dalam menyelesaikan pendidikan Strata-3 di Universitas Gadjah Mada.
9. Bapak Prof. Ir. Suryo Hapsoro Tri Utomo, Ph.D. selaku Rektor Universitas Budi Luhur masa bakti 2012 - 2016 atas dukungan, semangat dan motivasi dalam menyelesaikan pendidikan Strata-3 di Universitas Gadjah Mada.
10. Bapak – bapak dan Ibu - ibu serta rekan seperjuangan S3 Ilmu Komputer Universitas Gadjah Mada juga teman-teman OT yang namanya tidak dapat disebutkan satu persatu. Terima kasih karena telah diijinkan menjadi sahabat dan teman dalam menyelesaikan laporan disertasi ini,

11. Pak Sugeng, Mbak Retno, Mbak Rini, Mbak Vera, Mas Hoho, Mas Kuncoro, Pak Gori, Pak Medel, Pak Jumari, Pak Sudiman dan Bu Septi serta teman dan karyawan lain di FMIPA Universitas Gadjah Mada yang namanya tidak dapat disebutkan. Terima kasih telah memberikan banyak bantuan selama penulis menempuh studi pasca sarjana Universitas Gadjah Mada,
12. Segenap civitas Akademika Universitas Budi Luhur yang namanya tidak dapat disebutkan satu persatu. Terima kasih untuk dukungan yang terus diberikan hingga laporan disertasi ini terselesaikan.

Penulis menyadari bahwa penulisan laporan disertasi ini masih jauh dari sempurna dan banyak kekurangan. Karena itu penulis sangat mengharapkan saran dan masukan serta kritikan dari semua pihak, sehingga penulis dapat memperbaiki dan menyempurnakannya.

Akhir kata penulis berharap laporan disertasi ini dapat memberikan manfaat bagi kita semua, terutama bagi perkembangan ilmu pengetahuan serta perkembangan Ilmu Komputer dan Teknologi Informasi.

Yogyakarta, 28 Mei 2018

Imelda

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
PERNYATAAN.....	iii
HALAMAN MOTTO.....	iv
HALAMAN PERSEMBAHAN.....	v
PRAKATA.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
DAFTAR ISTILAH.....	xv
INTISARI.....	xvii
<i>ABSTRACT</i>	xviii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	6
1.5 Manfaat Penelitian.....	6
1.6 Kontribusi Penelitian.....	7
1.7 Sistematika Penulisan.....	11
BAB 2 KAJIAN PUSTAKA.....	12
2.1 Penelitian tentang <i>Smart Video Sensor (SVS)</i>	12
2.2 Penelitian tentang Ciri.....	15
BAB 3 LANDASAN TEORI.....	21
3.1 <i>Smart Video Sensor</i>	21
3.2 Objek Lalu Lintas.....	23
3.3 Pengolahan Citra Digital.....	24
3.4 Ciri.....	29
3.4.1 Ciri Bentuk.....	30
3.4.2 Ciri Tekstur.....	44
3.4.3 Ciri Warna.....	56
3.5 <i>Traffic Parameter</i>	61
3.6 Pemampatan Video.....	66
3.7 <i>eXtensible Markup Language (XML)</i>	67

3.8	Transmisi Data.....	69
3.9	<i>Histogram Equalization</i>	70
3.10	<i>Gaussian Mixture Model (GMM)</i>	70
3.11	Operasi Morfologi.....	73
3.11.1	Operasi Dilasi.....	73
3.11.2	Operasi Erosi.....	74
3.11.3	Operasi <i>Opening</i>	74
3.11.4	Operasi <i>Closing</i>	75
3.12	<i>K-Nearest Neighbor (KNN)</i>	75
3.13	Kalman Filter.....	77
BAB 4	METODE PENELITIAN.....	85
4.1	Gambaran Umum Sistem Pemantauan dan Pengaturan Lalu Lintas	85
4.2	Arsitektur <i>Smart Video Sensor (SVS)</i>	89
4.1.1	Modul Sensor Video.....	92
4.1.2	Modul Ekstraksi Ciri.....	94
4.1.3	Modul Representasi Ciri.....	96
4.1.4	Modul Pemampatan Video.....	98
4.1.5	Modul Transmisi Data.....	99
4.3	Alat Penelitian.....	100
4.4	Data Penelitian.....	100
4.5	Rencana Pengujian.....	101
4.5.1	Rencana pengujian pada representasi ciri dalam format XML	101
4.5.2	Rencana pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS.....	103
4.5.3	Rencana pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS....	106
BAB 5	HASIL DAN PEMBAHASAN.....	108
5.1	Ekstraksi Ciri dan Representasi Ciri dalam SVS.....	108
5.1.1	Pra pemrosesan.....	109
5.1.2	Segmentasi.....	110
5.1.3	Ekstraksi Ciri.....	114
5.1.4	Representasi.....	116
5.2	Hasil Pengujian.....	119
5.2.1	Hasil pengujian pada representasi ciri dalam format XML..	120

5.2.2 Hasil pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS	127
5.2.2.1 Training (Pelatihan)	128
5.2.2.1.1 Data video	129
5.2.2.1.2 Pra pemrosesan	129
5.2.2.1.3 Segmentasi	129
5.2.2.1.4 Ekstraksi ciri.....	129
5.2.2.1.5 K-Nearest Neighbor	130
5.2.2.2 Testing (Pengujian)	132
5.2.3 Hasil pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS	134
BAB 6 PENUTUP.....	138
6.1 Kesimpulan	138
6.2 Saran	139
DAFTAR PUSTAKA	140

DAFTAR GAMBAR

Gambar 1.1 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta.....	2
Gambar 1.2 Perbedaan Arsitektur SVS Sánchez dkk. (2012) dan usulan	8
Gambar 1.3 Perbedaan alur proses Sánchez dkk. (2012) dan usulan.....	10
Gambar 3.1 Tahapan dasar Pengolahan Citra Digital (Gonzalez dkk., 2008)	26
Gambar 3.2 Beberapa ciri bentuk.....	31
Gambar 3.3 HOG (a) Area objek 8x8 (b) blok	35
Gambar 3.4 Sudut 180°	35
Gambar 3.5 Ilustrasi PHOG	37
Gambar 3.6 Convex area.....	38
Gambar 3.7 Contoh himpunan Haar like feature	41
Gambar 3.8 Contoh cara kerja operator LBP	45
Gambar 3.9 Contoh perhitungan LBP.....	46
Gambar 3.10 Contoh pembentukan GLCM pada arah 0° dan 45°	49
Gambar 3.11 Ilustrasi <i>Color</i> HOG	57
Gambar 3.12 Proses rekursif <i>Kalman Filter</i> (Welch dan Bishop, 2006)	79
Gambar 4.1 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang berjalan saat ini	86
Gambar 4.2 Ilustrasi kemanfaatan SVS pada pelacakan kendaraan yang terklasifikasi	87
Gambar 4.3 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang diusulkan	88
Gambar 4.4 Arsitektur <i>Smart Video Sensor</i> yang diusulkan.....	90
Gambar 4.5 Tahapan pada <i>Smart Video Sensor</i>	93
Gambar 4.6 Modul Ekstraksi Ciri	94
Gambar 4.7 Tahapan pada Modul Ekstraksi Ciri.....	94
Gambar 4.8 Tahapan Ekstraksi Ciri	96
Gambar 4.9 Modul Representasi Ciri.....	97
Gambar 4.10 Tahapan pada Modul Representasi Ciri	97
Gambar 4.11 Tahapan pada Modul Pemampatan Video	98
Gambar 4.12 Tahapan pada Modul Transmisi Data	100
Gambar 4.13 Tahapan pengujian pada representasi ciri	102
Gambar 4.14 Tahapan pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS	104
Gambar 4.15 Tahapan pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS	107
Gambar 5.1 Model Ekstraksi Ciri dan Representasi Ciri.....	108
Gambar 5.2 Citra sebelum dan sesudah <i>Histogram Equalization</i>	110
Gambar 5.3 Citra sebelum dan sesudah segmentasi menggunakan <i>Gaussian</i> <i>Mixture Model</i>	111
Gambar 5.4 Citra sebelum dan sesudah morfologi <i>opening</i> dan <i>closing</i>	111
Gambar 5.5 Hubungan antara ukuran segmen dengan akurasi kendaraan di	

Babarsari	115
Gambar 5.6 Skema representasi dari ciri-ciri objek lalu lintas yang diusulkan..	118
Gambar 5.7 Struktur XML	119
Gambar 5.8 <i>Frame video 1</i>	121
Gambar 5.9 Skema representasi ciri dari frame video 1	122
Gambar 5.10 Ciri dalam format XML dari frame video 1	123
Gambar 5.11 Keluaran ciri dalam format XML frame video 1 ke data numerik	124
Gambar 5.12 <i>Frame Video 2</i>	124
Gambar 5.13 Ciri dalam format XML dari frame video 2	126
Gambar 5.14 Keluaran ciri dalam format XML ke data numerik	127
Gambar 5.15 Model klasifikasi kendaraan berdasarkan Arsitektur SVS	128
Gambar 5.16 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Babarsari	131
Gambar 5.17 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Galeria	132
Gambar 5.18 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Condong Catur	133
Gambar 5.19 Pelacakan Kendaraan sesuai jenisnya yang ciri-ciri objeknya berasal dari SVS	135
Gambar 5.20 Hasil pengujian pelacakan kendaraan berdasarkan Arsitektur SVS di Yogyakarta (a) Babar Sari, (b) Galeria, (c) Condong Catur	136

DAFTAR TABEL

Tabel 2.1 Pustaka penelitian tentang Smart Video Sensor	13
Tabel 2.2 Rangkuman pustaka penelitian tentang ciri objek lalu lintas.....	17
Tabel 2.3 Pengelompokan ciri berdasarkan kategori aplikasi.....	19
Tabel 3.1 Karakteristik Mendasar dari Arus Lalu Lintas (Board, 2000)	62
Tabel 4.1 <i>Confusion Matrix</i> (Santra dan Christy, 2012; Visa dkk., 2011)	105
Tabel 5.1 Hasil segmentasi beberapa frame di Babarsari	113
Tabel 5.2 Hasil keluaran kelas kendaraan.....	133
Tabel 5.3 Akurasi klasifikasi kendaraan berdasarkan Arsitektur SVS	134
Tabel 5.4 Ilustrasi 1 objek antara <i>centroid</i> dari <i>Kalman Filter</i> dan ciri dalam format XML.....	137
Tabel 5.5. Ilustrasi objek lain antara <i>centroid</i> dari <i>Kalman Filter</i> dan ciri dalam format XML.....	137

DAFTAR LAMPIRAN

Lampiran A. Kumpulan Algoritma	159
Lampiran B. Manfaat Ciri	170
Lampiran C. Kesesuaian nilai centroid dalam Kalman Filter	177
Lampiran D Nama dan Jumlah dimensi ciri	183
Lampiran E. Data set	185
Lampiran F. Rekomendasi Spesifikasi <i>Smart Video Sensor</i>	186

DAFTAR ISTILAH

AIA	: <i>Automated Imaging Association</i>
ANN	: <i>Artificial Neural Network</i>
ATCS	: <i>Area Traffic Control System</i>
AVI	: <i>Audio Video Interleave</i>
CCTV	: <i>Closed-circuit television</i>
DCCU	: <i>Data Center and Control Unit</i>
DIO	: <i>Difference of IN/OUT traffic flow</i>
Dishub	: <i>Dinas Perhubungan</i>
EOT	: <i>End of Transmission</i>
EPHOG	: <i>Edge based-Pyramid of Histogram of Orientation Gradients</i>
ETX	: <i>End of Text</i>
FLV	: <i>Flash Video</i>
FLVQ	: <i>Fuzzy Learning Vector Quantization</i>
FLVQ-PSO	: <i>Fuzzy Learning Quantization Particle Swarm Optimization</i>
<i>fn</i>	: <i>false negatives</i>
<i>fp</i>	: <i>false positives</i>
GIS	: <i>Geographic Information Systems</i>
GMM	: <i>Gaussian Mixture Model</i>
GLCM	: <i>Gray Level Co-occurrence Matrices</i>
GPS	: <i>Global Positioning System</i>
HBB	: <i>Height of the bounding box</i>
HGV	: <i>Heavy Goods Vehicle</i>
HOG	: <i>Histograms of Oriented Gradients</i>
HSV	: <i>Hue, Saturation and Value</i>
HTML	: <i>HyperText Markup Language</i>
IHCM	: <i>Indonesian Highway Capacity Manual and Software</i>
IPHOG	: <i>Intensity based-Pyramid of Histogram of Orientation Gradients</i>
LM	: <i>Local Monitor</i>
LPU	: <i>Local Processing Unit</i>
LVQ	: <i>Learning Vector Quantization</i>
LZW	: <i>Adaptive Dictionary Based</i>
MBF	: <i>Measurement-Based Feature</i>
MLP	: <i>Multi Layer Perceptron</i>
MP4	: <i>Moving Picture 4</i>
MOV	: <i>Apple's QuickTime container format</i>
MKV	: <i>Mastroska</i>
MPEG	: <i>Moving Picturea Expert Group</i>
OVA	: <i>one-vs-all</i>
OVO	: <i>one-vs-one</i>
PHF	: <i>peak-hour factor</i>
PPV	: <i>positive predictive value</i>

RLE	: <i>Run Length Encoding</i>
ROI	: <i>Region Of Interest</i>
RTMC	: <i>Regional Traffic Management Center</i>
SGML	: <i>Standard Generalized Markup Language</i>
SOH	: <i>Start of Heading</i>
SMS	: <i>Space mean speed</i>
STSI	: <i>single time-spatial image</i>
STX	: <i>Start of Text</i>
SVM	: <i>Support Vector Machine</i>
SVS	: <i>Smart Video Sensor</i>
SYN	: <i>Synchronous</i>
TF	: <i>Traffic flow</i>
<i>tn</i>	: <i>true negatives</i>
TOR	: <i>Time occupancy ratio</i>
<i>tp</i>	: <i>true positives</i>
TSI	: <i>time-spatial image</i>
XML	: <i>eXtensible Markup Language</i>
VOB	: <i>Video Object</i>
WBB	: <i>Width of the bounding box</i>
<i>wkNN</i>	: <i>weighted k-nearest neighbor</i>
W3C	: <i>World Wide Web Consortium</i>
3GPP	: <i>3rd Generation Partnership Project</i>

INTISARI

Di Yogyakarta dan kota-kota besar lainnya, Kepolisian dan Dinas Perhubungan masih memonitor lalu lintas secara manual. Hal ini menimbulkan kesulitan apabila beberapa permasalahan lalu lintas terjadi bersamaan. Kesulitan lain yang muncul, data video yang tersedia kadang tidak lengkap karena beberapa frame video ternyata hilang. Akibatnya, data video kurang bermanfaat karena data video tidak dapat diolah lebih lanjut. Penelitian ini bertujuan mengatasi permasalahan frame video yang hilang dengan memodifikasi konsep Arsitektur *Smart Video Sensor (SVS)* yang fokusnya membuat Modul Ekstraksi Ciri dan Modul Representasi Ciri dan yang ciri-cirinya diperlukan untuk pengontrolan lampu lalu lintas, pelacakan kendaraan, mengatasi kepadatan lalu lintas dan parameter lalu lintas lainnya yang langsung dapat digunakan untuk menganalisa data lalu lintas selanjutnya. *SVS* mengekstrak ciri-ciri objek lalu lintas, memampatkan data video dan mengirimkan hasilnya ke *Data Center and Control Unit (DCCU)* atau *Local Processing Unit (LPU)*. *SVS* secara lengkap terdiri dari Modul Sensor Video, Modul Pemampatan Video, Modul Ekstraksi Ciri, Modul Representasi Ciri dan Modul Transmisi Data. Hasil penelitian ini adalah ciri-ciri objek lalu lintas dalam bentuk XML. Di *DCCU*, sistem lain akan menerima data video yang dimampatkan dan ciri-ciri untuk diolah sesuai kebutuhan. Kontribusi penelitian ini adalah memodifikasi konsep Arsitektur *Smart Video Sensor* untuk menyelesaikan masalah frame video yang hilang dengan mengekstrak banyak ciri objek lalu lintas. Ciri-ciri ini dipilih secara hati-hati agar dapat merepresentasi objek-objek dengan baik. Ciri-ciri ini berasal dari kendaraan bermotor yang siap digunakan sesuai kebutuhan aplikasi di *DCCU* dan di *LPU* untuk analisa data lalu lintas. Kontribusi penelitian ini juga menghasilkan skema representasi ciri yang memiliki *time stamp* untuk menyimpan informasi waktu frame terkini. *Time stamp* berguna untuk menghubungkan frame terkini dengan frame berikutnya. Kontribusi penelitian ini juga menghasilkan metode pelacakan kendaraan yang terklasifikasi secara unik karena pemrosesan dasar telah dilakukan oleh *SVS*. Pengujian penelitian ini bertujuan untuk mendemonstrasikan seberapa baik *SVS* bekerja. Ciri-ciri objek lalu lintas yang telah diterima, selanjutnya diuji dengan cara merepresentasi ciri, membuat metode pelacakan kendaraan yang terklasifikasi menggunakan ciri-ciri objek lalu lintas yang berasal dari *SVS*. Pengujian dilakukan pada 3 lokasi yang berbeda di Yogyakarta. Lokasinya adalah: Babarsari, Galeria, dan Condong Catur. Masing-masing menggunakan 1000 frame. Keakuratan tertinggi ditemukan pada objek sepeda motor yaitu di Galeria (90,71%), diikuti oleh objek mobil di Galeria (70,91%), kemudian objek bus di Condong Catur (6,25%), dan terakhir objek truk di Condong Catur (2,44%).

Kata kunci - *Smart Video Sensor*, ekstraksi ciri, objek lalu lintas, skema representasi, pelacakan kendaraan sesuai jenisnya

ABSTRACT

In Yogyakarta and other cities, the Police and the Department of Transportation were still monitoring traffic manually. They have difficulties when some of the traffic problems occur simultaneously. Another difficulty that arise, sometime available video data is incomplete because some video frames was lost. As a result, video data less useful because video data can not be further processed. This research aims to solve the missing video frame problem by modifying the concept of Smart Video Sensor (SVS) Architecture which focuses on creating the Feature Extraction Module and Feature Representation Module and whose features are required for traffic light control, vehicle tracking, tracking congestion and other traffic parameters that can directly be used to analyze traffic data for further processed. The SVS extracts the features of the traffic objects, compresses the video data and sends the results to the Data Center and Control Unit (DCCU) or Local Processing Unit (LPU). SVS consists of Video Sensor Module, Video Compression Module, Feature Extraction Module, Feature Representation Module and Data Transmission Module. The result of this research were the features of traffic objects in the form of eXtensible Markup Language (XML). At DCCU, other systems will receive compressed video data and features to be processed as needed. The contribution of this research was to modify the concept of Smart Video Sensor (SVS) Architecture to solve the missing video frame problem by extracting many features of traffic objects. These features were carefully selected in order to represent objects well. These features come from motor vehicles that got ready to be deployed as per application requirements in Data Center and Control Unit (DCCU) and in Local Processing Unit (LPU) for traffic data analysis. The contribution of this research also resulted in the feature representation scheme that has a time stamp for storing the current frame time information. Time stamp is useful for connecting the current frame with the next frame. The contribution of this study also resulted in uniquely classified vehicle tracking method since basic processing has been performed by SVS. The test of this research aims to demonstrate how well SVS works. The traffic object features that have been received, then tested by representing features, creating classified vehicle tracking method using traffic object features derived from SVS. Tthe tests were conducted at 3 different locations in Yogyakarta. The locations are: Babarsari, Galeria, and Condong Catur. Each uses 1000 frames. The highest accuracy was found on the motorcycles object in Galeria (90,71%), followed by cars object in Galeria (70,91%), then buses object in Condong Catur (6,25%), and last trucks object in Condong Catur (2,44%).

Keyword – Smart Video Sensor, feature extraction, traffic objects, representation scheme, vehicle tracking by type

BAB 1

PENDAHULUAN

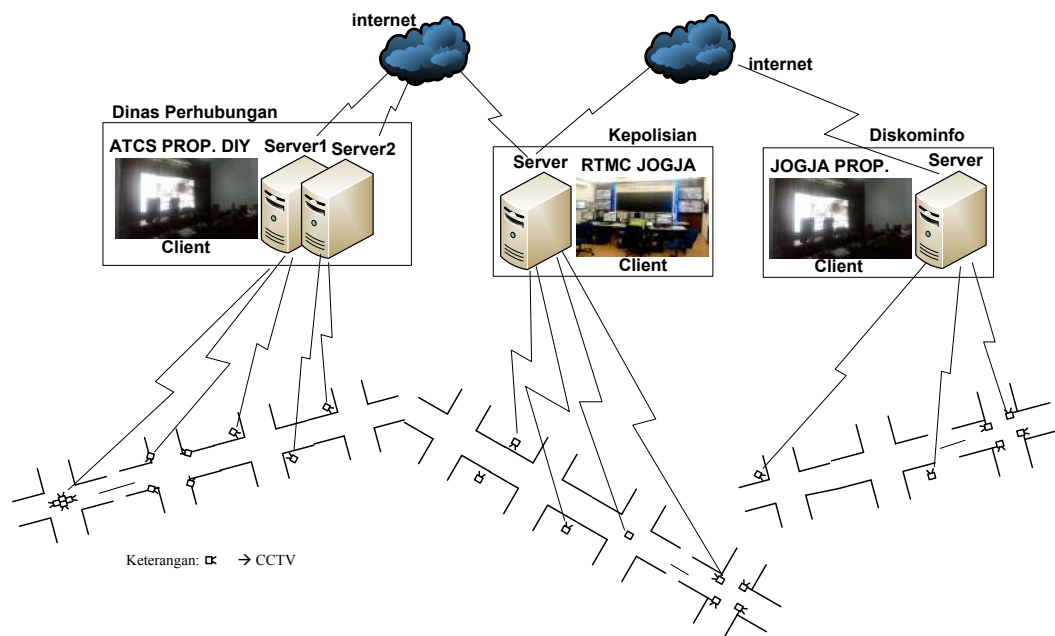
1.1 Latar Belakang

Sejak tahun 2007 (Munawar, 2007, 2008b, 2008a) sampai saat ini, pihak Kepolisian Yogyakarta masih memonitor lalu lintas secara manual. Monitor secara manual artinya operator secara langsung memantau lalu lintas dari semua *Closed Circuit TeleVision* (CCTV) yang ada di Yogyakarta. Pihak Kepolisian dapat memonitor lalu lintas dari berbagai CCTV milik Dinas Perhubungan (Dishub), Dinas Komunikasi dan Informatika (Diskominfo) dan Dinas Lalu Lintas (Dislantas) di ruang *Regional Traffic Management Center* (RTMC). Gambar 1.1 menunjukkan Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta di *Area Traffic Control System* (ATCS) Dishub, JOGJA PROP Diskominfo dan RTMC Yogyakarta yang ada saat ini. Operator Dishub, Diskominfo dan Dislantas (selanjutnya disebut dengan Dinas yang terkait) dapat memonitor lalu lintas dari berbagai CCTV Yogyakarta yang secara *real time* dikirim ke Server. Di ruang RTMC, ATCS dan JOGJA PROP terdapat banyak hasil rekaman video yang di lihat bersamaan. Di ruang tersebut terdapat monitor besar, beberapa Server dan beberapa *Personal Computer* (PC) / *Client* untuk melihat kondisi lalu lintas terkini dari perempatan yang ada di berbagai wilayah Yogyakarta.

Pihak Kepolisian membutuhkan informasi lalu lintas yang berasal dari CCTV Dishub, Diskominfo dan Dislantas untuk melakukan analisa data lalu lintas. Bila pihak Kepolisian membutuhkan data video dari CCTV milik Dishub dan Diskominfo Yogyakarta untuk bukti pelanggaran lalu lintas, bukti peradilan atau bukti kecelakaan maka mereka harus memintanya ke Dishub dan Diskominfo sesuai prosedur kebijakan pimpinan Dishub dan Diskominfo karena kewenangan data video milik masing-masing Dinas yang terkait dan data bersifat rahasia. Selanjutnya, Dishub dan Diskominfo akan mengecek ketersediaan data videonya karena masa simpan video hanya 30 hari saja.

Dinas Perhubungan pada tahun 2012 menyatakan bahwa masalah yang

dihadapi adalah data video sering diterima tidak lengkap/*discontinue* karena banyak frame video yang hilang. Akibatnya data video tidak dapat di olah lebih lanjut. Kegagalan menghadirkan frame-frame penting ini menghambat proses analisis video. Kegagalan ini juga menghambat proses mendapatkan informasi lalu lintas mengenai kondisi lalu lintas terkini, penyebab kecelakaan dan permasalahan lalu lintas yang sering terjadi bersamaan di beberapa tempat yang berbeda. Bukti otentik berupa data video tidak dapat diperoleh secara maksimal untuk proses di pengadilan apabila Kepolisian dan Dinas yang terkait tidak dapat memanfaatkan data video secara maksimal karena data video tidak dapat diolah lebih lanjut. Data video diperlukan untuk mendapatkan informasi yang akurat mengenai gambaran yang menyeluruh di tempat kejadian perkara dan kecepatan kendaraan sesaat sebelum dan sesudah terjadi kecelakaan. Proses ini merupakan proses audit yang memerlukan pelacakan kendaraan.



Gambar 1.1 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta

Penelitian ini bertujuan untuk mengatasi permasalahan frame video yang hilang dengan memodifikasi konsep Arsitektur *Smart Video Sensor* (SVS) yang telah dibuat oleh Sánchez dkk., (2012) dan Ripolles dkk., (2012). Peneliti-peneliti

ini membuat SVS untuk memantau objek dalam skala besar yang berada di dalam gedung. Peneliti-peneliti ini membuat SVS yang dapat mengekstrak objek, membentuk ciri objek dalam XML, melakukan pelacakan objek dan klasifikasi objek secara lokal. Tiap objek mengekstrak 7 ciri. Objek dapat berupa tas, orang atau kumpulan orang. Klasifikasi objeknya memiliki akurasi diatas 80% (Sánchez dkk., 2012; Ripolles dkk., 2012).

Penelitian tentang SVS telah banyak dilakukan. Li dkk., (2004) menyatakan SVS merupakan kamera video yang cerdas yang memanfaatkan prosesor dan perangkat lunak untuk mencapai inteligensi pada tingkat tertentu. SVS dikatakan '*smart*' atau cerdas apabila sistem dapat menganalisis dan memberi informasi yang efisien, cepat dan tepat yang menunjukkan sebuah sistem cerdas. SVS memiliki kemampuan analisa secara otomatis tanpa campur tangan manusia. SVS dapat menghasilkan informasi sesuai kebutuhan. SVS dapat memberi informasi terstruktur dan bermakna yang diperlukan untuk mengolah data dari SVS dan mengirimkan hasilnya ke tempat yang membutuhkan. SVS dapat dengan mudah terintegrasi ke perangkat lain. SVS dapat digunakan secara efisien, interaktif, tepat dan fleksibel untuk dipakai di beberapa aplikasi cerdas lainnya (Belbachir, 2010; Li dkk., 2004). Beberapa SVS dapat digunakan untuk *Smart Video Surveillance System* yang mendeteksi *scene* objek (Li dkk., 2004; Ripolles dkk., 2012; Tomasi dkk., 2014; Sánchez dkk., 2012). Biasanya komponen utama SVS terdiri dari: *video capture*, analisis, saluran komunikasi dan *back-end processing*. Karakteristik kunci dari SVS adalah: (a) sistem dapat memberi informasi terstruktur dan bermakna yang diperlukan untuk mengolah data dari SVS dan mengirimkan hasilnya ke tempat yang membutuhkan, (b) sistem dapat menganalisis data dan menghasilkan informasi sesuai kebutuhan, (c) sistem dapat digunakan secara efisien, interaktif, tepat dan fleksibel untuk dipakai di beberapa aplikasi cerdas lainnya (Li dkk., 2004).

SVS adalah sensor video yang dapat mengekstrak ciri-ciri dari data video dan menghasilkan ciri-ciri objek. Berbeda dengan penelitian sebelumnya yang menggunakan SVS untuk mengekstrak ciri-ciri objek di dalam gedung, pada penelitian ini SVS mengekstrak ciri-ciri objek lalu lintas, memampatkan video

dan mengirimkan hasilnya ke *Data Center and Control Unit (DCCU)* atau *Local Processing Unit (LPU)*. DCCU berfungsi sebagai server atau data center. DCCU juga berfungsi melakukan pengontrolan lalu lintas secara menyeluruh untuk banyak ruas lalu lintas di jalan raya. LPU berada di SVS. LPU berfungsi melakukan pengaturan lama lampu lalu lintas. LPU tidak mengekstrak ciri. Di DCCU, terdapat beberapa aplikasi seperti klasifikasi, pelacakan dan lainnya yang dibutuhkan oleh Pihak Kepolisian. SVS telah melakukan pemrosesan dasar, sehingga sistem lainnya yang ada di DCCU siap menggunakan ciri yang dibutuhkan untuk pengolahan data lebih lanjut.

Di dalam SVS, tiap objek yang di ekstrak cirinya memiliki *time stamp* untuk menyimpan informasi waktu terkini. Informasi waktu disimpan untuk menyelesaikan masalah data video lalu lintas yang tidak lengkap. Dengan *time stamp*, frame terkini akan terhubung dengan frame berikutnya. Data video yang tidak lengkap dapat diatasi dengan memberikan *time frame* sehingga tetap dapat digunakan untuk menghitung *traffic parameter*. Dengan penyelesaian ini, data video dapat diolah lebih lanjut. Selain itu, data *video* yang sudah dimampatkan ukurannya menjadi kecil sehingga data video dapat diterima lebih cepat. Waktu pengolahan data lebih cepat karena aplikasi yang ada di DCCU atau LPU dapat memproses dan menggunakan ciri yang sudah diekstrak sesuai kebutuhan. Pihak terkait memperoleh manfaat SVS karena mereka lebih cepat mendapatkan informasi parameter lalu lintas terkini.

Fokus penelitian ini adalah membangun Modul Ekstraksi Ciri dan Modul Representasi Ciri yang ada di dalam SVS. Awalnya, *scene* merupakan masukan SVS. Selanjutnya, SVS mengekstrak ciri objek, memampatkan data video dan mengirimkan hasilnya ke LPU (aplikasi lokal) atau DCCU (aplikasi lain yang membutuhkan). LPU adalah unit pemrosesan untuk keperluan lokal. DCCU atau LPU menerima data video yang sudah dimampatkan dan ciri objek lalu lintas. Ciri objek ini dapat langsung digunakan oleh DCCU atau LPU sesuai kebutuhan karena tahapan pengolahan data sudah dikerjakan oleh SVS. Ciri objek di sinkronisasi dengan data video yang asli agar dapat dipastikan bahwa ciri objek sesuai dengan data videonya. Data video asli dibutuhkan untuk melihat kondisi

real yang terjadi di jalan raya.

Pengujian penelitian ini bertujuan untuk mendemonstrasikan seberapa baik SVS bekerja. Ciri-ciri objek lalu lintas yang telah di terima, selanjutnya diuji. Caranya dengan merepresentasi ciri, membuat model klasifikasi kendaraan dan model pelacakan kendaraan sesuai jenisnya menggunakan ciri-ciri objek lalu lintas yang berasal dari SVS. Pengujian ini ingin menunjukkan bahwa ciri objek lalu lintas yang berasal dari SVS dapat dimanfaatkan untuk klasifikasi dan pelacakan kendaraan sesuai jenisnya. Hal ini dilakukan untuk memastikan ciri objek lalu lintas yang berasal dari SVS dapat digunakan oleh aplikasi lain.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, secara ringkas dapat disimpulkan bahwa rumusan permasalahan penelitian ini:

- a. Adanya kesulitan mempergunakan data video yang diterima tidak lengkap karena data video tersebut tidak dapat diolah lebih lanjut.
- b. Adanya kesulitan mendapat informasi dari data video bila permasalahan lalu lintas terjadi bersamaan di beberapa tempat yang berbeda.
- c. Adanya kesulitan dalam mendapatkan informasi mengenai kondisi lalu lintas terkini.

1.3 Tujuan Penelitian

Penelitian ini secara rinci memiliki tujuan sebagai berikut:

- a. mengatasi permasalahan frame video yang hilang dengan memodifikasi konsep Arsitektur *Smart Video Sensor* yang fokusnya membuat Modul Ekstraksi Ciri dan Modul Representasi Ciri dan yang ciri-cirinya diperlukan untuk pengontrolan lampu lalu lintas, pelacakan kendaraan, mengatasi kepadatan lalu lintas dan parameter lalu lintas lainnya yang langsung dapat digunakan untuk menganalisa data lalu lintas selanjutnya.
- b. menghasilkan pelacakan kendaraan yang terklasifikasi untuk membuktikan bahwa ciri-ciri objek lalu lintas yang berasal dari *Smart Video Sensor* dapat digunakan dengan baik untuk melacak objek.

1.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

- a. Data video lalu lintas yang diteliti hanya data video yang direkam pada pagi, siang dan sore hari, tidak termasuk kondisi gelap (malam hari).
- b. Data video yang digunakan hanya kendaraan yang menghadap depan dan belakang di beberapa lokasi persimpangan Yogyakarta yaitu Babarsari, Galeria dan Condong Catur.
- c. Objek lalu lintas yang diteliti hanya objek yang bergerak.
- d. Prioritas objek lalu lintas yang diteliti hanya kendaraan bermotor yang meliputi: bus, minibus, sepeda motor, truk.
- e. Fokus penelitian hanya pada Modul Ekstraksi Ciri dan Modul Representasi Ciri dari SVS.

1.5 Manfaat Penelitian

Manfaat penelitian ini:

- a. Pihak yang terkait (pihak kepolisian, Dinas Perhubungan, Dinas Komunikasi dan Informatika, Dinas Lalu Lintas) dapat menggunakan data video yang telah diolah untuk analisa data lalu lintas.
- b. Pihak yang terkait dapat menerima ciri objek lalu lintas dari *Smart Video Sensor* (SVS) yang berbentuk XML untuk digunakan pada aplikasi selanjutnya di platform yang berbeda.
- c. Pihak yang terkait lebih cepat mendapatkan informasi parameter lalu lintas terkini.
- d. Pihak yang terkait dapat memproses dan menggunakan ciri yang sudah diektrak sesuai kebutuhan karena tahapan pemrosesan dasar seperti pra pemrosesan, segmentasi dan ekstraksi ciri telah dikerjakan oleh SVS.
- e. Pihak yang terkait dapat memperoleh informasi tentang klasifikasi dan pelacakan kendaraan yang diperoleh dari hasil perhitungan parameter lalu lintas dan mengetahui kondisi lalu lintas terkini.

1.6 Kontribusi Penelitian

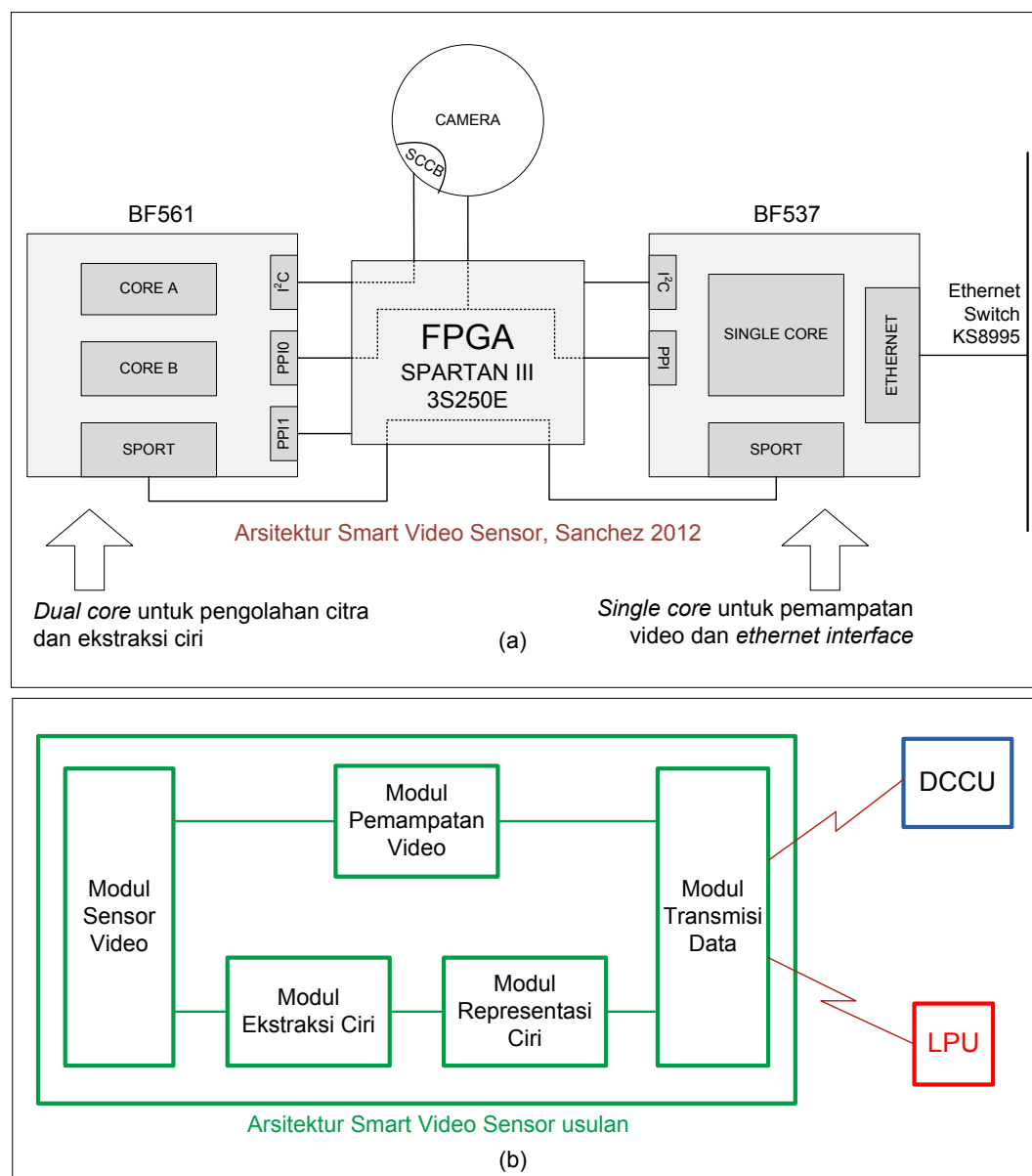
Kontribusi penelitian ini :

- a. memodifikasi konsep Arsitektur *Smart Video Sensor* untuk menyelesaikan masalah frame video yang hilang dengan mengekstrak banyak ciri objek lalu lintas. Ciri-ciri ini dipilih secara hati-hati agar dapat merepresentasikan objek dengan baik. Ciri-ciri ini berasal dari kendaraan bermotor yang siap digunakan sesuai kebutuhan aplikasi di DCCU dan di LPU untuk analisa data lalu lintas.
- b. menghasilkan skema representasi yang memiliki *time stamp* untuk menyimpan informasi waktu frame terkini. *Time stamp* berguna untuk menghubungkan frame terkini dengan frame berikutnya.
- c. menghasilkan pelacakan kendaraan yang terklasifikasi secara unik karena ciri-ciri objek lalu lintasnya yang berasal dari *Smart Video Sensor* (SVS) sudah siap digunakan sehingga hasil pengujiannya lebih cepat diperoleh dibandingkan model klasifikasi dan pelacakan kendaraan biasa. Hal ini dapat terjadi karena dalam tahapan pemrosesan dasar telah dilakukan oleh SVS sehingga waktu pengolahan datanya menjadi lebih singkat.

Gambar 1.2 menunjukkan perbedaan Arsitektur *Smart Video Sensor* antara Sánchez dkk., (2012) dengan Arsitektur *Smart Video Sensor* yang diusulkan. Arsitektur *Smart Video Sensor* Sánchez dkk., (2012) melakukan Ekstraksi Ciri dan Pemampatan Video dalam prosesor yang berbeda. Masing-masing prosesornya memiliki *time stamp* yang dibuat dengan mengganti byte pertama frame dengan *time code* (kode waktu pada frame). *Time stamp* nya berguna untuk kesesuaian waktu antara ciri yang di ekstrak dalam format XML dengan video yang dimampatkan.

Dari Arsitektur *Smart Video Sensor* Sánchez dkk., (2012) diperoleh ide untuk membuat Arsitektur *Smart Video Sensor* yang diusulkan. Komponen Arsitektur SVS yang diusulkan adalah Modul Sensor Video, Modul Pemampatan Video, Modul Ekstraksi Ciri, Modul Representasi Ciri dan Modul Transmisi Data. Pada Arsitektur *Smart Video Sensor* yang diusulkan komponen Modul Ekstraksi Ciri dan Modul Pemampatan Video ditempatkan terpisah agar dapat melakukan

fungsinya masing-masing dengan baik. Modul Ekstraksi Ciri berfungsi mengekstrak ciri-ciri objek tiap frame. Modul Pemampatan Video berfungsi memampatkan video untuk meminimalkan ukuran video. Kemudian, muncullah kebutuhan untuk membentuk ciri dalam format XML dan menyajikan informasinya dalam bentuk yang efisien. Ciri yang diekstrak pada Modul Ekstraksi Ciri perlu dibentuk dalam format XML. Oleh karena itu, dibutuhkan Modul Representasi Ciri.



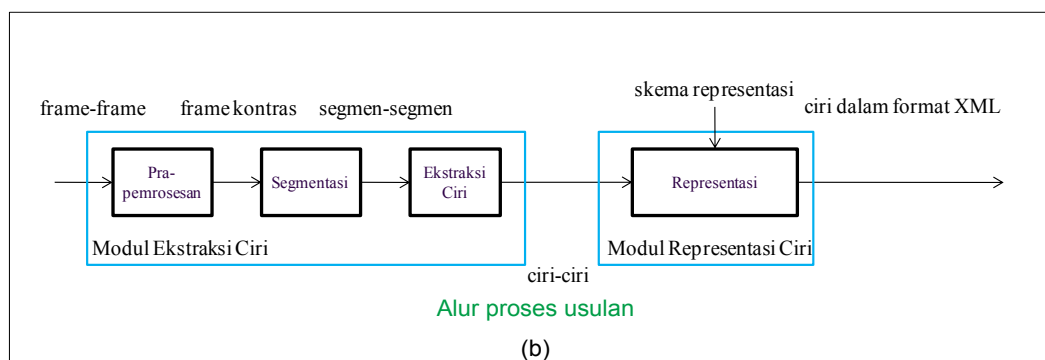
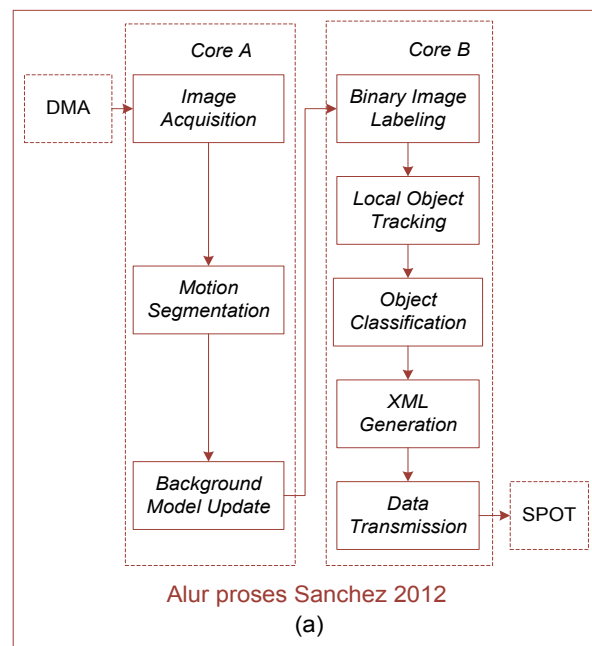
Gambar 1.2 Perbedaan Arsitektur SVS Sánchez dkk. (2012) dan usulan
Ciri dalam format XML perlu informasi yang berisi lokasi video, nama video,

nomor frame, *time stamp* (frame yang berisi informasi waktu frame terkini), nomor objek, nama dan nilai ciri. *Time stamp* penting karena dapat digunakan untuk menghubungkan waktu frame terkini dengan frame berikutnya dan kesesuaian antara ciri yang diekstrak dengan video yang dimampatkan. Karena kebutuhan inilah maka dibuatlah skema representasi. Skema representasi dibuat satu kali saja untuk menjadi acuan dalam membentuk ciri dalam format XML. Ciri yang telah diekstrak dan skema representasi menjadi masukan untuk Modul Representasi Ciri. Modul Representasi Ciri diletakkan setelah Modul Ekstraksi Ciri. Hasil Modul Representasi Ciri adalah ciri dalam format XML yang tiap framenya berisi informasi lokasi video, nama video, nomor frame, *time stamp*, nomor objek, nama dan nilai ciri. Kebutuhan lain adalah mengirimkan data video yang dimampatkan dan ciri dalam format XML. Oleh karena itu dibutuhkan Modul Transmisi Data untuk mengirimkannya. Modul Transmisi Data mengirimkan data video yang dimampatkan dan ciri dalam format XML ke DCCU dan LPU.

Gambar 1.3 menunjukkan perbedaan alur proses yang telah dilakukan oleh Sánchez dkk., (2012) dan alur proses yang diusulkan. Persamaannya adalah melakukan akuisisi objek, melakukan segmentasi dan *background model update*, mengekstrak ciri objek dan mengirimkan hasil ke tempat lain dalam format XML. Perbedaannya, Sánchez dkk., (2012) mengekstrak ciri, melakukan pelacakan dan klasifikasi, membentuk dalam format XML di dalam SVS kemudian hasilnya dikirim ke tempat lain. Sedangkan pada penelitian ini setelah mengekstrak ciri langsung dibentuk ke format XML dan dikirim ke tempat lain (DCCU dan LPU). Pelacakan yang terklasifikasi dilakukan di DCCU.

Keunggulan penelitian ini dibandingkan dengan penelitian Sánchez dkk. (2012) adalah: penelitian Sánchez dkk. (2012) manfaatnya lebih spesifik untuk klasifikasi dan pelacakan obyek yang ada di ruangan. Sedangkan penelitian ini manfaatnya lebih luas yaitu mempersiapkan berbagai kebutuhan Dinas terkait. Ciri-ciri yang diperoleh tidak hanya berguna untuk klasifikasi dan pelacakan kendaraan tetapi dapat memenuhi kebutuhan Dinas terkait seperti mendeteksi kendaraan, menghitung jumlah kendaraan, menghitung kecepatan, menghitung

kepadatan lalu lintas, melakukan verifikasi kendaraan, menghitung percepatan kendaraan dan parameter lainnya yang dibutuhkan oleh Dinas terkait. Bahkan ciri yang diekstrak masih bisa ditambahkan sesuai kebutuhan Dinas terkait. Klasifikasi dan pelacakan pada penelitian ini merupakan beberapa pengujian yang dilakukan untuk membuktikan bahwa ciri-ciri objek dalam format XML yang berasal dari SVS dapat dibaca dan diterima dengan benar. Klasifikasi untuk mengetahui jenis kendaraan. Pelacakan untuk mengetahui posisi kendaraan di waktu tertentu. Waktu dapat diketahui karena ciri-ciri objek tersimpan dalam frame yang juga menyimpan time frame. Oleh karena itu masalah frame video yang hilang dapat terselesaikan.



Gambar 1.3 Perbedaan alur proses Sánchez dkk. (2012) dan usulan

1.7 Sistematika Penulisan

Sistematika penyusunan penulisan penelitian ini meliputi:

Bab 1. Pendahuluan

Bab ini berisi latar belakang penelitian, rumusan masalah penelitian, batasan masalah, tujuan penelitian, manfaat penelitian, kontribusi penelitian dan sistematika penulisan.

Bab 2. Kajian Pustaka

Bab ini berisi penelitian tentang Smart Video Sensor dan ciri.

Bab 3. Landasan Teori

Bab ini berisi teori-teori yang relevan dengan penelitian disertasi.

Bab 4. Metode Penelitian

Bab ini berisi informasi mengenai Gambaran Umum Sistem Pemantauan dan Pengaturan Lalu Lintas, Arsitektur *Smart Video Sensor*, alat dan data penelitian serta pengujian.

Bab 5. Hasil dan Pembahasan

Bab ini memaparkan hasil dan pembahasan tentang Modul Ekstraksi Ciri dan Modul Representasi Ciri yang ada di dalam SVS. Bab ini juga memaparkan hasil dan pembahasan tentang pengujian dari ciri-ciri objek yang telah diterima dengan merepresentasi ciri, membuat model pelacakan dari yang terklasifikasi yang mengikuti Arsitektur SVS sebagai pembuktian bahwa *Smart Video Sensor* telah bekerja dengan baik.

Bab 6. Penutup

Bab ini menyajikan kesimpulan dari hasil penelitian beserta saran yang relevan dengan penelitian disertasi ini.

BAB 2

KAJIAN PUSTAKA

2.1 Penelitian tentang *Smart Video Sensor (SVS)*

Penelitian tentang *Smart Video Sensor (SVS)* telah dibuat oleh beberapa peneliti (Matrella dan Marani, 2011; Sánchez dkk., 2012; Li dkk., 2004; Ripolles dkk., 2012; Tomasi dkk., 2014). Li dkk., (2004) mengusulkan pendekatan terpadu untuk mendeteksi objek bergerak yang dapat digunakan untuk *Smart Video Sensor (SVS)*. Li dkk., (2004) memberikan ilustrasi penjelasan bermakna terkait aliran video mentah yang penting bagi sistem. Peneliti ini juga menyajikan struktur sistem dan konsep pemrosesan video cerdas dengan pendekatan yang efektif untuk mendeteksi objek bergerak dari aliran video berdasarkan template *background* dengan berbagai teknik integrasi untuk mengatasi perubahan iluminasi, gangguan bayangan dan *noise* yang timbul. Peneliti ini menggunakan pendekatan statistik untuk mendapatkan *background model*, yang diperbarui secara *real time* agar bisa menyesuaikan diri terhadap perubahan iluminasi dan objek di tempat kejadian. Peneliti juga menerapkan ambang batas otomatis atau adaptif untuk memisahkan kandidat objek dan latar objek. Peneliti juga menggunakan analisis morfologi dan pengukuran area wilayah yang terhubung untuk menghilangkan gangguan latar objek. Peneliti ini menggunakan ruang warna *Hue, Saturation and Value (HSV)* untuk menghilangkan bayangan dan mendapatkan objek bergerak yang akurat. Peneliti menyimpulkan bahwa lebih banyak fitur, seperti warna, bentuk, tekstur, bahkan fitur audio, yang digunakan bersama dapat memberikan hasil yang lebih akurat. Hasil percobaan dengan pendekatan yang diusulkan menunjukkan kinerja yang memuaskan. Beberapa pustaka penelitian tentang *Smart Video Sensor* yang menjadi acuan penelitian ini dapat ditunjukkan pada Tabel 2.1.

Tabel 2.1 Pustaka penelitian tentang Smart Video Sensor

No	Penulis	Fitur	Metode	Deskripsi	Keterkaitan dengan penelitian
1.	(Li dkk., 2004)	Area, centroid, bounding box, HSV color space	<i>Statistical algorithm</i> untuk background model, <i>HSV color space</i> untuk membedakan antara objek dan bayangan	Paper menjelaskan pendekatan efektif untuk mendeteksi objek bergerak dari <i>video stream</i> berdasarkan <i>background template</i> dengan berbagai teknik yang terintegrasi	referensi penjelasan 'smart' pada SVS
2.	(Sánchez dkk., 2012)	<i>Perimeter, bounding boxes, area, centroid, Extent, Dispersedness, Invers Dispersedness</i>	<i>Motion segmentation</i> dan <i>background update</i> dengan metode <i>background model</i>	Penelitiannya ini mendeteksi orang, kelompok orang dan barang bawaan yang tertinggal dalam gedung.	Konsep SVS dan beberapa fitur pada riset ini menjadi acuan penelitian
3.	(Ripolles dkk., 2014)	<i>Perimeter, bounding boxes, area, centroid, Extent, Dispersedness, Invers Dispersedness</i>	<i>Motion segmentation</i> dan <i>background update</i> dengan metode <i>background model</i>	Penelitian Sánchez dkk., (2012) dikembangkan peneliti ini dalam 3 dimensi	referensi untuk pengembangan kedepan
5.	(Liu dkk., 2013a)	<i>invariant moments, wavelet, size, color, angel, scale</i>	<i>Berbagai metode disampaikan disini, seperti GMM, SURF</i>	Pentingnya sistem dan analisis video cerdas untuk transportasi dan pengawasan	referensi <i>intelligent video system and analysis</i> yang pernah diteliti
6.	(Tomasi dkk., 2016)	<i>Optical flow, corner</i>	<i>Algoritme Lukas dan Kanade, Haris corner detection</i>	Sistem <i>embedded real-time</i> dan <i>smart video sensor</i> untuk estimasi gerakan berdasarkan deteksi dan pelacakan	referensi untuk pengembangan kedepan

Sánchez dkk. (2012) memperkenalkan arsitektur perangkat keras dan perangkat lunak yang fleksibel untuk *Smart Video Sensor*. Sensor yang telah dibuatnya ini telah diterapkan dalam aplikasi pengawasan video. Beberapa *sensor video* dipasang terpisah. Sensor ini merupakan simpul sensor dari sistem

surveilans yang terdistribusi. Dalam sistem yang dibuatnya, *node sensor video* memproses gambar secara lokal untuk mengekstrak dan mengklasifikasi objek. Kemudian, simpul sensor melaporkan hasil pemrosesan ke *node* lain di *Cloud* dalam bentuk deskripsi XML. Arsitektur perangkat keras dari masing-masing *node sensor* telah dikembangkan dalam penelitiannya ini menggunakan 2 prosesor *Digital Signal Processing* (DSP) dan sebuah pengendali *Field-Programmable Gate Array* (FPGA), interkoneksi antar prosesor dan aliran data gambar. Perangkat lunak *node* yang dikembangkan dalam penelitiannya ini didasarkan pada komponen *pluggable* dan berjalan pada *run-time* eksekusi yang disediakan. Proses yang dikembangkan dari penelitiannya meliputi akuisisi, segmentasi, pelabelan, pelacakan, klasifikasi dan ekstraksi ciri. Akuisisinya menggunakan operasi *trigger Direct Memory Access (DMA) receive* untuk memperoleh sebuah citra, yang memiliki tugas melakukan *trigger DMA transmission* dan memberi *cover time stamp* pada tiap citra. *Motion segmentation* dan *background update* pada penelitian Sánchez dkk. (2012) menggunakan *background model*. Untuk mendeteksi objek dengan segmentasi warna pada ruang warna YUV. Pelabelannya menggunakan *component-labelling algorithm* (algoritme pelabelan komponen) dengan *contour tracking technique* (teknik penelusuran kontur). Ciri yang diperoleh pada pelabelan objek adalah *perimeter*, *bounding boxes*, *area* dan *centroid*. Pada pelabelannya ini *noise Gaussian* serta objek yang tidak mencapai ambang batas ukuran tertentu akan disaring. Algoritme pelacakan didasarkan pada *superposition method* (metode superposisi). Klasifikasinya menggunakan *decision tree*. Ciri yang digunakan pada klasifikasi adalah *Extent*, *Dispersedness*, *Inverse Dispersedness*. Hasil awal penelitiannya menunjukkan bahwa sistem dapat mencapai hingga 7,5 frame per detik dalam kasus terburuk, dan tingkat positif klasifikasi objek di semua klas lebih dari 80%.

Ripolles dkk. (2012) mengembangkan penelitian Sánchez dkk. (2012). Ripolles dkk. (2012) memperkenalkan penyelesaian surveilans berbasis 3D untuk infrastruktur besar. Akurasi klasifikasi penelitiannya adalah 92% untuk orang, 81% untuk kelompok orang dan 99% untuk barang bawaan. Pada penelitiannya ini, rekonstruksi 3D mampu mendeteksi klasifikasi palsu secara otomatis dengan

mendeteksi benda yang ukuran atau posisinya tidak tepat. Klasifikasi barang bawaan sangat akurat, walau terkadang bisa salah untuk seseorang atau kelompok. Kelemahannya, beberapa klasifikasi yang keliru akan tetap tidak terdeteksi karena ukuran normal bagi orang atau barang berbeda.

2.2 Penelitian tentang Ciri

Chen dkk. (2009) menggunakan metode *Support Vector Machine* (SVM) untuk melakukan pengenalan dan klasifikasi kendaraan. Pada dasarnya SVM merupakan *classifier* biner, namun SVM dapat diadopsi untuk menangani klasifikasi multi klas. Pengembangan SVM dilakukan dengan menggunakan pendekatan *one-vs-one* (OVO) atau *one-vs-all* (OVA). *Blob* kendaraan yang bergerak di segmentasi menggunakan Algoritme *Gaussian mixture model background subtraction*. Penelitiannya menggunakan 2 tahap Algoritme. Tahap pertama pengenalan warna. Untuk memperkecil kerumitan perhitungan dan tetap tersedianya memory pada implementasi sistem perangkat keras yang *real time*, maka sistemnya menggunakan 8 *histogram* warna sebagai vektor SVM. Ciri vektor untuk kendaraan berwarna yang digunakan pada penelitiannya yaitu *3D colour histogram*. Sistemnya dapat mengenali warna 100% untuk warna hijau, merah, biru, kuning dalam basis datanya. Tahap kedua pengenalan jenis. Ciri untuk pengenalan jenis kendaraan meliputi *size*, *aspect ratio*, *width* dan *solidity*. Ciri ini digunakan agar dapat mengklasifikasi jenis kendaraan: mobil, van dan *Heavy Goods Vehicle* (HGV). Kualitas data dari kamera tidak sebagus yang ada di basis data. Beberapa citra tidak terlihat jelas karena getaran kamera yang diakibatkan oleh angin. Beberapa warna kendaraan berubah karena pantulan dan terangnya sinar matahari. Namun dengan kondisinya itu hasil rata-rata kesensitifan jenis kendaraannya untuk OVO: 0,759 dan OVA: 0,687; hasil rata-rata kespesifikasian jenis kendaraannya untuk OVO: 0,887 dan OVA: 0,858; sedangkan hasil rata-rata kesensitifan dan kespesifikasian warna untuk OVO: 0,956 dan OVA: 0,971.

Chen dan Ellis (2011), Chen dkk. (2012) dan Chen dkk. (2011) menggunakan ciri dan jumlah klas yang sama dalam melakukan klasifikasi kendaraan. Ciri *Measurement-Based Feature* (MBF) yang digunakan adalah *area*,

convex area, perimeter, filled area, equidiameter, width of the bounding box, height of the bounding box, length of major axis, length of minor axis, dispersedness, extent, solidity, eccentricity / eksentrisitas. Klas yang dihasilkan meliputi mobil, van, bus dan sepeda / sepeda motor. Chen dan Ellis, (2011) menggunakan ciri tambahan yaitu *Edge based-Pyramid of Histogram of Orientation Gradients* (EPHOG) dan *Intensity based-Pyramid of Histogram of Orientation Gradients* (IPHOG).

Rashid dkk. (2010) dan Mithun dkk. (2012) melakukan deteksi dan klasifikasi kendaraan melalui video menggunakan metode *time-spatial image* (TSI). Penelitian yang dilakukan oleh Mithun dkk. (2012) merupakan pengembangan dari penelitian yang dilakukan oleh Rashid dkk. (2010), dimana Mithun dkk. (2012) menggunakan metode *multiple time-spatial image* (MTSI) sedangkan Rashid dkk. (2010) hanya menggunakan *single time-spatial image* (STSI). Perluasan yang dilakukan oleh Mithun dkk. (2012) dilakukan untuk mengurangi kesalahan deteksi yang terutama disebabkan oleh kemacetan dan meminimalkan terjadinya kesalahan klasifikasi. Akurasi deteksi dan klasifikasi kendaraan dengan metode MTSI meningkat diatas 5% dibanding dengan metode STSI. Mithun dkk. (2012) melakukan 2 langkah untuk klasifikasi kendaraan. Pertama, kelas dibagi sesuai dengan jumlah rodanya seperti kendaraan roda dua (2W), roda tiga (3W), roda empat (4W), roda enam (6W) dengan menggunakan ciri *shape* (bentuk) dan *size* (ukuran). Kemudian skema klasifikasi diadopsi untuk menemukan jenis kendaraan tertentu seperti mobil atau jeep di kelas 4W, dan bus atau van yang ada di kelas 6W menggunakan ciri *shape-invariant* (variasi bentuk) dan tekstur. Rashid dkk. (2010) menggunakan 7 ciri dalam mendeteksi dan mengklasifikasi kendaraan yang meliputi: lebar, area, *compactness*, rasio panjang-lebar, *rasio major axis-minor axis*, *rectangularity* dan *solidity*. Sedangkan Mithun dkk. (2012) mengurangi penggunaan ciri *solidity* sehingga ciri yang digunakan hanya 6 ciri. Rashid dkk. (2010) mengklasifikasi kendaraan menjadi 5 kelas meliputi: kelas A (sepeda atau motor), kelas B (becak atau van), kelas C (bajaj), kelas D (mobil), dan kelas E (mikrobus atau leguna). Sedangkan Mithun dkk., (2012) mengklasifikasi kendaraan menjadi 7 tipe meliputi: tipe I (sepeda motor),

tipe II (becak), tipe III (bajaj), tipe IV (mobil), tipe V (jip atau minivan), tipe VI (van tertutup), dan tipe VII (bus).

Fazli dkk. (2012) melakukan penelitian menggunakan metode *Neural Network Multi Layer Perceptron (MLP) with two hidden layer* untuk menentukan jenis kendaraan. Ada 6 ciri yang digunakan meliputi: *length* / panjang, *width* / lebar, *area*, *diameter*, *total area of object* / total area objek, *ratio of distance between the centre of object and length main axis* / rasio jarak antara pusat objek dan poros utama. Hasil klasifikasi kendaraan diperoleh 3 klas: kendaraan berat, kendaraan ringan dan sepeda motor. Hasil akurasi sistemnya lebih dari 90%.

Liu dkk. (2013) melakukan survei tentang deteksi dan pelacakan kendaraan yang objeknya dibedakan berdasarkan 3 klas ciri yaitu warna, tekstur dan bentuk. Rata-rata pendekatan yang mewakili metode yang digunakan pada ciri visual pada kendaraan yaitu simetri, warna tepi, kontur, tekstur, bayangan dan bagian dari mobil seperti lampu kendaraan, kaca, ban. Hasil dari deteksi, pengenalan pola dan pelacakan dapat dianalisa untuk mengekstraksi parameter arus lalu lintas seperti tipe kendaraan, jumlah kendaraan, kepadatan kendaraan, kecepatan rata-rata dan informasi kecelakaan lalu lintas. Tabel 2.2 menunjukkan pustaka penelitian tentang ciri objek lalu lintas.

Tabel 2.2 Rangkuman pustaka penelitian tentang ciri objek lalu lintas

No.	Penulis	Ciri	Metode	Kelas kendaraan	Hasil
1.	Matos dan Souza, (2012)	median jumlah tepi, blok PCA, radius pemisah	KNN adaptif	Sepeda motor, mobil, bus, truk	95%
2.	Rashid dkk., (2010); Mithun dkk., (2012)	Lebar, area, <i>compactness</i> , rasio panjang dan lebar, rasio major axis dan minor axis, <i>rectangularity</i>	<i>Multiple time spatial image</i>	Sepeda motor, becak, bemo, sedan, toyota, bus, truk	97%
3.	Kafai dan Bhanu, (2012a)	Sudut, tinggi, lebar, jarak	<i>Dynamic Bayesian Network</i>	Sedan, pickup truck, SUV/minivan, tidak diketahui	98,2% dari 845

Tabel 2.2 (lanjutan)

No.	Penulis	Ciri	Metode	Kelas kendaraan	Hasil
4.	Fazli dkk., (2012)	panjang, lebar, area objek, luas objek, diameter lingkaran, rasio jarak antara pusat objek dan poros utama panjang	berbasis <i>neural network multi layer perceptron</i> (MLP) dengan 2 lapisan tersembunyi	kendaraan berat, kendaraan ringan dan sepeda motor	90% dari 182
5.	Daigavane dkk., (2011)	Panjang, lebar, tinggi, ukuran, kecepatan	<i>Neural Network</i>	Kecil, sedang, besar	90%
6.	Rashid dkk., (2010)	Lebar, area, <i>compactness</i> , rasio panjang dan lebar, rasio major axis dan minor axis, <i>rectangularity</i> , <i>solidity</i>	<i>Time spatial image</i> (TSI)	Sepeda motor, becak, bemo, sedan, toyota, bus, truk	91%
7.	Daya dkk., (2010)	jarak, permukaan, rasio dan laju kendaraan	<i>Artificial Neural Network</i> (ANN)	9 klas	95% dari 270 citra
8.	Xiong dan He, (2009)	bentuk, panjang, tinggi, lebar	<i>Wavelet transform</i>	1) sepeda motor, 2) mobil dan mobil dengan 1 axle trailer, van/pick up, van/pick up dengan 1 axle trailer, 3) bus dan 2 atau 3 axle truk, 4) 2 atau 3 axle semi dengan 1 atau 2 axle trailer, 6 atau 7 axle semi dengan single trailer, 5) >5-axle semi dengan multiple trailer.	90% dari 341
9.	Shaoqing dkk., (2009)	<i>the number of regions</i> (jumlah region), <i>the number of colors</i> (jumlah warna), <i>big window</i> , <i>low-gray region</i> (region abu-abu muda), <i>the edges number of the top part</i> (jumlah tepi bagian atas)	<i>Fuzzy rules classifier</i> yaitu kombinasi metode <i>fuzzy</i> dan aturan untuk <i>non</i> kendaraan	Mobil, truk, bus,	95% dari 438
10.	Messelodi dan Maria, (2007)	Kecepatan rata-rata, <i>region</i>	3D <i>model classifier</i>	Sepeda, sepeda motor, mobil, van, truk, bus kota, bus antar kota	96% dari 25

Tabel 2.2 (lanjutan)

No.	Penulis	Ciri	Metode	Kelas kendaraan	Hasil
11.	Morris dan Trivedi, (2008)	<i>area</i> , luas, <i>compactness</i> , elongasi, <i>perimeter</i> , cembung lambung <i>perimeter</i> , <i>bounding box</i> , parameter elips yang paling cocok, kekasaran, centroid	<i>weighted k-nearest neighbor (wkNN) classifier</i>	sedan, pickup, SUV, van, merged, sepeda motor, truk, kontraktor	-
12	Imelda, I, Harjoko, A, Nurwanto, P (2018)	LBP, area ROI, aspect ratio	<i>K-Nearest Neighbor</i>	Bus, minibus, motor dan truk	90,71 % untuk motor di Galeria

Ciri digunakan untuk membedakan antara obyek yang satu dengan objek lainnya (Hu dkk., 2011). Untuk mempermudah mengetahui penggunaan ciri maka ciri dikategorikan berdasarkan penggunaannya yaitu ciri yang digunakan untuk deteksi kendaraan, ciri yang digunakan untuk klasifikasi kendaraan dan ciri yang digunakan untuk pelacakan kendaraan. Pengelompokan ciri ini ditunjukkan pada Tabel 2.3.

Tabel 2.3 Pengelompokan ciri berdasarkan kategori aplikasi

Kategori aplikasi	Ciri	Metode	Referensi
Deteksi kendaraan	<i>Blob</i>	<i>Background Gaussian Mixture Model (GMM) dan shadow removal method</i>	Chen dkk., (2012)
	<i>Edge, gradient</i>	<i>Color contour algorithm, gradient based contour algorithm</i>	Ambardekar, (2007)
	<i>Edge, blob</i>	<i>Vertical edge detection and match filtering</i>	Kafai dan Bhanu, (2012)

Tabel 2.3 (lanjutan)

Kategori aplikasi	Ciri	Metode	Referensi
Deteksi plat nomor kendaraan	1) <i>perpendicular distance from license plate centroid to a line connecting two tail light centroids</i> ; 2) <i>right tail light width</i> ; 3) <i>left tail light width</i> ; 4) <i>right tail light-license plate angle</i> ; 5) <i>left tail light-license plate angle</i> ; 6) <i>right tail light-license plate distance</i> ; 7) <i>left tail light-license plate distance</i> ; 8) <i>bounding box width</i> ; 9) <i>bounding box height</i> ; 10) <i>license plate distance to bounding box bottom side</i> ; 11) <i>vehicle mask area</i>	<i>Blob detection and match filtering method</i>	Kafai dan Bhanu, (2012)
Klasifikasi kendaraan	<i>Measurement Based Feature (MBF), Intensity pyramid-based Histogram of Gradient (IPHOG)</i>	<i>Support Vector Machine (SVM)</i>	Chen dkk., (2012)
	<i>Blob</i>	<i>Color contour algorithm, gradient based contour algorithm</i>	Ambardekar, (2007)
	<i>LBP, area ROI, aspect ratio</i>	<i>K_Nearest Neighbor</i>	Pemodelan Ekstraksi Ciri pada SVS
Pelacakan kendaraan	<i>Blob, centroid location dan velocity</i>	<i>Kalman filter</i>	Chen dkk., (2012)
	<i>Blob</i>	<i>Tracking algorithm</i>	Ambardekar, (2007)
	<i>Centroid dan bounding box</i>	<i>Kalman Filter</i>	Pemodelan Ekstraksi Ciri pada SVS

BAB 3

LANDASAN TEORI

3.1 *Smart Video Sensor*

Smart Video Sensor dibuat berdasarkan tuntutan kebutuhan teknologi dan trend pasar terkini. Pernyataan ini diperkuat oleh penelitian yang dibuat oleh Liu dkk. (2013). Liu dkk. (2013) menyatakan bahwa teknologi dan trend pasar terkini telah menuntut kebutuhan yang signifikan akan adanya solusi yang layak untuk sistem dan analitik video / kamera. Liu dkk. (2013) melakukan survei mengenai *Intelligent Video Systems and Analytics*. Aspek utama yang harus dimiliki oleh *Intelligent Video Systems and Analytics* meliputi:

- a. sistem memiliki kemampuan aplikasi yang *real time* dan dapat digunakan untuk perhitungan yang kompleks,
- b. sistem memiliki kemampuan yang fleksibel dan handal dalam lingkungan dan cuaca yang berbeda,
- c. sistem memiliki kemampuan tingkat efisiensi dan akurasi yang tinggi,
- d. sistem memiliki kemampuan untuk dapat mengumpulkan data yang berasal dari *smart camera* di banyak kota,
- e. sistem memiliki standar dan kinerja yang dapat di evaluasi per periode,
- f. sistem dapat disimpan di *cloud computing* agar data video dengan mudah dapat dicari dan digunakan untuk *intelligent system* kota yang lebih besar,
- g. sistem lebih *intelligent* dari pengolahan video otomatis yang pernah ada.

Belbachir (2010) menyatakan ada 3 karakteristik penting dan praktis yang dinyatakan oleh *Automated Imaging Association (AIA)* tentang *smart camera*:

- a. kamera dapat terintegrasi ke perangkat lain dengan mudah.
- b. kamera dapat memanfaatkan prosesor dan perangkat lunak untuk mencapai tingkat komputasi intelijen tertentu.
- c. kamera memiliki kemampuan analisa secara otomatis oleh komputer tanpa campur tangan manusia.

Selanjutnya, AIA mengkategorikan *vision system* dalam 3 kategori produk:

- a. *Smart camera*, seperti sebuah *vision system* yang lengkap atau hampir lengkap, yang terdapat di dalam kamera itu sendiri.
- b. *Vision sensor*, terutama sebagai sebuah *lower end smart camera* yang kinerjanya hampir sama dengan *high end smart camera*.
- c. *Embedded vision processor*, dimana kameranya terkait menjadi satu dengan *external computing*.

Beberapa peneliti menyatakan pendapatnya tentang *Smart Video Sensor*. Li dkk. (2004) menyatakan *Smart video sensor* dikatakan “*smart*” apabila sebuah sistem dapat menganalisis dan memberi informasi yang efisien, cepat dan tepat yang menunjukkan sebuah sistem cerdas. Karakteristik kuncinya adalah:

- a. Sistem dapat memberi informasi terstruktur dan bermakna yang diperlukan untuk mengolah data mentah video dan mengirimkan hasilnya ke tempat yang membutuhkan.
- b. Sistem dapat menganalisis data dan menghasilkan informasi sesuai kebutuhan.
- c. Sistem dapat digunakan secara efisien, interaktif, tepat dan fleksibel untuk dipakai di beberapa aplikasi cerdas lainnya.

Smart Video Sensor dapat digunakan untuk mendeteksi *scene* objek untuk *smart video surveillance system*. *Smart video sensor* yang dibuat oleh Li dkk. (2004), komponen utamanya terdiri dari:

- a. *Video capture*. Gambar yang diambil dari *scene* dan objek nyata ditransformasikan ke dalam sinyal video atau citra digital yang berurutan.
- b. Analisis. Objek yang bergerak dideteksi langsung dari video dan cirinya diekstrak dengan algoritme yang dijalankan secara *real time*. Keluarannya yaitu data video yang sudah dimampatkan dalam bentuk bit dan hasil ekstraksi ciri video berikut penjelasannya.
- c. Saluran komunikasi. *Bandwidth* yang lebar diperlukan untuk mentransmisikan video dan aliran deskripsi yang terkait dari *front-end* ke *back-end*. Saluran ini mungkin sangat pendek sebagai bus internal untuk mengintegrasikan semua komponen secara bersamaan, atau sebagai saluran komunikasi jarak jauh

untuk mengirimkan data *smart video* ke *back-end* dengan pengkodean berbasis konten dan terukur.

- d. *Back-end processing*. Pemrosesan tingkat tinggi dan pengambil keputusan bertindak atau bereaksi sesuai *description stream*. *Description* yang diterima oleh komponen ini dapat disimpan dalam sebuah memori lokal, atau sebuah basis data melalui sebuah *interface* sehingga data yang dikumpulkan dapat digunakan lebih lanjut untuk mendapatkan *pattern* dan *knowledge video*.

Beberapa *Smart Video Sensor* bila disatukan merupakan *Video Sensor Network*. Penelitian tentang *key issue* dalam *Video Sensor Network* menyatakan bahwa sensor kamera mengumpulkan data video yang kaya dengan informasi dan menawarkan potensi yang luar biasa bila digunakan dalam jaringan sensor nirkabel (Ren dan Yang, 2010).

3.2 Objek Lalu Lintas

Objek lalu lintas adalah objek yang ada di area/ruang lalu lintas. Ruang Lalu Lintas Jalan menurut Undang – Undang Republik Indonesia Nomor 22 Tahun 2009 Bab 1 pasal 1 adalah prasarana yang diperuntukkan bagi gerak pindah kendaraan, orang, dan/atau barang yang berupa jalan, angkutan dan fasilitas pendukung. Angkutan yaitu kendaraan yang digunakan untuk memindahkan orang/barang dari satu tempat ke tempat lain (Kemenhub, 2009). Penyelenggaraan lalu lintas dan angkutan jalan dibutuhkan jaringan lalu lintas yang saling terhubung agar objek lalu lintas dapat sampai ke tempat tujuan dengan aman, selamat, tertib dan lancar (Kemenhub, 2013).

Objek lalu lintas meliputi kendaraan dan non kendaraan (Kemenhub, 2009). Kendaraan adalah suatu sarana angkut di jalan yang terdiri atas kendaraan bermotor dan kendaraan tidak bermotor. Kendaraan bermotor adalah setiap kendaraan yang digerakkan oleh peralatan mekanik berupa mesin selain kendaraan yang berjalan di atas rel. Kendaraan bermotor dikelompokkan berdasarkan jenisnya yaitu: sepeda motor, mobil penumpang, mobil bus dan mobil barang (Kemenhub, 2014). Sepeda motor adalah kendaraan bermotor beroda dua dengan atau tanpa rumah-rumah dan dengan atau tanpa kereta samping atau

kendaraan bermotor beroda tiga tanpa rumah-rumah. Kendaraan tidak bermotor adalah setiap kendaraan yang digerakkan oleh tenaga manusia dan/atau hewan. Kendaraan tidak bermotor dikelompokkan dalam kendaraan yang digerakkan oleh tenaga orang dan kendaraan yang digerakkan oleh tenaga hewan. Kendaraan yang digerakkan oleh tenaga orang meliputi sepeda, becak dan gerobak. Kendaraan yang digerakkan oleh tenaga hewan salah satunya dokar. Objek non kendaraan meliputi manusia, hewan dan tumbuhan. Manusia atau orang yang berada di lalu lintas yaitu pejalan kaki, pengemudi, penumpang. Pejalan kaki adalah setiap orang yang berjalan di ruang lalu lintas jalan. Pejalan kaki dapat berjalan di lalu lintas seorang diri atau berkelompok. Pejalan kaki bisa berjalan menyebrang jalan baik di jembatan penyebrangan atau di *zebra cross*. Pejalan kaki bisa juga berjalan menyusuri jalan diatas trotoar. Hewan atau binatang yang berada di lalu lintas yaitu hewan yang melintas di jalan raya (Kemenhub, 2009). Hewan kadang menyebrang sendiri atau berkelompok secara tiba-tiba. Hewan itu bisa juga digunakan sebagai tenaga untuk menggerakkan kendaraan tidak bermotor. Tumbuhan yang ada di area lalu lintas tumbuh dan berkembang. Tumbuhan ini dipelihara agar menambah asri jalan. Namun tumbuhan kadang menutupi rambu lalu lintas. Oleh karena itu, daun dari tumbuhan yang menutupi rambu lalu lintas atau yang sudah mulai masuk ke area lalu lintas harus dipangkas agar tidak mengganggu aktivitas di jalan raya. Kadang hujan deras disertai petir dan halilintar menyebabkan pepohonan di pinggir jalan tumbang dan jatuh menimpa kendaraan yang sedang bergerak maupun kendaraan yang diam.

3.3 Pengolahan Citra Digital

Sebuah citra disebut citra digital jika citra dapat didefinisikan sebagai sebuah fungsi dua dimensi, $f(x,y)$, dimana x dan y adalah koordinat bidang datar, berhingga (*finite*), dan nilainya diskrit. Nilai fungsi f di setiap pasangan koordinat (x,y) disebut intensitas atau level keabuan (*grey level*) dari citra di titik itu. Sebuah citra digital terdiri dari sejumlah elemen berhingga, dimana masing-masing citra memiliki lokasi dan nilai tertentu. Elemen-elemen yang bentuknya seperti sebuah array digital disebut *picture elements*, *image elements*, *pels*, atau piksel-piksel

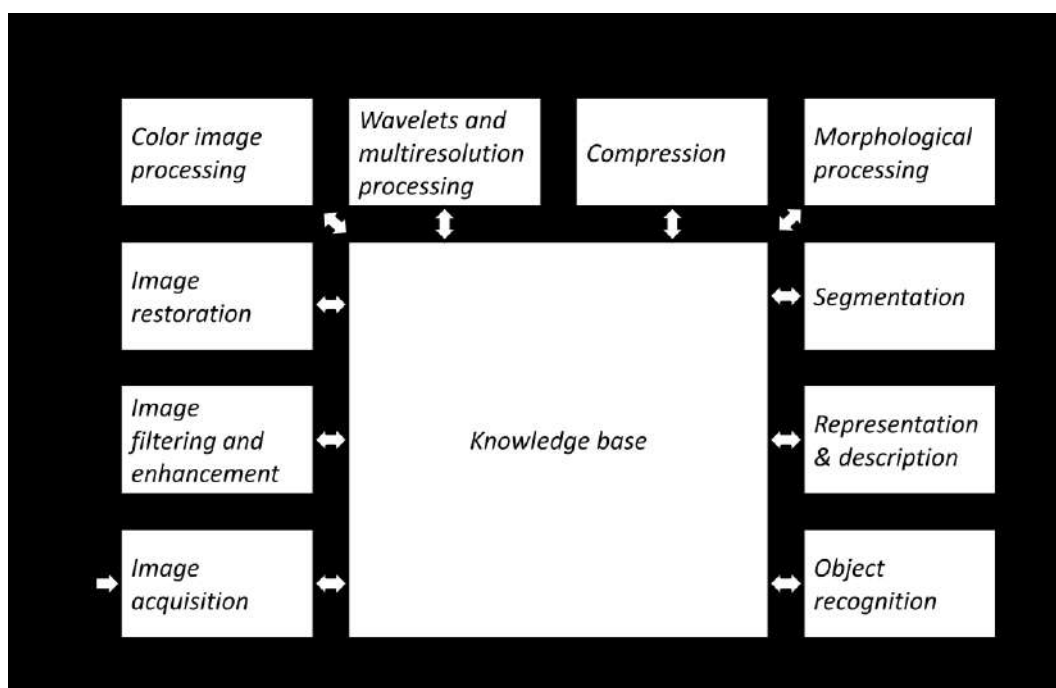
(*pixels*). *Pixel* adalah istilah yang digunakan paling banyak untuk menunjukkan unsur-unsur citra digital (Gonzalez dkk., 2008). Citra digital dibentuk oleh kumpulan titik. Kumpulan titik ini dinamakan piksel. Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi (Kadir dan Susanto, 2013). Sebuah piksel mempunyai koordinat berupa (x, y) , dimana x menyatakan posisi kolom, y menyatakan posisi baris dan piksel pojok kiri-atas mempunyai koordinat $(0, 0)$ dan piksel pada pojok kanan - bawah mempunyai koordinat $(N-1, M-1)$.

Tujuan pengolahan citra digital untuk menyampaikan gagasan bahwa semua metode dapat diterapkan pada citra untuk tujuan yang berbeda. Tahapan dasar pengolahan citra digital ditunjukkan pada Gambar 3.1. Tahapan pengolahan citra digital secara berurutan: *image acquisition* (akuisisi citra), *image filtering and enhancement* (peningkatan kualitas citra), *image restoration* (restorasi citra), *color image processing* (pengolahan citra berwarna), *wavelets and multiresolution processing* (pengolahan *wavelet* dan multi resolusi), *compression* (pemampatan citra), *morphological processing* (pengolahan morfologi), *segmentation* (segmentasi), *representation & description* (representasi dan deskripsi), *object recognition* (pengenalan objek). Masing-masing tahapan menjadi masukan *knowledge base* yang secara timbal balik dapat berguna bagi tahapan itu sendiri. Secara umum hasil keluaran dari *image acquisition*, *image filtering and enhancement*, *image restoration*, *color image processing*, *wavelets and multiresolution processing*, *compression*, *morphological processing* adalah citra. Sedangkan hasil keluaran dari *segmentation*, *representation & description*, *object recognition* adalah atribut citra (Gonzalez dkk., 2008).

Image acquisition (akuisisi citra) atau pembentukan citra yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) merupakan proses pengambilan citra dari kamera yang diubah menjadi entitas yang dapat dikelola. Proses akuisisi citra terdiri dari tiga tahap. Pertama, kamera menangkap cahaya atau energi yang dipantulkan dari objek. Kedua, sebuah sistem optik/sensor memfokuskan cahaya atau energi. Ketiga, Cahaya atau energi diteruskan oleh sensor untuk diubah menjadi citra digital. Pada proses ini ditentukan format data dan media

penyimpanannya.

Image filtering and enhancement (peningkatan kualitas citra) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) adalah proses manipulasi sebuah citra yang hasilnya lebih bagus dari citra aslinya yang dapat digunakan untuk aplikasi khusus. Kata khusus menjadi penting karena hasil yang dicapai menggunakan teknik peningkatan kualitas sesuai kebutuhan. Pemilihan metode yang digunakan untuk meningkatkan kualitas citra sangat penting dan memberikan hasil yang berbeda.



Gambar 3.1 Tahapan dasar Pengolahan Citra Digital (Gonzalez dkk., 2008)

Image restoration (restorasi citra) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) adalah proses untuk membuat kualitas citra turun. Ini terjadi akibat adanya tambahan derau. Tujuan restorasi citra adalah agar lebih mirip dengan keadaan aslinya. Sifat *image restoration* (restorasi citra) adalah objektif, artinya teknik restorasi cenderung dibuat berdasarkan pada model matematika atau probabilitas untuk degradasi citra atau penurunan kualitas citra. Ini sangat bertolak belakang dengan *image enhancement*. *Image enhancement* adalah peningkatan kualitas citra. Sifat *image enhancement* adalah subyektif,

artinya tekniknya didasarkan pada preferensi subyektif manusia mengenai hasil peningkatan kualitas citra yang dianggap “baik”.

Color image processing (pengolahan citra berwarna) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) merupakan area yang penting karena peningkatan citra digital yang semakin meningkat khususnya melalui internet. Warna digunakan untuk mengekstraksi ciri yang menarik di dalam sebuah citra. Penggunaan warna dalam pengolahan citra dimotivasi oleh dua faktor utama. Pertama, warna adalah *descriptor* kuat yang dapat menyederhanakan pengidentifikasian objek dan ekstraksi dari sebuah *scene*. Kedua, manusia dapat membedakan ribuan nuansa warna dan intensitas, dibandingkan hanya sekitar dua lusin warna abu-abu. Faktor kedua ini sangat penting dalam analisis citra manual.

Wavelets yang ditunjukkan pada Gambar 3.1 menurut Burrus dkk., (1998) merupakan alat analisis yang biasa digunakan untuk menyajikan data atau fungsi atau operator ke dalam komponen-komponen frekuensi yang berlainan, dan kemudian mengkaji setiap komponen dengan suatu resolusi yang sesuai dengan skalanya. *Wavelet* merupakan gelombang mini (*small wave*) yang mempunyai kemampuan mengelompokkan energi citra dan terkonsentrasi pada sekelompok kecil koefisien, sedangkan kelompok koefisien lainnya hanya mengandung sedikit energi yang dapat dihilangkan tanpa mengurangi nilai informasinya.

Compression (pemampatan citra) atau kompresi yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) berguna untuk mengurangi ukuran file data citra agar saat data citra disimpan hanya membutuhkan lebih sedikit ruang memori tempat penyimpanan dan saat citra dikirim hanya membutuhkan sedikit *bandwidth*. Meskipun teknologi penyimpanan semakin meningkat secara signifikan dalam dekade terakhir, tetapi kapasitas transmisi tidak meningkat. Oleh karena itu pemampatan citra sangat dirasakan manfaat dan pentingnya terutama dalam penggunaan internet, apalagi jika berisi data citra penting yang perlu dikirimkan melalui internet.

Morphological processing (pengolahan morfologi) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) berkaitan dengan *tools* untuk

komponen ekstraksi citra yang sangat berguna dalam representasi dan deskripsi bentuk. Operasi morfologi merupakan operasi pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Lubang pada objek dapat ditutup melalui operasi morfologi. Objek yang saling berhimpitan dapat dipisahkan melalui morfologi.

Segmentation (segmentasi) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) memiliki prosedur untuk membagi citra menjadi bagian-bagian yang lebih kecil berbentuk segmen. Segmen adalah piksel-piksel kelompok piksel. Segmen-segmen ini terpisah di beberapa tempat. Umumnya segmentasi bila berdiri sendiri merupakan bagian tersulit dalam pengolahan citra digital. Segmentasi membutuhkan identifikasi secara individu. Apabila algoritme segmentasi lemah akhirnya dapat mengakibatkan kegagalan. Tingkat akurasi segmentasi harus tinggi untuk menjamin keberhasilan segmentasinya.

Representasi yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) berarti mengkonversi data ke bentuk yang sesuai untuk pemrosesan komputer yang diperlukan. Keputusan pertama yang harus dibuat adalah apakah data harus direpresentasikan sebagai batas atau sebagai wilayah yang lengkap. Representasi batas cocok digunakan ketika fokusnya adalah pada karakteristik bentuk eksternal, seperti sudut dan infleksi. Representasi wilayah cocok digunakan ketika fokusnya adalah pada sifat internal, seperti tekstur atau bentuk tulang. Pemilihan sebuah representasi hanyalah salah satu bagian dari solusi untuk mengubah data awal menjadi bentuk yang sesuai untuk pemrosesan komputer berikutnya. Sebuah metode juga harus ditentukan secara spesifik untuk menggambarkan data sehingga terjadi transformasi ciri citra data awal menjadi bentuk yang sesuai untuk pemrosesan komputer berikutnya. Sebuah metode juga harus ditentukan untuk menggambarkan data sehingga ciri yang menarik ditandai dengan tanda khusus. Deskripsi, juga disebut *feature selection* (seleksi ciri), yang berkaitan dengan mengekstrak atribut agar informasi kuantitatif yang dibutuhkan semakin memiliki arti yang penting atau menjadi dasar untuk membedakan satu kelas objek dari yang lain. *Representation & description* (representasi dan deskripsi) hampir selalu mengikuti keluaran dari tahap segmentasi, yang biasanya

merupakan data mentah piksel, yang merupakan salah satu batas suatu wilayah objek (yaitu himpunan piksel yang memisahkan satu daerah dari citra yang lain) atau semua titik di wilayah itu sendiri.

Object recognition (pengenalan objek) yang ditunjukkan pada Gambar 3.1 menurut Gonzalez dkk., (2008) adalah proses yang menetapkan label (misalnya "kendaraan") ke objek berdasarkan deskripsinya. Pengenalan objek juga mencakup pengolahan citra digital dengan pengembangan metode untuk pengenalan objek secara individu.

3.4 Ciri

Ciri adalah sifat atau tanda yang dimiliki oleh objek untuk membedakan dari objek yang lain. Ciri objek merupakan karakteristik yang melekat pada objek (Kadir dan Susanto, 2013). Ciri yang baik memiliki keunikan, stabil dan *independent*. Mingqi dkk. (2008) menjelaskan bahwa ciri bentuk yang efisien harus memiliki sifat esensi meliputi:

- a. *Identifiability*/kemampuan untuk mengidentifikasi objek: bentuk yang mirip menurut penglihatan manusia dapat dinyatakan berbeda oleh ciri.
- b. *Translation, rotation and scale invariance*/tidak dipengaruhi oleh variasi translasi, rotasi dan skala: perubahan lokasi, rotasi dan skala dari bentuk objek yang sama tidak mempengaruhi ekstraksi cirinya sehingga tetap dikenali sebagai satu objek.
- c. *Affine invariance*/tahan terhadap transformasi *affine*. Transformasi *affine* bisa dibentuk dari translasi, skala dan rotasi yang bisa diterapkan secara bergantian. Ciri yang diekstrak nilainya harus tetap. Jika nilai ciri yang diekstrak berubah, nilai variannya harus kecil ketika objek mengalami transformasi *affine*.
- d. *Noise resistance*/tahan terhadap derau: ciri harus kuat menghadapi derau. Ciri tetap dapat dipakai mengenali objek walaupun banyak objek dari citra yang diuji.
- e. *Occultation invariance*: ketika beberapa bentuk dari bagian objek tertutupi oleh objek yang lain, ciri yang tersisa tidak berubah saat dibandingkan dengan bentuk aslinya.

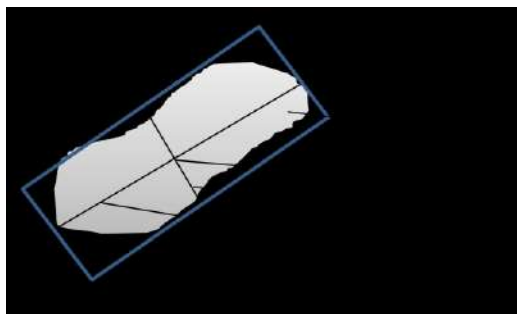
- f. *Statistically independent*/tidak memiliki ketergantungan terhadap apapun secara statistik.
- g. *Reliable*/handal: sama seperti pola, ciri yang diektrak harus sesuai dengan objeknya.

Ciri dibedakan antara *low level feature* (ciri tingkat rendah) dan *high level feature* (ciri tingkat tinggi). Menurut Tamrakar dkk. (2012); Chan dan Vasconcelos (2012), ciri tingkat rendah adalah ciri yang sudah tidak dapat diturunkan. Menurut Nixon dan Aguado (2008), ciri tingkat rendah menjadi ciri dasar yang dapat diambil secara otomatis dari citra tanpa menambahkan informasi bentuk apapun. Fungsi ciri tingkat rendah adalah sebagai informasi untuk analisis selanjutnya (Nixon dan Aguado, 2008). Sedangkan menurut Chang dkk. (2008) dan Glotin dkk. (2009), ciri tingkat tinggi adalah ciri yang masih dapat diturunkan. Ciri tingkat tinggi langsung dapat diperoleh saat mengekstraksi ciri tingkat rendah karena ciri tingkat tinggi dihitung dari ciri tingkat rendah.

Ekstraksi ciri adalah salah satu proses yang pertama kali dilakukan untuk mengenali sebuah objek dalam sebuah citra. Ekstraksi ciri bertujuan untuk memperoleh *region* penting dalam sebuah citra berdasarkan karakteristik intrinsik dari objek yang akan dikenali atau diklasifikasi. *Region* dapat dibedakan berdasarkan bentuk, tekstur, intensitas dan lainnya. Ciri yang digunakan dalam banyak tulisan (Choras, 2007; Xiao dkk., 2007; Choras, 2007; Gupta, 2011; Chadha dan Carolina, 2012; Dharani dan Aroquiaraj, 2013) adalah ciri bentuk, ciri tekstur dan ciri warna.

3.4.1 Ciri Bentuk

Pengambilan gambar berbasis bentuk terdiri dari pengukuran kemiripan antara bentuk dengan ciri yang direpresentasikan. Beberapa ciri geometrik sederhana dapat digunakan untuk mendeskripsikan bentuk. Ciri geometrik sederhana hanya dapat membedakan bentuk melalui perbedaan besar (Mingqiang dkk., 2008). Ciri bentuk diperoleh dari citra biner yang ditransformasi dari citra grayscale. Citra biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu putih dan hitam.



Gambar 3.2 Beberapa ciri bentuk

Berikut dijelaskan ciri bentuk sebagai berikut:

1. *Bounding box*

Bounding box adalah kotak terkecil yang didapatkan untuk mencakup seluruh objek area (Suliman dkk., 2010). *Bounding box* sejajar dengan *major axis*. *Bounding box* bermanfaat untuk pelabelan objek (Sánchez dkk., 2012), untuk mendeteksi posisi ban kendaraan (Boyle dan Ferryman, 2015), deteksi kendaraan (Sina dkk., 2013; C dan Kumar, 2015), klasifikasi objek (Sánchez dkk., 2012), (Morris dan Trivedi, 2008; Kafai dan Bhanu, 2012; Rabi, 2013), untuk menghitung jumlah kendaraan (Sina dkk., 2013; Basri dkk., 2015), *Bounding box* berisi *width of the bounding box*, *height of the bounding box*, koordinat x dan koordinat y. Koordinat x dan y digunakan untuk membedakan posisi objek itu dengan objek yang lain, menentukan posisi awal objek dan mempermudah perhitungan nilai ciri. Gambar 3.2 menunjukkan ciri *Bouding Box*.

2. *Width of the bounding box (WBB)*

Width of the bounding box (WBB) adalah lebar *bounding box* (Wenger dkk., 2010). WBB digunakan untuk menentukan jenis kendaraan (Chen dan Ellis, 2011; Chen dkk., 2012, 2011; Wenger dkk., 2010). Gambar 3.2 menunjukkan ciri *width of the bounding box*.

3. *Height of the bounding box (HBB)*

Height of the bounding box (HBB) adalah tinggi *bounding box* (Wenger dkk., 2010). HBB digunakan untuk menentukan jenis kendaraan (Chen dan Ellis,

2011; Chen dkk., 2012, 2011; Wenger dkk., 2010). Gambar 3.2 menunjukkan ciri *height of the bounding box*.

4. *Major axis* atau panjang *major axis*

Major axis atau panjang *major axis* adalah diameter panjang dari objek (Chen dan Ellis, 2011; Chen dkk., 2012, 2011; Wenger dkk., 2010). Gambar 3.2 menunjukkan ciri *major axis*.

5. *Minor axis* atau panjang *minor axis*

Minor axis atau panjang *minor axis* adalah diameter pendek dari objek (Chen dan Ellis, 2011; Chen dkk., 2012, 2011; Wenger dkk., 2010). Gambar 3.2 menunjukkan ciri *minor axis*.

6. *Area Region of Interest (ROI)*

Area Region of Interest (ROI) adalah sebuah area segi empat yang membatasi objek. Area ROI berasal dari nilai area *bounding box*. (Huang dkk., 2012; Srivastava dan Singh, 2014; Mithun dkk., 2012). Area ROI dihitung dengan persamaan (3.1).

$$\text{Area ROI} = \text{WBB} \times \text{HBB} \quad (3.1)$$

7. *Centroid*

Centroid adalah titik tengah dari koordinat objek (Li dkk., 2008; Morris dan Trivedi, 2008). *Centroid* dapat digunakan untuk menandai posisi kendaraan (Saran dan Sreelekha, 2015). *Centroid* juga dikenal dengan istilah *Mass Point*. *Mass Point* adalah titik massa. Nilai yang diambil untuk titik massa adalah titik tengah dari *object area/white area* (Mann dkk., 2012; Bayer dan Hauser, 2012; Dutta dan Shen, 2016). Gambar 3.2 menunjukkan ciri *centroid*.

8. *Filled area*

Filled area adalah area yang diisi (Chen dkk., 2011). *Filled area* digunakan untuk menentukan jenis kendaraan. Gambar 3.2 menunjukkan ciri *filled area*.

9. *Object Area*

Object Area adalah area suatu objek (Li dkk., 2010; Han dkk., 2008, 2011; Lin dan Siang, 2012). Area adalah jumlah piksel dalam objek (Wenger dkk., 2010). *Object area* juga dikenal sebagai *White Area*. *White Area* adalah area putih

dinyatakan dengan jumlah piksel dari daerah objek yang terpilih (Gonzalez dkk., 2008; Morris dan Trivedi, 2008; Wenger dkk., 2010).

10. *Aspect Ratio*

Aspect Ratio adalah rasio antara *height of the bounding box* (HBB) dan *width of the bounding box* (WBB) (Wenger dkk., 2010). *Aspect Ratio* digunakan untuk mengklasifikasi kendaraan dan menghitung jumlah kendaraan (Kota dan Rao, 2013). *Aspect Ratio* dapat ditunjukkan pada persamaan (3.2).

$$\text{Aspect Ratio} = \frac{HBB}{WBB} \quad (3.2)$$

11. *Perimeter*

Perimeter atau keliling area adalah panjang tepian/garis batas objek (Wenger dkk., 2010). Bila objek berbentuk segi empat maka perimeter ini sering disebut *perimeter bounding box*. *Perimeter bounding box* adalah keliling *bounding box* = $2 \times (\text{width of the bounding box} + \text{height of the bounding box})$ (Evans dkk., 2013).

12. *Dispersedness (D)*

Dispersedness (D) (Sánchez dkk., 2012) juga dikenal dengan *Compactness* (Wenger dkk., 2010). *Compactness* adalah kekompakan atau kepadatan piksel dalam suatu area. Bentuk terkecil dari *Dispersedness (D)* adalah lingkaran. *Dispersedness (D)* digunakan untuk mengklasifikasi jenis kendaraan dan menghitung jumlah kendaraan (Kota dan Rao, 2013; Chen dkk., 2011). *Dispersedness (D)* dapat dihitung dengan persamaan (3.3).

$$D = \frac{l^2}{A} \quad (3.3)$$

dimana l adalah *perimeter* objek dan A adalah *area* (Chen dkk., 2011; Sánchez dkk., 2012).

13. *Inverse Dispersedness*

Inverse Dispersedness diperlukan untuk menormalisasi *Dispersedness* karena *Dispersedness* tidak memiliki batas atas (Sánchez dkk., 2012). *Inverse Dispersedness* dihitung dengan persamaan (3.4).

$$\text{Inverse Dispersedness} = \frac{4\pi}{\text{Dispersedness}} \quad (3.4)$$

14. Edge Density

Ciri *Edge Density* (kerapatan tepi) diekstrak dengan cara membagi area menjadi beberapa *block area* yang lebih kecil (Bravo dkk., 2013). *Edge Density* (kerapatan tepi) dapat dihitung dengan persamaan (3.5):

$$\text{Edge density} = \frac{1}{\text{Block area}} \sqrt{\sum_{\text{block}} (g_v^2 + g_h^2)} \quad (3.5)$$

dengan g_v dan g_h adalah respon vertikal dan horisontal dari operator deteksi tepi (misalnya menggunakan Sobel) pada setiap piksel di dalam blok (Bravo dkk., 2013).

Pada eksperimen yang dilakukan (Bravo dkk., 2013) hanya sisi yang paling terlihat dari model objek kendaraan yang dibagi menjadi blok yang lebih kecil. Ini digunakan untuk menghitung ciri vektor *Edge Density* yang sesuai.

15. grayHOG

grayHOG adalah *Histogram of Oriented Gradient* (HOG) yang di dapat dari area objek keabuan (Mithun dkk., 2012). Penggunaan ciri HOG pertama kali diperkenalkan oleh (Dalal dan Triggs, 2005). HOG dapat digunakan untuk deteksi kendaraan (Arróspide dkk., 2012; Tian dkk., 2013), klasifikasi kendaraan (Lee dkk., 2015) dan pelacakan kendaraan (Tian dkk., 2013). Tahapan yang dilakukan untuk memperoleh vektor ciri HOG (Sugiharto dan Harjoko, 2016) adalah:

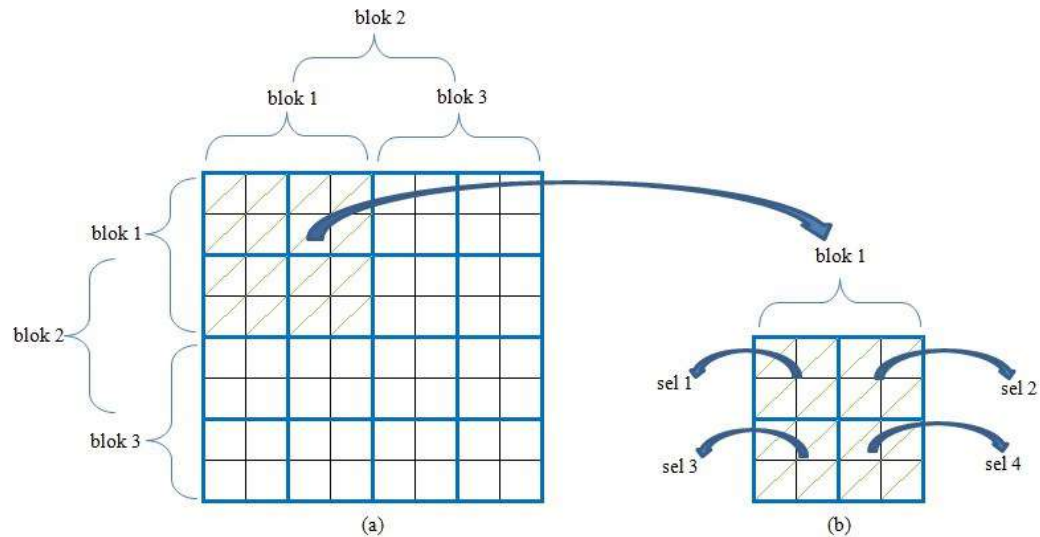
- a. Hitung gradien horizontal $I_x = I * D_x$ dan vertikal $I_y = I * D_y$ dari ROI 128x128 dengan D_x dan D_y merupakan operator matriks horizontal dan

$$\text{vertikal. } D_x = [-1 \ 0 \ 1] \text{ dan } D_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

- b. Hitung kekuatan $|G| = \sqrt{I_x^2 + I_y^2}$

- c. Hitung arah orientasi $\theta = \arctan \frac{I_x}{I_y}$

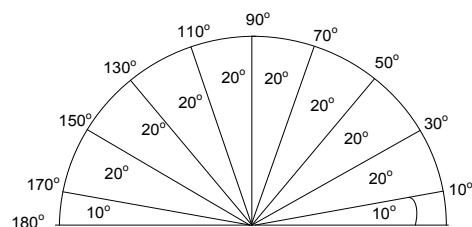
- d. ROI dibagi menjadi blok-blok berukuran 16x16 yang saling beririsan (jika ukuran ROI adalah 128 x 128 maka terdapat $15 \times 15 = 225$ blok) dan masing-masing blok memiliki 4 sel masing-masing berukuran 8 x 8.



Gambar 3.3 HOG (a) Area objek 8x8 (b) blok

Untuk sederhananya, diberikan contoh area objek 8 x 8, kemudian dibentuk menjadi blok 4 x 4. Lalu dibuat blok berikutnya yang beririsan setengahnya. Gambar 3.3 menunjukkan 1 blok = 4 x 4 maka jumlah bloknya = 9 blok. Blok ini dibagi menjadi sel = 2 x 2.

- e. Kuantisasi orientasi dihitung dalam 9 bin $0^\circ \leq \theta \leq 180^\circ$



Gambar 3.4 Sudut 180°

Gambar 3.4 menunjukkan rumus sudut 180°. Contoh, bila sudut = 75° maka lihat rumus untuk sudut antara 70° - 90°. Lalu dari rumusnya maka

akan diperoleh hasil yang menunjukkan h mana yang sesuai. Ini menggunakan interpolasi bilinear.

16. *Pyramidal Histogram of Oriented Gradients* (PHOG) / piramidHOG

Pyramidal Histogram of Oriented Gradients (PHOG) / piramidHOG adalah piramid yang elemen-elemennya berisi HOG. PHOG berfungsi untuk multi resolusi. PHOG merupakan pengembangan ciri HOG yang dikembangkan oleh (Bosch dkk., 2007). Ciri ini mewakili area dengan bentuk lokal dan tata ruang bentuk. Deskriptor terdiri dari sebuah *Histogram Of Oriented Gradient* dalam setiap blok area yang ada pada setiap tingkat resolusi. Metode PHOG menggunakan level L, yang menentukan jumlah blok yang diperhitungkan untuk analisis. Di setiap level, area dibagi menjadi 2^L blok untuk setiap ruang. Bentuk lokal ditangkap oleh distribusi melalui orientasi tepi dalam suatu area ($L=0$), sedangkan tata ruang lainnya ada di beberapa resolusi ($L>0$).

Dua pendekatan PHOG berbeda diuji. Pertama, deskriptornya dihitung menggunakan orientasi tepi $\theta(x,y)$ di atas setiap blok area, sedangkan dalam pendekatan kedua, kontribusi dari setiap orientasi tepi tertimbang menurut besarnya $m(x,y)$ (Bravo dkk., 2013). Besaran dan orientasi gradien pada piksel (x,y) dihitung dengan persamaan (3.6-3.7):

$$m(x,y) = \sqrt{g_x(x,y)^2 + g_y(x,y)^2} \quad (3.6)$$

$$\theta(x,y) = \arctan\left(\frac{g_x(x,y)}{g_y(x,y)}\right) \quad (3.7)$$

dengan $g_x(x,y)$ dan $g_y(x,y)$ masing-masing adalah gradien area sepanjang arah x dan y.

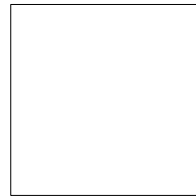
Vektor PHOG yang dihasilkan untuk setiap level digabungkan untuk mendapatkan vektor ciri akhir. (Bravo dkk., 2013) menggunakan $L=3$ dalam eksperimennya. Pada kasus ini, jarak Chi-square X^2 antara vektor sampel uji tunggal, X_{test} dan himpunan pelatihan X_{train} vektor ciri PHOG tunggal digunakan untuk menganalisis persamaan (3.8):

$$X^2 = \sum_{i=2}^n \frac{(X_{test,i} - X_{train,i})^2}{(X_{test,i} - X_{train,i})} \quad (3.8)$$

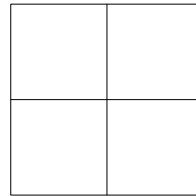
dengan i adalah elemen ke- i dari vektor ciri dengan panjang n .

Di setiap lokasi model, ciri PHOG gabungan dibandingkan dengan sampel pelatihan sesuai dengan persamaan (3.23) dan jarak minimum ditambahkan ke vektor kesamaan. PHOG dapat digunakan untuk mencari tempat parkir kosong di suatu area (Bravo dkk., 2013). PHOG juga dapat digunakan untuk deteksi kendaraan (Wang dkk., 2013) dan klasifikasi kendaraan (Mishra dan Banerjee, 2013; Zhang dan Zhao, 2011). Gambar 3.5 menunjukkan ilustrasi PHOG.

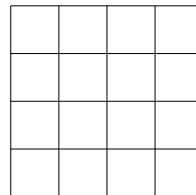
Level = 0
 $2^0 \times 2^0 = 1$
 1 bin = 9 ciri



Level = 1
 $2^1 \times 2^1 = 4$
 1 bin = 9 ciri
 4 bin = 9 x 4 = 36 ciri



Level = 2
 $2^2 \times 2^2 = 16$
 1 bin = 9 ciri
 16 bin = 9 x 16 = 144 ciri



Jumlah ciri: $9 + 36 + 144 = 189$

Gambar 3.5 Ilustrasi PHOG

Tahapan dalam menentukan ciri PHOG (Bosch dkk., 2007) adalah :

- a. ROI dikenakan operasi deteksi tepi (misalnya menggunakan detektor Canny) untuk mendapatkan kontur tepi.
- b. ROI hasil deteksi tepi kemudian dipecah menjadi sel-sel pada j level piramida atau hirarki, yang tiap jumlahnya ditentukan dengan rumus $c_j = 2^j$.

- c. masing-masing level piramida dihitung ciri HOG untuk mendapatkan histogram yang merepresentasikan ciri bentuk lokal.
- d. ciri PHOG merupakan keseluruhan vektor ciri HOG dari setiap level piramida.

17. *Equivdiameter* (E)

Equivdiameter (E) dinyatakan dalam persamaan (3.9) sebagai:

$$E = \sqrt{\frac{4A}{\pi}} \quad (3.9)$$

dimana A = area, E = diameter sebuah lingkaran dengan area yang sama seperti region. *Equivdiameter* digunakan untuk menentukan jenis kendaraan. (Chen Sdkk., 2011; Gonzalez dkk., 2009).

18. *Convex area*

Convex area adalah area yang memiliki semua sudut interior kurang dari 180°. *Convex area* digunakan untuk menentukan jenis kendaraan. (Gonzalez dkk., 2008; Chen dkk., 2011). Gambar 3.6 menunjukkan ciri *convex area*.



Gambar 3.6 Convex area

19. *Solidity* atau soliditas

Solidity atau soliditas adalah tingkat kesolidan objek. *Solidity* digunakan untuk menentukan jenis kendaraan. (Chen dkk., 2009; Rashid dkk., 2010; Chen dkk., 2011). *Solidity* dihitung dengan area dibagi *convex area*. *Solidity* dihitung dengan persamaan (3.10).

$$Solidity = \frac{area}{convex\ area} \quad (3.10)$$

20. *Convex hull*

Convex hull adalah poligon berbentuk *convex* yang meliputi sebuah objek.

- a. *Convex hull X* adalah titik-titik koordinat X dari *convex hull* (Gonzalez dkk., 2008; Morris dan Trivedi, 2008)
- b. *Convex hull Y* adalah titik-titik koordinat Y dari *convex hull* (Gonzalez dkk., 2008; Morris dan Trivedi, 2008).

21. *Orientation*

Orientation adalah sudut antara sumbu x dan panjang *major axis* dari *ellipse* (Wenger dkk., 2010). *Orientation* dapat digunakan untuk klasifikasi kendaraan (Chun-Hong dan Zhi-Lei, 2016) dan pelacakan kendaraan (Shruthi, 2011).

22. *Extent*

Extent adalah rasio area objek dengan area *bounding box* (Wenger dkk., 2010). *Extent* digunakan untuk menentukan jenis kendaraan. (Chen dan Ellis, 2011; Chen dkk., 2012, 2011; Wenger dkk., 2010). *Extent* dihitung dengan persamaan (3.11).

$$Extent = \frac{object\ area}{area\ bounding\ box} \quad (3.11)$$

23. *EulerNumber*

EulerNumber adalah jumlah objek dalam region dikurangi dengan jumlah *holes* (lubang) dalam objek itu (Gonzalez dkk., 2009). *EulerNumber* dihitung dengan persamaan (3.12).

$$EulerNumber = \text{jumlah objek dalam region} - \text{jumlah holes objek} \quad (3.12)$$

Euler number dapat digunakan untuk deteksi kendaraan dan plat nomor kendaraan (Kodwani, 2013)

24. *Eccentricity* atau eksentrisitas

Eccentricity atau eksentrisitas dari sebuah *ellipse* adalah ukuran untuk mengukur tingkat kebundaran area *ellipse* suatu objek. *Eccentricity* untuk menggambarkan putaran atau seberapa besar *ellipse* membentang. Jika eksentrisitas dari sebuah *ellipse* mendekati satu (seperti 0,8 atau 0,9), maka bentuk *ellipse* panjang dan kurus. Jika eksentrisitas = nol, maka bentuk *ellipse* seperti lingkaran. Jika eksentrisitas = 1, maka bentuk *ellipse* seperti

parabola. Jika eksentrisitas > 1 , maka bentuk *ellipse* seperti hiperbola. *Eccentricity* adalah rasio jarak antara foci dan panjang *major axis* dari *ellipse* (Wenger dkk., 2010). *Ellipse* memiliki 2 titik fokus. Oleh karena ada 2 fokus maka disebut *foci*. *Eccentricity* digunakan untuk menentukan jenis kendaraan (Chen dkk., 2011). *Eccentricity* dihitung dengan persamaan (3.13).

$$Eccentricity = \frac{\text{jarak antara foci}}{\text{major axis}} \quad (3.13)$$

25. *Area ratio* atau rasio area

Area ratio atau rasio area adalah perbandingan antara area dan area ROI (Huang dkk., 2012). *Area ratio* dihitung dengan persamaan (3.14).

$$Area\ ratio = \frac{area}{area\ ROI} \quad (3.14)$$

26. *Axis Ratio*

Axis Ratio juga dikenal sebagai *Major Axis - Minor Axis Ratio*. *Axis Ratio* adalah rasio antara panjang *major axis* dan panjang *minor axis* (Rashid dkk., 2010). *Axis Ratio* dihitung dengan persamaan (3.15).

$$Axis\ Ratio = \frac{Major\ Axis}{Minor\ Axis} \quad (3.15)$$

27. *Circularity*

Circularity adalah tingkat kebulatan objek (Chen dkk., 2011; Mingqiang dkk., 2008). *Circularity* dihitung dengan persamaan (3.16).

$$C = \frac{4\pi A}{l^2} \quad (3.16)$$

dengan A adalah *area* dan l adalah *perimeter*.

28. *Elongation*

Elongation adalah ukuran yang memiliki range nilai $[0\ 1]$, syaratnya WBB nilainya lebih kecil dari HBB. Bentuk simetris di semua sumbu seperti lingkaran atau persegi akan memiliki nilai *elongasi* = 0 sedangkan *aspect ratio* yang nilainya besar akan memiliki perpanjangan lebih dekat ke 1 (Yang dkk., 2008). *Elongation* salah satu ciri yang cocok digunakan untuk klasifikasi

(Morris dan Trivedi, 2008). Menurut (Karaimer dkk., 2015), *Elongation* dihitung dengan persamaan (3.17):

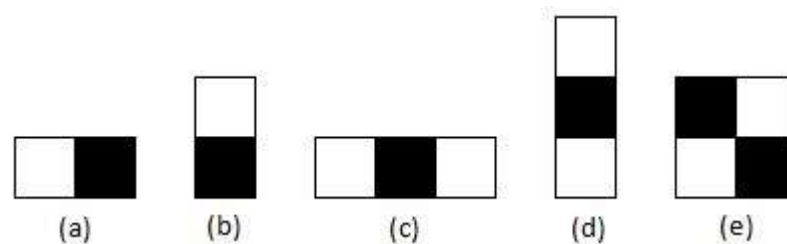
$$Elongation = 1 - \frac{WBB}{HBB} \quad (3.17)$$

29. Haar like feature

Haar like feature (ciri mirip Haar) adalah filter persegi panjang sederhana. *Haar like feature* memiliki nilai skalar yang mewakili perbedaan dalam jumlah intensitas antara daerah persegi panjang yang berdekatan. Nilai-nilai ciri dapat dihitung dengan cepat menggunakan representasi gambar integral. Sederhananya, ciri ini dievaluasi pada posisi yang berbeda dan dengan ukuran yang berbeda secara mendalam sesuai resolusi dasar dari *classifier*. Sebagai contoh, ketika resolusi *classifier* adalah 24 x 18 piksel maka ciri yang dihasilkan adalah 91620 nilai ciri. Setiap ciri dievaluasi pada semua sampel pelatihan dan kepadatan probabilitas satu dimensi untuk setiap objek dan kelas non-objek. Threshold (ambang batas) yang memisahkan kedua distribusi ini dipilih untuk setiap ciri. Ciri-ciri ini bersama dengan ambang batas masing-masing dan polaritas membentuk pengklasifikasi yang lemah untuk algoritma pembelajaran (Adhikari dkk., 2011). Gambar 3.7 menunjukkan contoh himpunan *Haar like feature*. *Classifier Haar like feature* secara matematis dapat dihitung dengan persamaan (3.18):

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{jika } pf(x) > p\theta \\ -1 & \text{jika } pf(x) \leq p\theta \end{cases} \quad (3.18)$$

dengan x adalah resolusi dasar dari *classifier*, ciri Haar-like, *threshold* (nilai ambang) untuk ciri dan polaritas yang menunjukkan arah ketidaksetaraan (Adhikari dkk., 2011).



Gambar 3.7 Contoh himpunan Haar like feature

Haar like featue yang digunakan oleh (Mohamed dkk., 2015) hanya menggunakan area objek Haar (a) dan (b) dari Gambar 3.24. (Mohamed dkk., 2015) menyatakan langkah pertama dalam menghitung ciri Haar adalah menggunakan konsep area objek integral yang dihitung dengan persamaan (3.19-3.21) yang diperkenalkan oleh (Viola dan Jones, 2001), yang memberikan representasi dari area objek input dan mengurangi waktu komputasi ciri-ciri ini. Kecepatan area objek integral dalam perhitungan menunjukkan jumlah persegi panjang hanya menggunakan empat referensi, maka perbedaan antara dua persegi panjang yang berdekatan, dapat dihitung dengan hanya enam referensi dan delapan untuk tiga persegi panjang.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3.19)$$

dengan: $ii(x,y)$ adalah area objek integral dan $i(x,y)$ adalah area objek asli. Kemudian penggunaan dua persamaan memungkinkan perhitungan haar windows pada berbagai skala dalam suatu area objek.

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (3.20)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3.21)$$

dengan $s(x,y)$ menunjukkan jumlah kumulatif garis dan $s(x,-1) = ii(-1,y)=0$.

Pendekatan yang dilakukan oleh (Mohamed dkk., 2015) hanyalah menggunakan 2 filter persegi panjang (horisontal dan vertikal) dengan skala yang berbeda yaitu 2x2, 4x4, 8x8 dan 16x16. Detail area objek yang difilter tetap mempertahankan tepi yang paling penting karena ukuran filter digandakan. Hasil *Haar like feature* yang menggunakan dua filter persegi panjang 24x24 adalah 8712 piksel. *Haar like feature* dapat digunakan untuk deteksi kendaraan (Han dkk., 2009; Mohamed dkk., 2015; Adhikari dkk., 2011) dan klasifikasi kendaraan (Mishra dan Banerjee, 2013).

30. Gabor

Filter *Gabor* dibentuk oleh fungsi kernel *Gaussian* dalam domain spasial dua dimensi. Filter *Gabor* adalah gelombang bidang sinusoidal yang diformulasikan dalam persamaan (3.22 - 3.23):

$$G(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i2\pi \frac{x'}{\lambda} + \varphi\right) \quad (3.22)$$

dengan:

$$\begin{aligned} y' &= x \cos\theta + y \sin\theta \\ x' &= -x \sin\theta + y \cos\theta \end{aligned} \quad (3.23)$$

G adalah kernel filter *Gabor* pada koordinat spasial x, y dengan dimensi tertentu dan parameter lain yang dapat dideskripsikan dengan : θ adalah orientasi gelombang, ψ adalah fase offset, σ adalah standar deviasi, γ adalah rasio aspek, i adalah bilangan kompleks ($a + bi$), λ adalah panjang gelombang faktor sinusoidal. Pada perhitungannya akan dipisahkan antara bagian nyata dan bagian imajiner. Setelah itu, konvolusi gambar asli dihitung dengan kernel *Gabor*, baik bagian nyata maupun imajiner. Bagian imajiner dari kernel *Gabor* dapat dihitung dengan persamaan (3.24):

$$G_{imag}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \varphi\right) \quad (3.24)$$

Sedangkan bagian nyata dari kernel *Gabor* dapat dihitung dengan persamaan (3.25):

$$G_{real}(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right) \quad (3.25)$$

Ciri vektor berisi besaran dari setiap kombinasi area objek. Area objek ini dihasilkan oleh konvolusi bagian nyata dan kernel bagian imajiner dengan area objek asli. Persamaan bagian imajiner (3.24) membangun bagian imajiner dari kernel *Gabor* dengan ukuran tertentu, dan persamaan bagian nyata (3.25) menghasilkan bagian nyata dari kernel *Gabor* dengan ukuran yang sama. (Nurhadiyatna dkk., 2015). *Gabor* dapat digunakan untuk deteksi kendaraan (Teoh, 2011) dan klasifikasi jenis kendaraan (Muzakki dkk., 2017; Nurhadiyatna dkk., 2015). *Gabor* juga dapat digunakan untuk verifikasi kendaraan (Guo dkk., 2014; Guo, 2013).

31. *Scale Invariant Feature Transform* (SIFT)

Scale Invariant Feature Transform (SIFT) diperkenalkan oleh (Lowe, 2004). SIFT dapat digunakan untuk klasifikasi kendaraan (Mishra dan Banerjee,

2013). SIFT biasanya digunakan untuk mendeteksi objek dan pengenalan. Ciri yang diekstrak menggunakan algoritma SIFT adalah invarian untuk rotasi dan skala. Ciri ini tidak terpengaruh oleh perubahan sudut pandang, *noise* dan pencahayaan. Algoritma SIFT menggunakan 4x4 piksel ketetanggaan untuk menghitung histogram orientasi tepi. Orientasi tepi ini diwakili oleh 8 bin untuk setiap 4x4 piksel ketetanggaan.

3.4.2 Ciri Tekstur

Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu *region* yang cukup besar sehingga sifat-sifat tersebut secara alami dapat berulang dalam *region* tersebut. Ciri tekstur dihitung berdasarkan distribusi statistik dari intensitas piksel terhadap posisi antar piksel dan direpresentasikan dalam sebuah matriks.

1. *Local Binary Pattern (LBP)*

Local Binary Pattern (LBP) merupakan suatu operasi citra yang mentransformasikan sebuah citra menjadi sebuah susunan label integer yang menggambarkan kenampakan skala kecil dari suatu citra. Label atau statistik tersebut, biasanya merupakan sebuah histogram, kemudian digunakan lagi untuk analisis citra yang lebih lanjut. LBP adalah ekstraktor ciri kuat yang mengubah citra menjadi label bilangan integer menggunakan mean (nilai rata-rata) yang efisien secara komputasi. Sebelum proses ekstraksi ciri, beberapa langkah pra pemrosesan citra diterapkan untuk standarisasi dan meningkatkan kecerahan area. Awalnya, ukuran citra disamakan, selanjutnya citra RGB diubah menjadi citra *gray* (keabuan) dengan mempertahankan nilai intensitasnya. Selain itu, teknik peningkatan kontras juga diterapkan untuk mengurangi variasi iluminasi citra (Rabiu, 2013). *LBP* digunakan untuk mengekstrak atribut unik dari setiap objek. LBP pertama kali diperkenalkan oleh (Ojala dkk., 2002) sebagai skala keabuan dan tekstur deskriptor invarian rotasi. LBP memberi label piksel P_n untuk ($n = 0, 1, \dots, 7$). Pada prinsipnya, sebuah invarian rotasi LBP dihitung 3x3 piksel ketetanggaannya dalam intensitas area dengan membagi nilai intensitas ketetanggaan P_n berdasarkan

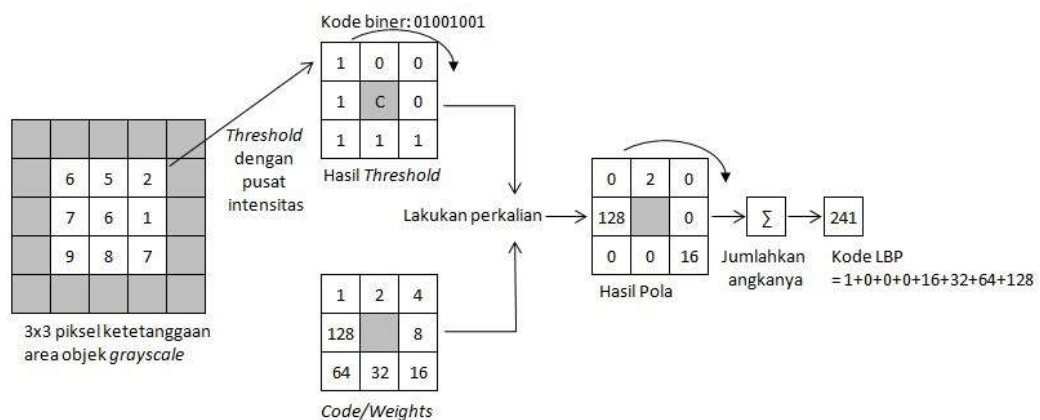
piksel pusat P_c . Hasilnya adalah bilangan biner (Rabiu, 2013). Sebuah piksel pada lokasi (x_c, y_c) menghasilkan LBP dihitung dengan persamaan (3.26):

$$LBP(x_c, y_c) = \sum_{n=0}^7 S(P_n - P_c) 2^n \quad (3.26)$$

dengan P_c adalah nilai level keabuan dari piksel pusat dan P_n adalah 8 nilai ketetanggaan sekelilingnya dan S dapat dihitung dengan persamaan (3.27):

$$S(P_n - P_c) = \begin{cases} 1, & P_n > c \\ 0, & P_n \leq c \end{cases} \quad (3.27)$$

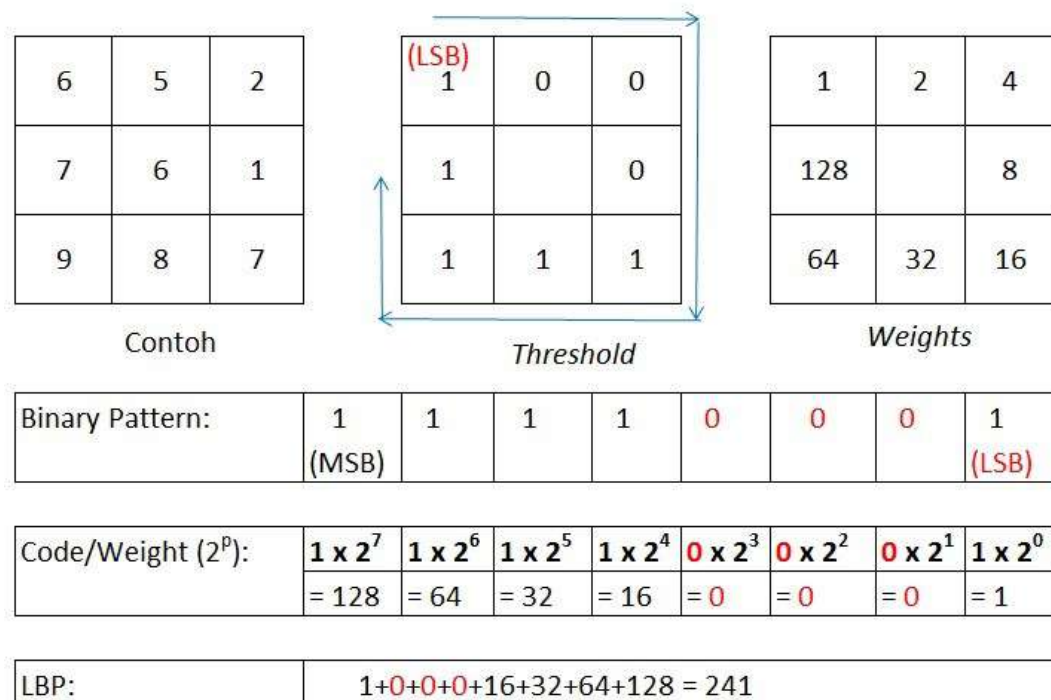
Jika intensitas tetangga yang dibandingkan lebih besar atau sama dengan pusat intensitasnya maka nilai transformasi binernya adalah 1. Sebaliknya, jika intensitas tetangga yang dibandingkannya lebih kecil daripada pusat intensitasnya maka nilai transformasi binernya adalah 0. Setelah ke-8 tetangganya dibandingkan, nilai transformasi tersebut akan disusun dan dikonversikan ke dalam bentuk desimal (Firdaus dkk., 2016). Gambar 3.8 menunjukkan contoh cara kerja operator LBP dan Gambar 3.9 menunjukkan cara perhitungan LBP.



Gambar 3.8 Contoh cara kerja operator LBP

Model dasar LBP akan menghasilkan keluaran $2^8 = 256$. Jika area deskripsi tekstur terlalu kecil histogramnya sangat jarang dan tidak ekspresif. Selain itu, histogram LBP dapat dipengaruhi suara dan perubahan yang kecil. Hasil penelitian menunjukkan bahwa pola *Uniform* (seragam) LBP mewakili lebih dari 90% tektur area informasi dalam deskripsi tekstur area objek. Pola

Uniform LBP dapat digunakan untuk mengurangi panjang vektor ciri dan menerapkan deskriptor invariant rotasi sederhana. Ide ini dimotivasi oleh fakta bahwa beberapa pola biner terjadi lebih umum dalam gambar tekstur daripada yang lain. Pola biner lokal disebut *Uniform* (seragam) jika pola biner berisi paling banyak dua 0-1 atau 1-0 transisi. Sebagai contoh, 00010000 (2 transisi) adalah pola yang seragam, 01010100 (6 transisi) tidak. Dalam perhitungan histogram LBP, histogram memiliki *bin* terpisah untuk setiap pola *Uniform* (seragam), dan semua pola tidak seragam ditetapkan ke satu *bin*.



Gambar 3.9 Contoh perhitungan LBP

Algoritma sederhana untuk mengukur keseragaman kode LBP adalah meringkas nilai absolut dari perbedaan antara kode dan kode yang digeser secara bergiliran. Pola *Uniform* LBP $U(G_n)$ dihitung dengan persamaan (3.28):

$$U(G_n) = |s(P_{n-1} - P_c) - s(P_0 - P_c)| + \sum_{n=1}^{n-1} |s(P_n - P_c) - s(P_{n-1} - P_c)| \quad (3.28)$$

Menggunakan pola *Uniform* (seragam), panjang vektor ciri untuk sebuah sel tunggal mengurangi dari 256 ke 59. Ke-58 pola biner seragam sesuai dengan

bilangan bulat 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254 dan 255 (Xu dkk., 2015). LBP dapat digunakan untuk deteksi kendaraan (Rabiu, 2013; Xu dkk., 2015; Neumann dkk., 2017) dan klasifikasi kendaraan (Rabiu, 2013; Firdaus dkk., 2016).

2. *Invariant Moment* (momen invarian)

Invariant Moment (momen invarian) adalah properti momen dari area dalam citra biner yang invarian terhadap translasi, rotasi dan skala (Gonzalez dkk., 2008; Ince, 2011; Li dkk., 2008).

Moment dua dimensi dari $(p + q)$ yang berurutan dari sebuah citra digital $f(x,y)$ yang memiliki ukuran $M \times N$ dapat dihitung dengan persamaan (3.29):

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (3.29)$$

dengan $p = 0,1,2,\dots$ dan $q=0,1,2,\dots$ adalah integer.

Central moment $(p + q)$ dapat dihitung dengan persamaan (3.30):

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (3.30)$$

untuk $p=0,1,2,\dots$ dan $q=0,1,2,\dots$ dapat dihitung dengan persamaan (3.31):

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{01}}{m_{00}} \quad (3.31)$$

central moment yang ternormalisasi, dinotasikan dengan η_{pq} dapat dihitung dengan persamaan (3.32):

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (3.32)$$

dengan persamaan (3.33)

$$\gamma = \frac{p + q}{2} + 1 \quad (3.33)$$

untuk $p + q = 2,3,\dots$

Tujuh invariant moment dapat diturunkan dari moment kedua dan ketiga yang dihitung dengan persamaan (3.34 - 3.40):

$$\emptyset_1 = \eta_{20} + \eta_{02} \quad (3.34)$$

$$\emptyset_2 = (\eta_{20} - \eta_{02})^2 + 4(\eta_{11})^2 \quad (3.35)$$

$$\emptyset_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.36)$$

$$\emptyset_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (3.37)$$

$$\emptyset_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (3.38)$$

$$- 3(\eta_{21} - \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})$$

$$[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\emptyset_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (3.39)$$

$$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\emptyset_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (3.40)$$

$$+(3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Invariant moment dapat digunakan untuk klasifikasi kendaraan (Hao dkk., 2013; Qin, 2008).

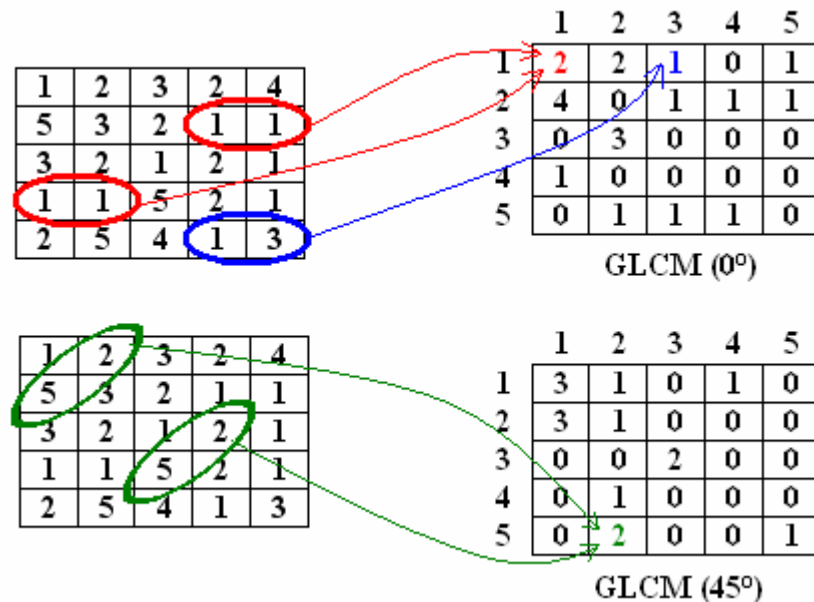
3. *Statistical Texture*

Statistical Texture adalah 22 nilai momen statistik berupa: *Contrast (CONTR)*, *Dissimilarity (DISSI)*, *Homogeneity (HOM)/Inverse Difference Moment (IDM)*, *Similarity (SIM)/Inverse Difference Moment Normalized (IDMNC)*, *Angular Second Moment (ASM)/Energy*, *Entropy (ENT)*, *Mean (sum mean)*, *Variance and standard deviation (VAR and SD)*, *Maximum probability (MaxP)*, *Correlation (COR)*, *Diagonal moment (DM)*, *Second diagonal moment (SDM)*, *Coefficient of variation (CVAR)*, *Sum entropy (SENT)*, *Sum variance (SVAR)*, *Different entropy (DENT)*, *Different variance (DVAR)*, *Cluster shade (CSH)/Skewness*, *Cluster prominence (CPR)*, *Sum average (SAVR)*, *Mean correlation 1 (MCOR1)*, *Mean correlation 2 (MCOR2)*. *Statistical texture* ini digunakan untuk klasifikasi (Abouelatta, 2013). Pada persamaan-persamaan berikut maka G adalah *Gray Level Co-occurrence Matrices (GLCM)* yang ternormalisasi. GLCM adalah representasi populer

untuk tekstur dalam area. GLCM berisi hitungan jumlah ciri yang diberikan (misalnya, tingkat abu-abu yang diberikan) terjadi dalam hubungan spasial tertentu dengan ciri lain yang diberikan. $G(i,j)$ adalah probabilitas munculnya pasangan piksel bernilai i dan j dari matrik G . n adalah jumlah elemen GLCM. i dan j adalah baris dan kolom pada G . $P(i,j)$ adalah nilai matrik GLCM pada indeks i,j . $G(i,j)$ dapat dihitung dengan persamaan (3.41):

$$G(i,j) = \frac{P(i,j)}{\sum_{i=0}^n \sum_{j=0}^n P(i,j)} \quad (3.41)$$

Arah pembentukan GLCM adalah 0° , 45° , 90° dan 135° . Gambar 3.10 menunjukkan contoh pembentukan GLCM pada arah 0° dan 45° . Pada penelitian ini dipakai pembentukan GLCM pada arah 0° .



Gambar 3.10 Contoh pembentukan GLCM pada arah 0° dan 45°

a. *Contrast (CONTR)*

Contrast adalah pengukuran terhadap variasi lokal dalam area. *Contrast* akan bernilai rendah untuk area seragam dan bernilai tinggi ketika area mempunyai variasi yang tinggi. *Contrast* berisi ukuran kontras intensitas antara piksel dan tetangganya yang meliputi keseluruhan gambar. *Contrast*

bernilai 0 (nol) untuk area konstan. *Contrast* dihitung dengan persamaan (3.42).

$$Contrast = \sum_{i=0}^{n-1} n^2 \sum_{j=0}^{n-1} G(i, j)(i - j)^2 \quad (3.42)$$

b. *Dissimilarity (DISSI)*

Dissimilarity mirip dengan *Contrast*. *Dissimilarity* akan bernilai tinggi ketika area lokal memiliki nilai *Contrast* yang tinggi. Ciri ini sensitif pada *grey level spatial variability* dan *tone area*. *Dissimilarity* dihitung dengan persamaan (3.43).

$$Dissimilarity = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i, j)|i - j| \quad (3.43)$$

c. *Homogeneity (HOM)/Inverse Difference Moment (IDM)*

Homogeneity (HOM) dikenal juga sebagai *Inverse Difference Moment (IDM)*. *Homogeneity* bernilai tinggi jika *grey level* dari setiap pasangan piksel adalah sama. Hal ini terjadi ketika area *homogeneity* lokal ada di dalam skala panjang *spatial*. *Homogeneity* berisi nilai yang mengukur kedekatan distribusi elemen dalam GLCM ke diagonal GLCM. Jangkauan nilai *Homogeneity* = [0 1]. *Homogeneity* bernilai 1 untuk GLCM diagonal. *Homogeneity* dihitung dengan persamaan (3.44):

$$Homogeneity = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{G(i, j)}{1 + (i - j)^2} \quad (3.44)$$

d. *Similarity (SIM)/Inverse Difference Moment Normalized (IDMNC)*

Similarity adalah sebuah ukuran dari *homogeneity* tingkat pertama. *Similarity* juga dikenal sebagai *Inverse Difference Moment Normalized (IDMNC)*. *Similarity* biasanya disebut *homogeneity* yang mengukur *homogeneity* lokal dari suatu area. *Similarity* dihitung dengan persamaan (3.45).

$$\text{Similarity} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{G(i,j)}{1 + |i - j|} \quad (3.45)$$

e. *Angular Second Moment (ASM)/Energy*

ASM mengukur *homogeneity* area. *ASM* sering kali didefinisikan sebagai *Energy* atau *Uniformity*. *ASM* merupakan kebalikan dari *Entropy*. *ASM* berisi jumlah elemen kuadrat dalam *Gray Level Co-occurrence Matrices* (GLCM). Jangkauan nilai $ASM = [0 \ 1]$. *ASM* bernilai 1 untuk area konstan. *ASM* dihitung dengan persamaan (3.46).

$$ASM = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i,j)^2 \quad (3.46)$$

f. *Entropy (ENT)*

Entropy mengukur keacakan dari distribusi *grey level* pada area. *Entropy* akan bernilai tinggi jika *grey level* terdistribusi dengan acak di seluruh area. *Entropy* dihitung dengan persamaan (3.47).

$$ENT = - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i,j) \cdot \ln[G(i,j)] \quad (3.47)$$

g. *Mean (sum mean) (μ)*

Mean adalah nilai rata-rata piksel yang terdapat pada area. *Mean* (μ) dapat dihitung dengan persamaan (3.48). μ_x adalah *mean* untuk setiap baris yang dihitung dengan persamaan (3.49). μ_y adalah *mean* untuk setiap kolom yang dihitung dengan persamaan (3.50).

$$\mu = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} iG(i,j) \quad (3.48)$$

$$\mu_x = \sum_{i=0}^{n-1} iG_x(i) \quad (3.49)$$

$$\mu_y = \sum_{j=0}^{n-1} jG_y(j) \quad (3.50)$$

h. *Variance and standard deviation (VAR dan SD)*

Variance (σ^2) menggambarkan sebaran nilai *grey level* pada sebuah area objek. *Variance* dihitung dengan cara mencari sebaran distribusi nilai pada matrik GLCM. Nilai *variance* berbanding lurus dengan luas sebaran *grey level*. *Variance* dapat dihitung dengan persamaan (3.51). *Variance* untuk setiap baris (σ_x^2) dihitung dengan persamaan (3.52). *Variance* untuk setiap kolom (σ_y^2) dihitung dengan persamaan (3.53). *Standard Deviation* adalah akar dari *variance*. *Standard Deviation* (σ), *Standard Deviation* untuk setiap baris (σ_x) dan *Standard Deviation* untuk setiap kolom (σ_y) dihitung dengan persamaan (3.54 - 3.56).

$$VAR = \sigma^2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i, j)(i - \mu)^2 \quad (3.51)$$

$$\sigma_x^2 = \sum_{i=0}^{n-1} (G_x(i) - \mu_x(i))^2 \quad (3.52)$$

$$\sigma_y^2 = \sum_{j=0}^{n-1} (G_y(j) - \mu_y(j))^2 \quad (3.53)$$

$$\sigma = \sqrt{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i, j)(i - \mu)^2} \quad (3.54)$$

$$\sigma_x = \sqrt{\sum_{i=0}^{n-1} (G_x(i) - \mu_x(i))^2} \quad (3.55)$$

$$\sigma_y = \sqrt{\sum_{j=0}^{n-1} (G_y(j) - \mu_y(j))^2} \quad (3.56)$$

i. *Maximum probability (MaxP)*

Maximum probability (MaxP) menghitung *grey level*, yang mempunyai probabilitas paling tinggi dalam GLCM. max_i^m adalah nilai tertinggi i, max_j^m adalah nilai tertinggi j. *MaxP* akan bernilai tinggi jika terjadinya pasangan piksel yang paling dominan adalah tinggi. *MaxP* dihitung dengan persamaan (3.57).

$$MaxP = max_i^m max_j^n G(i, j) \quad (3.57)$$

j. *Correlation (COR)*

Correlation (COR) adalah ukuran ketergantungan nilai antara piksel yang satu dengan piksel yang lain. Ketika skala tekstur lokal lebih tinggi dari jarak *spatial*, maka *correlation* bernilai tinggi, begitu pula sebaliknya. *Correlation* berisi ukuran korelasi piksel ke tetangganya yang meliputi keseluruhan gambar. Jangkauan nilai *Correlation* = [-1 1]. *Correlation* bernilai 1 atau -1 untuk citra berkolerasi sempurna atau negatif. *Correlation* bernilai NaN untuk citra konstan. *Correlation* dihitung dengan persamaan (3.58).

$$Correlation = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i, j) [(i - \mu_x)(j - \mu_y)]}{(\sigma_x \sigma_y)} \quad (3.58)$$

k. *Diagonal moment (DM)*

Diagonal moment (DM) adalah ukuran perbedaan *correlation* untuk *grey level* tinggi dan rendah. *Diagonal moment (DM)* dihitung dengan persamaan (3.59).

$$DM = \sum_{i=0}^{n-1} j \sum_{j=0}^{n-1} (0,5G(i, j)|i - j|)^{0,5} \quad (3.59)$$

l. *Second diagonal moment (SDM)*

Second diagonal moment (SDM) adalah ukuran momen kedua dari perbedaan *correlation* untuk *grey level* tinggi dan rendah. *Second diagonal moment (SDM)* dapat dihitung dengan persamaan (3.60).

$$SDM = \sum_{i=0}^{n-1} j \sum_{j=0}^{n-1} (0,5G(i, j)|i - j|) \quad (3.60)$$

m. *Coefficient of variation (CVAR)*

Coefficient of variation (CVAR) adalah ukuran dispersi transisi *grey level* yang dihitung dalam kaitannya dengan nilai rata-rata. *Coefficient of variation (CVAR)* dihitung dengan persamaan (3.61).

$$CVAR = \frac{\sigma}{\mu} \quad (3.61)$$

n. *Sum entropy (SENT)*

Sum entropy (SENT) adalah ukuran jumlah perbedaan *micro* (lokal) citra. *Sum entropy (SENT)* dapat dihitung dengan persamaan (3.62).

$$SENT = - \sum_{i=2}^{2n} G_{x+y}(i) \ln[G_{x+y}(i)] \quad (3.62)$$

o. *Sum variance (SVAR)*

Sum variance (SVAR) mengungkapkan heterogenitas spasial citra. *Sum variance (SVAR)* dapat dihitung dengan persamaan (3.63).

$$SVAR = - \sum_{i=2}^{2n} G_{x+y}(i) (i - SENT)^2 \quad (3.63)$$

p. *Difference entropy (DENT)*

Difference entropy (DENT) adalah ukuran jumlah perbedaan *local variability (micro)* citra. *Difference entropy (DENT)* dapat dihitung dengan persamaan (3.64).

$$DENT = - \sum_{i=0}^{n-1} G_{x-y}(i) \ln[G_{x-y}(i)] \quad (3.64)$$

q. *Difference variance (DVAR)*

Difference variance (DVAR) adalah ukuran *local variability*. *Difference variance (DVAR)* dapat dihitung dengan persamaan (3.65).

$$DVAR = - \sum_{i=0}^{n-1} G_{x-y}(i) (i - DENT)^2 \quad (3.65)$$

r. *Cluster shade (CSH)/Skewness*

Cluster shade (CSH)/Skewness adalah ukuran asimetri dalam distribusi statistik, di mana kurva terdistorsi atau miring baik ke kiri atau ke kanan. *Cluster shade (CSH)/Skewness* dapat diukur untuk menentukan sejauh mana distribusi berbeda dari distribusi normal. *Cluster shade (CSH)* dapat dihitung dengan persamaan (3.66).

$$CSH = - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i,j)[(i+j) - (\mu_x + \mu_y)]^3 \quad (3.66)$$

s. *Cluster prominence (CPR)*

Cluster prominence (CPR) adalah sebuah versi dua dimensi dari level gray histogram kurtosis, yang merupakan sebuah ukuran dari bagaimana peak adalah sebuah distribusi. *Cluster prominence (CPR)* dapat dihitung dengan persamaan (3.67).

$$CPR = - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i,j)[(i+j) - (\mu_x + \mu_y)]^4 \quad (3.67)$$

t. *Sum average (SAVR)*

Sum average (SAVR) mengukur relasi area antara *clear* dan *dense* di dalam sebuah citra. *Sum average (SAVR)* dapat dihitung dengan persamaan (3.68-3.69).

$$SVAR = \sum_{i=0}^{n-1} iG_{x+y}(i) \quad (3.68)$$

dengan

$$G_{x+y}(i+j) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i,j) \quad (3.69)$$

u. *Mean correlation 1 (MCOR1)*

Ciri Mean correlation 1 (MCOR1) juga dikenal sebagai *Information Measure of Correlation 1 (IMCI)*. Dalam ciri ini, dua turunan array dari GLCM harus digunakan. Array pertama merepresentasikan ringkasan baris di dalam GLCM, sedangkan array kedua merepresentasikan ringkasan dari

kolom di dalam GLCM. *Mean correlation 1 (MCOR1)* dapat dihitung dengan persamaan (3.70-3.73).

$$MCOR1 = \frac{ENT - HXY1}{\max(HX, HY)} \quad (3.70)$$

dengan

$$HXY1 = - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G(i, j) \ln[G_x(i), G_y(j)] \quad (3.71)$$

$$G_x(i) = \sum_{j=0}^{n-1} G(i, j) \quad (3.72)$$

$$G_y(j) = \sum_{i=0}^{n-1} G(i, j) \quad (3.73)$$

HX dan HY adalah entropi dari $P_x(i)$ dan $P_y(j)$.

v. *Mean correlation 2 (MCOR2)*

Ciri *Mean correlation 2 (MCOR2)* adalah bentuk lain untuk menghitung *mean correlation*. Ciri *Mean correlation 2 (MCOR2)* juga dikenal sebagai *Information Measure of Correlation 2 (IMC2)*. *Mean correlation 2 (MCOR2)* dapat dihitung dengan persamaan (3.74-3.75).

$$MCOR2 = \sqrt{1 - \exp^{-2(HXY2-ENT)}} \quad (3.74)$$

dengan

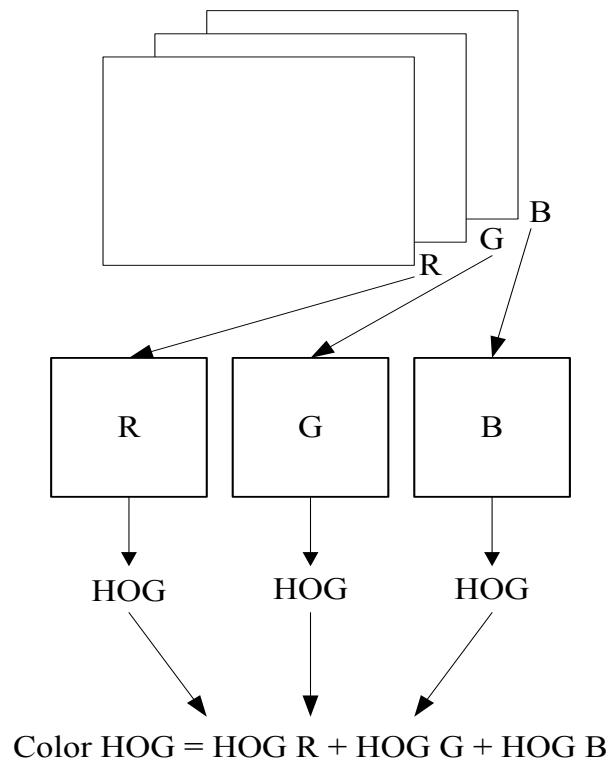
$$HXY2 = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} G_x(i)G_y(j) \ln[G_x(i)G_y(j)] \quad (3.75)$$

3.4.3 Ciri Warna

Ciri warna diperoleh dari citra berwarna.

1. *ColorHOG*

ColorHOG adalah HOG yang di dapat dari area objek berwarna (Srivastava dan Singh, 2014). *ColorHOG* dapat di gunakan untuk mendeteksi objek (Goto dkk., 2013). Gambar 3.11 menunjukkan ilustrasi *colorHOG*.



Gambar 3.11 Ilustrasi Color HOG

2. Color Moments

Color Moments adalah ukuran yang dapat membedakan area berdasarkan ciri warnanya. Konsep dasar di balik momen warna terletak pada asumsi bahwa distribusi warna dalam area dapat diartikan sebagai distribusi probabilitas (Kavya dan Harisha, 2015). *Color Moments* dapat digunakan untuk klasifikasi (van de Sande dkk., 2008). Tiga momen warna dapat didefinisikan:

Mean: *Mean* adalah nilai rata-rata warna dalam area. *Mean* dapat dihitung dengan persamaan (3.76):

$$E_i = \sum_{j=1}^N \frac{1}{N} P_{ij} \quad (3.76)$$

Standard Deviation (Std): *Standard Deviation* adalah akar kuadrat dari varians distribusi. *Standard Deviation* dapat dihitung dengan persamaan (3.77):

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^{j=1} (P_{ij} - E_i)^2\right)} \quad (3.77)$$

Skewness: *Skewness* adalah ukuran tingkat asimetri dalam distribusi. *Skewness* dapat dihitung dengan persamaan (3.78):

$$S_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^{j=1} (P_{ij} - E_i)^3\right)} \quad (3.78)$$

Color Moments berisi perhitungan dari *mean* dan *standar deviation* - dua momen yang pertama - dari komponen warna R,G dan B yaitu MeanR, StdR, MeanG, StdG, MeanB, StdB (Vinayak dan Jindal, 2017).

3. *Color Histogram*

Color Histogram (histogram warna) adalah ciri yang paling banyak digunakan karena kedekatannya dibandingkan dengan ciri lain dan yang paling penting karena kemudahannya dalam mengekstrak area. *Color Histogram* menggambarkan distribusi warna menggunakan sekumpulan *bin* (Kavya dan Harisha, 2015). *Color Histogram* berfungsi sebagai representasi efektif dari konten warna area. *Color Histogram* mudah dikomputasi. *Color Histogram* adalah grafik yang berisi kemunculan setiap nilai intensitas yang ditemukan di area tersebut. *Color Histogram* diperoleh dengan menghitung semua piksel area yang memiliki nilai intensitas tersebut. Untuk area grayscale 8-bit ada 256 kemungkinan intensitas berbeda. Sehingga histogram akan menampilkan secara grafis 256 nilai grayscale yang menunjukkan distribusi piksel di antara angka-angka itu. Histogram bisa juga diambil dari area berwarna. *Color Histogram* adalah representasi distribusi warna dalam suatu area. Ini adalah deskripsi statistik standar dari distribusi warna dalam hal frekuensi kejadian dari berbagai daerah dalam ruang warna. Ada dua jenis *color histogram* (histogram warna) yaitu *Global Color Histogram* (GCH) dan *Local Color Histogram* (LCH). GCH mewakili satu area utuh dengan histogram warna tunggal dan area akan dikodekan dengan histogram warna. Jarak antara dua area akan ditentukan oleh jarak diantara histogram warnanya. Sementara LCH

membagi area menjadi blok tetap dan mengambil histogram warna dari masing-masing blok tersebut. Sementara membandingkan dua jarak area dihitung menggunakan histogram antara area yang satu dengan area yang lain. Jarak antara dua area akan ditentukan oleh penjumlahan dari semua jarak ini. Jadi ketika membandingkan GCH mungkin hasil yang diperoleh tidak konsisten dalam hal kesamaan area ketika dibandingkan dengan LCH. Kuantisasi warna adalah proses mengurangi jumlah warna berbeda yang digunakan dalam suatu area, biasanya area baru sedapat mungkin mirip secara visual dengan area aslinya. Untuk area warna yang sebenarnya, jumlah jenis warna sampai $2^{24} = 16777216$, sehingga ekstraksi langsung ciri warna dari warna asli akan menyebabkan komputasi yang besar. Untuk mengurangi komputasi, tanpa menurunkan kualitas area yang signifikan, beberapa warna perwakilan di ekstrak untuk mewakili area, yang akan mengurangi ruang penyimpanan dan meningkatkan kecepatan pemrosesan. (Pushpalatha dkk., 2013). *Color Histogram / RGB Histogram* dapat digunakan untuk deteksi (Gandhi dan Trivedi, 2007) dan klasifikasi (van de Sande dkk., 2008; Gandhi dan Trivedi, 2007; Wang dkk., 2014).

4. *Color Correlogram*

Color Correlogram (kolerasi warna) adalah salah satu pendeskripsi warna spasial yang paling menjanjikan. Yang menarik dari ciri ini adalah:

- a. ini mencakup kolerasi spasial warna
- b. dapat digunakan untuk menggambarkan distribusi global korelasi spasial lokal warna
- c. mudah untuk melakukan perhitungan
- d. ukuran cirinya cukup kecil

Color Correlogram bertujuan untuk mengkarakterisasi distribusi warna piksel dan kolerasi spasial pasangan warna. Dimensi pertama dan kedua dari histogram tiga dimensi adalah warna dari setiap pasangan piksel dan dimensi ketiga adalah jarak spasialnya. *Color Correlogram* adalah tabel yang di indeks oleh pasangan warna, dimana k-entry untuk (i,j) menentukan probabilitas untuk menemukan piksel warna "j" pada jarak "k" dari piksel i pada area. I

merepresentasikan seluruh himpunan piksel area. $c(j)$ merepresentasikan himpunan piksel yang warnanya adalah $c(i)$ (Pushpalatha dkk., 2013). *Color Correlogram* dapat dihitung dengan persamaan (3.79):

$$\gamma_{i,j}^{(k)} \triangleq Pr_{p_1 \in I_{c(i)}, p_2 \in I} [p_2 \in I_{c(j)} | |p_1 - p_2| = k] \quad (3.79)$$

dengan $i, j \in \{1, 2, \dots, N\}$, $k \in \{1, 2, \dots, d\}$ dan $|p_1 - p_2|$ adalah jarak antara piksel p_1 dan p_2 . *Color Correlogram* dapat digunakan untuk klasifikasi kendaraan berwarna (Wang dkk., 2014).

5. *ColorEnergy*

ColorEnergy adalah *Energy* dekomposisi wavelet dua dimensi (Jialiang dkk., 2016). Transformasi wavelet memiliki ciri yang baik untuk *space-fequency localization* dan multiresolusi. Alasan utama popularitas transformasi wavelet terletak pada kerangka teoritisnya yang lengkap, fleksibilitasnya yang besar dalam pemilihan hal yang mendasar dan kompleksitas perhitungannya rendah. *Decomposition Wavelet Transform* (DWT) untuk area objek sebagai sinyal 2D (2 dimensi) dapat diturunkan dari 1D (1 dimensi) DWT. Cara termudah untuk memperoleh skala dan fungsi wavelet untuk dua dimensi adalah dengan mengalikan 2 fungsi 1D (Kakarwal dan Deshmukh, 2010). Fungsi penskalaan untuk DWT 2D dapat diperoleh dengan mengalikan 2 fungsi skala 1D yang dihitung dengan persamaan (3.80):

$$\phi(x,y) = \phi(x) \phi(y) \quad (3.80)$$

Fungsi wavelet untuk DWT 2D dapat diperoleh dengan mengalikan 2 fungsi wavelet. Untuk kasus 2D, ada tiga fungsi wavelet yang secara jelas dapat dilihat secara horisontal yang dapat dihitung dengan persamaan (3.81):

$$\psi(I) (x,y) = \phi(x) \psi(y) \quad (3.81)$$

Secara vertikal yang dapat dihitung dengan persamaan (3.82):

$$\psi(II) (x,y) = \psi(x) \phi(y) \quad (3.82)$$

dan arah diagonal yang dapat dihitung dengan persamaan (3.83):

$$\psi(III) (x,y) = \psi(x) \psi(y) \quad (3.83)$$

ColorEnergy digunakan untuk pengenalan kendaraan (Jialiang dkk., 2016), klasifikasi kendaraan (Zhang dkk., 2008) dan pelacakan kendaraan (Srikaew dkk., 2009).

3.5 Traffic Parameter

Traffic parameter atau parameter lalu lintas adalah suatu tolok ukur yang dipakai sebagai patokan dalam sistem transportasi lalu lintas (Klein dan Kelley, 1996; Board, 2000). Parameter lalu lintas merupakan karakteristik arus lalu lintas yang dapat diklasifikasikan sebagai berikut:

- a. *Quantify Measures/Ukuran Kuantitas*: Untuk mengetahui berapa banyak atau pada tingkat apa lalu lintas bergerak atau menunggu untuk pindah.
- b. *Quality Assessment Measures/Ukuran Penilaian Kualitas*: Untuk mengetahui berapa baik pergerakan lalu lintas.
- c. *Movement Measures/Ukuran Pergerakan*: Untuk mengetahui dari mana lalu lintas itu datang dan kemana lalu lintas itu pergi.
- d. *Composition/Classification Measures/Ukuran Komposisi/Klasifikasi*: Untuk mengetahui jenis lalu lintas yang bergerak.

Parameter lalu lintas terutama terdiri dari arus lalu lintas, kepadatan lalu lintas, kecepatan dan panjang antrian. Parameter lalu lintas digunakan untuk mengendalikan lalu lintas supaya lalu lintas tertib dan lancar. Ini tercermin dari parameternya (Deng dan Li, 2012).

Karakteristik ini dapat diamati dan dipelajari di tingkat mikroskopis dan makroskopis. Dari Tabel 3.1 dapat dilihat bahwa penggolongan parameter lalu lintas terdiri dari:

1. Parameter mikroskopis, mencirikan perilaku setiap kendaraan dalam arus lalu lintas yang saling mempengaruhi satu sama lain. Parameter mikroskopis meliputi *time headways* (waktu antara), *individual speeds* (kecepatan individu), dan *distance headways* (jarak antara).
2. Parameter makroskopis, mencirikan arus lalu lintas sebagai satu kesatuan (sistem) sehingga gambaran operasional sistem dapat diperoleh secara

menyeluruh. Parameter makroskopis meliputi *flow rates* (tingkat arus lalu lintas), *average speeds* (kecepatan rata-rata), *density rates* (tingkat kepadatan).

Tabel 3.1 Karakteristik Mendasar dari Arus Lalu Lintas (Board, 2000)

Karakteristik lalu lintas	Mikroskopis (unit individu)	Makroskopis (unit grup)
<i>Flow</i> (Arus)	<i>Time headways</i> (waktu antara)	<i>Flow rates</i> (tingkat arus lalu lintas)
<i>Speed</i> (Kecepatan)	<i>Individual speeds</i> (kecepatan individu)	<i>Average speeds</i> (kecepatan rata-rata)
<i>Density</i> (Kepadatan)	<i>Distance headways</i> (jarak antara)	<i>Density rates</i> (tingkat kepadatan)

Menurut Board (2000), May (1990) dan Mathew & Rao (2006), ada 3 variabel dasar yaitu *volume* atau *flow rate* (tingkat arus), *speed* (kecepatan) dan *density* (kepadatan) yang dapat digunakan untuk menggambarkan parameter lalu lintas di jalan raya. *Volume* atau *flow rate* (tingkat arus) adalah parameter umum yang digunakan untuk mengukur kondisi lalu lintas yang aliran lalu lintasnya terganggu dan terputus. Sedangkan *speed* (kecepatan) dan *density* (kepadatan) khusus untuk mengukur kondisi lalu lintas yang aliran lalu lintasnya terganggu.

Volume atau *flow rate* (tingkat arus lalu lintas) adalah jumlah kendaraan yang melintas pada suatu titik di ruas jalan dalam interval waktu tertentu. Ada perbedaan antara *volume* dan *flow rate*. *Volume* tidak membedakan arah dan lajur serta diukur pada interval waktu yang panjang (lama). *Volume* dapat dinyatakan dalam periode tahunan, harian, jam atau beberapa jam. *Flow rate* membedakan arah dan lajur serta diukur pada interval waktu yang pendek (kurang dari 1 jam; biasanya 15 menit). Board (2000) menyatakan bahwa *peak flow rate* dan *volume per jam* menghasilkan *peak-hour factor* (PHF) yaitu rasio dari *hourly volume* (total volume per jam) dan *peak flow rate within the hour* (arus lalu lintas tertinggi dalam 1 jam). PHF dapat dihitung dengan persamaan (3.84):

$$PHF = \frac{\text{total volume per jam}}{\text{ arus lalu lintas tertinggi (dalam jam)}} \quad (3.84)$$

Jika digunakan periode 15 menit maka menurut Board (2000) PHF dapat dihitung dengan persamaan (3.85):

$$PHF = \frac{V}{4 \times V_{15}} \quad (3.85)$$

dengan: PHF= *peak-hour factor*,

V = total volume per jam (jumlah kendaraan / jam), dan

V₁₅ = volume arus lalu lintas tertinggi dalam 15 menit dari rentang pengamatan 1 jam

Menurut Board (2000) jika PHF diketahui maka arus lalu lintas tertinggi dapat dihitung dengan persamaan (3.86):

$$v = \frac{V}{PHF} \quad (3.86)$$

dengan: v = arus lalu lintas tertinggi dalam 15 menit (jumlah kendaraan / 15 menit)

V = *peak-hour volume* (jumlah kendaraan / jam), dan

PHF= *peak-hour factor*.

Speed (kecepatan) didefinisikan sebagai nilai pergerakan kendaraan dihitung berdasarkan jarak persatuan waktu, umumnya diukur kilometer per jam (km/jam). *Average travel speed* (kecepatan tempuh rata-rata) dihitung berdasarkan panjang segmen jalan raya dibagi waktu tempuh rata-rata kendaraan yang melintas. Menurut Board (2000), kecepatan tempuh rata-rata dapat dihitung dengan persamaan (3.87):

$$S = \frac{nL}{\sum_{i=1}^n t_i} = \frac{L}{\frac{1}{n} \sum_{i=1}^n t_i} = \frac{L}{t_a} \quad (3.87)$$

dengan: S = kecepatan tempuh rata-rata (km/jam)

L = panjang segmen jalan (km)

t_i = waktu tempuh kendaraan ke-i yang melintasi jalan (dalam jam)

n = jumlah pengamatan

t_a = $\frac{1}{n} \sum_{i=1}^n t_i$ = waktu tempuh rata-rata yang melintasi L (dalam jam)

Average running speed (kecepatan rata-rata kendaraan bergerak) – arus lalu lintas diukur berdasarkan berapa waktu tempuh kendaraan yang melintasi suatu area di jalan raya. Perhitungannya adalah panjang segmen dibagi dengan waktu rata-rata kendaraan bergerak yang melintasi segmen.

Time mean speed - kecepatan rata-rata kendaraan yang diamati saat melewati titik di jalan raya, disebut *average spot speed* (kecepatan rata-rata di

jalan). Kecepatan kendaraan secara individu yang melintasi titik dicatat dan dihitung rata-ratanya.

Free-flow speed - kecepatan rata-rata kendaraan, diukur dalam kondisi lalu lintas dengan volume rendah. Untuk kondisi lalu lintas yang lancar, kecepatan tempuh rata-rata kendaraan sama dengan jarak tempuh dibagi waktu tempuh.

Traffic parameter di persimpangan digunakan sebagai data referensi yang dapat memprogram strategi sinyal lalu lintas dan tata letak jaringan lalu lintas untuk manajemen lalu lintas. *Traffic parameter* di persimpangan antara lain: *traffic flow*, *time occupancy* dan *space mean speed*. Pengukuran traffic parameter ini diberikan sebagai berikut:

Traffic flow (TF) / arus lalu lintas adalah jumlah lalu lintas obyek yang melewati zona deteksi selama waktu tertentu. Menurut Board (2000), *traffic flow* (TF) dapat dihitung dengan persamaan (3.88):

$$TF_i = \frac{S_v}{T_{dur}} \quad (3.88)$$

dengan: i = index dari jalur lalu lintas,

S_v = jumlah kendaraan yang melewati zona deteksi

T_{dur} = waktu pengamatan

Time occupancy ratio (TOR) adalah rasio akumulatif waktu kendaraan yang lewat melalui zona deteksi dibagi dengan durasi waktu. Menurut Board (2000), TOR dapat dihitung dengan persamaan (3.89):

$$TOR = \frac{\sum_{i=1}^{S_v} T_{pass,i}}{T_{dur}} \quad (3.89)$$

dengan $T_{pass,i}$ menunjukkan waktu kendaraan untuk melewati zona deteksi. Menurut Board (2000), $T_{pass,i}$ dapat dihitung dengan persamaan (3.90):

$$T_{pass,i} = \frac{(K_{frame_{out}} - K_{frame_{in}})}{Frame_rate} \quad (3.90)$$

dengan K_{frame_in} dan K_{frame_out} menunjukkan *frame serial number* ketika kendaraan ke- i datang dan meninggalkan zona deteksi. *Frame_rate* adalah *frame rate* kamera.

Space mean speed (SMS) - didefinisikan sebagai total jarak perjalanan

dibagi dengan total waktu tempuh. *space mean speed* dapat dihitung dengan persamaan (3.91) (Shi dkk., 2009; Zou dkk., 2009):

$$SMS = \frac{TF \cdot (L_y + L_{zone})}{TOR} \quad (3.91)$$

dengan : TF = *traffic flow*,

L_y = *average length of vehicle* / panjang rata-rata kendaraan,

L_{zone} = *length of detection zone* / panjang zona deteksi.

Difference of IN/OUT traffic flow (DIO) adalah perbedaan TF_{in} dan TF_{out} . TF_{in} dan TF_{out} berarti arus lalu lintas yang datang dan pergi dari persimpangan pada satu periode sinyal lalu lintas. Ini merupakan parameter penting untuk mendeteksi kemacetan lalu lintas. Jika DIO melebihi nilai *threshold* yang telah di setting di awal, maka berarti kemacetan lalu lintas terjadi pada persimpangan. Menurut (Shi dkk., 2009; Zou dkk., 2009), perhitungan *difference of IN/OUT traffic flow (DIO)* dapat dihitung dengan persamaan (3.92):

$$DIO = TF_{in} - TF_{out} \quad (3.92)$$

Density (kerapatan) adalah jumlah kendaraan yang menempati suatu panjang jalan atau lajur kendaraan per km atau kendaraan per km per lajur. Menurut (Shi dkk., 2009; Zou dkk., 2009), *density* dapat dihitung dengan persamaan (3.93):

$$D = \frac{v}{S} \quad (3.93)$$

dengan: v = *flow rate* (jumlah kendaraan/jam)

S = *average travel speed* / kecepatan tempuh rata-rata (km/jam), dan

D = *density* (jumlah kendaraan / km)

Morris & Trivedi (2008) membuat sistem yang dapat memberikan informasi tentang arus (jumlah kendaraan / waktu), densitas (jumlah kendaraan / jarak), dan kecepatan (dalam mil per jam) yang dapat diperkirakan dalam interval 30 detik dengan jeda waktu rata-rata diatas 5 detik. Arus lalu lintas secara statistik dapat dihasilkan dengan menghitung jumlah kendaraan yang melintas dengan perubahan setiap 30 detik. Kepadatan adalah jumlah rata-rata kendaraan dalam tampilan kamera dinormalisasi dengan panjang jalan. Kecepatan adalah kecepatan estimasi rata-rata kendaraan. Perhitungan untuk kepadatan dan kecepatan diperoleh dari

perhitungan kalibrasi secara manual.

3.6 Pemampatan Video

Pemampatan video bertujuan untuk meminimalkan ukuran video di dalam tempat penyimpanan (Gibson, 2000). Cara pemampatan video dengan mengurangi duplikasi data di dalam video. Hasil pemampatan video dapat mempercepat proses pengiriman video. Ada dua teknik yang dapat dilakukan dalam melakukan pemampatan video yakni *Lossless Compression* dan *Lossy Compression*. *Lossless Compression* merupakan pemampatan video dimana hasil dekompresi dari video yang sudah dimampatkan sama dengan video aslinya. Tidak ada informasi yang hilang. Namun rasio pemampatan video dengan metode ini sangat rendah. Contoh pemampatan yang menggunakan *Lossless Compression* adalah *Run Length Encoding* (RLE), *Entropy Encoding* (Huffman, Aritmatik) dan *Adaptive Dictionary Based* (LZW). *Lossy Compression* adalah pemampatan video yang hasil dekompresinya tidak sama dengan video aslinya karena ada informasi yang hilang, tetapi masih bisa ditoleransi oleh persepsi mata. Mata tidak dapat membedakan perubahan kecil pada video. Metode ini menghasilkan rasio pemampatan yang lebih tinggi daripada metode lossless. Contoh pemampatan yang menggunakan *Lossy Compression* adalah *color reduction*, *chroma subsampling*, dan *transform coding* seperti transformasi *Fourier* dan *Wavelet*.

Proses pemampatan merupakan proses mengkodekan video (*encode*) sehingga diperoleh ukuran video lebih kecil dari ukuran semula. Format video beragam yaitu .AVI (*Audio Video Interleave*), .FLV (*Flash Video*), .MP4 (*Moving Picture 4*), .MOV (*Apple's QuickTime container format*), .VOB (*DVD Video Object*), .MKV (*Mastroska*), .MPEG (*Moving Picturea Expert Group*) dan .3GP atau 3GPP (*3rd Generation Partnership Project*). Video yang sudah dimampatkan disimpan dalam file dengan format tertentu misalnya .MPEG. Sedangkan proses dekompresi adalah proses untuk menguraikan video yang dimampatkan untuk dikembalikan lagi (*decoding*) menjadi video yang tidak dimampatkan. Algoritme pemampatan yang baik adalah algoritme yang membutuhkan waktu untuk pemampatan dan dekompresi paling sedikit sehingga paling cepat diperoleh

hasilnya. Metode pemampatan yang baik adalah metode pemampatan yang mampu memampatkan file video menjadi file yang berukuran paling minimal. Semakin kecil persentase video yang dimampatkan, semakin bagus kualitas hasil pemampatan tersebut. Metode pemampatan yang baik adalah metode pemampatan yang mampu mengembalikan video hasil pemampatan menjadi video semula tanpa kehilangan informasi apapun. Walaupun ada informasi yang hilang akibat pemampatan, hal itu diperkecil seminimal mungkin. Format video hasil pemampatan yang baik adalah yang cocok dengan kebutuhan pengiriman dan penyimpanan data.

3.7 eXtensible Markup Language (XML)

eXtensible Markup Language (XML) adalah bahasa *markup* untuk keperluan umum yang disarankan oleh *World Wide Web Consortium* (W3C). Dalam kaitannya dengan bahasa pemrograman, kata “*extensible*” berarti pengembang dapat mengembangkan kemampuannya. Informasi dalam XML disimpan di dalam *tag*. *Markup* berarti sebuah bahasa yang dibangun dengan menggunakan *tag* yang mengelilinginya atau *mark up text* (Bray dkk., 2000). W3C adalah sebuah komunitas internasional yang secara terbuka membangun standard untuk menjamin perkembangan *Web* untuk jangka panjang. XML dikembangkan oleh kelompok kerja XML (awalnya dikenal sebagai *Editorial Review Board SGML*). XML dibentuk oleh W3C pada tahun 1996 dipimpin oleh Jon Bosak dari *Sun Microsystems* dengan partisipasi aktif dari *XML Special Interest Group*.

XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standar layaknya *heading*, *paragraph*, tabel dan lain sebagainya. File XML berbentuk teks sehingga bila diperlukan maka file XML dapat dibaca tanpa memerlukan bantuan *software* khusus. Hal ini memudahkan pengembang aplikasi yang menggunakan XML untuk *men-debug* programnya. Penulisan dokumen XML harus memenuhi aturan standar atau *well formed*. XML menjelaskan tentang suatu klas objek data yang disebut dokumen

XML dan yang sebagian besar menjelaskan perilaku program komputer yang diprosesnya (Bray dkk., 2000; Bray dkk., 2006; Bray dkk., 2013). Sebuah dokumen XML dikatakan *well formed* jika dokumen berisi satu atau lebih elemen, dokumen berisi *single root* elemen yang berisi beberapa elemen di dalamnya, tiap elemen harus ditutup, tag pembuka dan penutup harus sama (*case sensitive*), nilai *attribute* berada di dalam tanda petik. XML adalah profil aplikasi atau bentuk terbatas dari *Standard Generalized Markup Language* (SGML). Adapun konstruksi dokumen XML sesuai dengan dokumen SGML. Dokumen XML terdiri dari unit penyimpanan yang disebut entitas, yang berisi data *parsed* atau data *unparsed*. Data *parsed* terdiri dari karakter-karakter dimana beberapa karakternya berbentuk data karakter dan sisanya berbentuk *markup*. *Markup* mengkodekan deskripsi tata letak penyimpanan dan struktur logis. XML menyediakan mekanisme untuk menyelesaikan kendala pada tata letak penyimpanan dan struktur logis. XML memerlukan *parser* yang menerjemahkan antara data dan *tag* untuk menampilkan informasi. Pada XML, data dapat disimpan sebagai atribut dan isi elemen yang diletakkan diantara *tag* pembuka dan *tag* penutup. XML menggunakan elemen yang ditandai dengan *tag* pembuka (diawali dengan '<' dan diakhiri dengan '>'), *tag* penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam *tag* pembuka misal <form name="isidata">). *Tag* dan atribut pada XML dapat digunakan sesuai kebutuhan.

XML memiliki banyak kegunaan. Ada dua penggunaan utama untuk XML. Pertama, untuk menggambarkan data tingkat rendah, contohnya konfigurasi file. Kedua, untuk menambahkan metadata ke dokumen, misalnya bila ada kalimat tertentu yang ingin dimiringkan atau dicetak tebal (Fawcett dkk., 2012). Selain itu, XML digunakan dalam membuat dokumen *markup* untuk keperluan pertukaran data antar sistem yang beraneka ragam. Format penyimpanan data XML membuat platform ini independen dan berbagai aplikasi dapat memrosesnya dengan mudah. XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun tetap dapat dimodifikasi untuk menggambarkan isi dari dokumen.

Tipe file XML adalah XML, XSL, dan XSD. XML merupakan standar format dari file. XSL merupakan standar untuk memodifikasi data yang diimpor atau diekspor. XSD merupakan standar yang mendefinisikan struktur *database* dalam XML. Karakteristik XML adalah adaptif, mudah pemeliharaannya, sederhana, mudah dipindah-pindahkan (*portability*). Adaptif artinya dapat beradaptasi untuk membuat bahasa sendiri seperti *Microsoft* membuat bahasa MSXML atau *Macromedia* dalam mengembangkan MXML. Mudah dipindah-pindahkan (*portability*) artinya XML mempunyai kemudahan perpindahan (*portability*) informasi dari satu sistem ke sistem lain yang berbeda *platform* misalnya dari Windows ke *Unix*, atau dari PC ke *Macintosh* bahkan dari internet ke *handphone* dengan teknologi WAP (Carey, 2007, 2010; Parsons dan Oja, 2012).

3.8 Transmisi Data

Transmisi data adalah proses pengiriman data dari *transmitter* (pengirim) ke *receiver* (penerima) dari satu tempat ke tempat lain (Stallings, 2007). Transmisi data terjadi antara pengirim dan penerima melalui satu atau beberapa media transmisi. Media transmisi ada 2 jenis yaitu *guided media transmission* dan *unguided media transmission*. Dua jenis media transmisi ini komunikasinya berbentuk gelombang elektromagnetik. Pada *guided media transmission*, gelombang dikendalikan menggunakan jalur fisik. Contoh *guided media* adalah kabel *twisted pair*, kabel *coaxial* dan fiber optik. *Unguided media* menyediakan sarana untuk mentransmisikan gelombang elektronik namun tidak mengendalikannya. *Unguided media* biasa dikenal dengan *wireless*/media nirkabel. Contoh *unguided media* adalah udara, air, laut, tanah dan ruang fakum (RF).

Sebuah sistem transmisi dapat mengirimkan data secara *simplex*, *half-duplex* atau *full-duplex*. Pada transmisi *simplex*, data dikirim hanya satu arah. Satu *station* sebagai *transmitter*/pengirim dan yang lain sebagai *receiver*/penerima. Pada transmisi *half-duplex*, kedua *station* dapat mengirimkan data, tetapi pada saat yang sama hanya satu *station* yang dapat mengirimkan data. Waktu yang

digunakan untuk pergantian proses pengiriman dan penerimaan disebut *turnaround time*. Pada transmisi *full-duplex*, kedua station dapat mengirimkan data secara bersamaan. Pada transmisi *full-duplex*, proses transmisi lebih cepat dibandingkan *half-duplex* karena tidak ada “*turnaround time*”.

3.9 Histogram Equalization

Histogram Equalization (ekualisasi histogram) adalah suatu proses perataan histogram dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata (Hawlder dan Hossain, 2014; Xu dkk., 2014; Diego, 2014; Yadav dkk., 2014; Kadir dan Susanto, 2013). Untuk melakukan *histrogram equalization* diperlukan suatu fungsi distribusi kumulatif.

Elemen histogram yang sudah diekualisasi dapat dihitung dengan persamaan (3.94):

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j \quad k = 0, 1, 2, \dots, L - 1 \quad (3.94)$$

dengan:

MN = total jumlah piksel dalam citra

n_j = jumlah piksel yang memiliki intensitas sebesar r_k

L = jumlah tingkat intensitas yang mungkin muncul di dalam citra

s_k = elemen/nilai histogram yang di ekualisasi

3.10 Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) merupakan sebuah metode yang tahan terhadap perubahan lingkungan yang sebenarnya dengan memodelkan distribusi *background* dan mendeteksi objek yang bergerak pada video (Basri dkk., 2015). *Gaussian Mixture Model* (GMM) adalah model probabilistik yang mengasumsikan bahwa semua titik data dihasilkan dari campuran sejumlah distribusi Gaussian dengan parameter tertentu. Metode GMM dipopulerkan oleh Stauffer dan Grimson, (1999). GMM digunakan untuk memodelkan tiap piksel dari citra yang berurutan ke dalam distribusi model yang diperbaharui berdasarkan perbedaan

antara frame saat ini (t) dengan frame sebelumnya (t-1).

Menurut Stauffer dan Grimson, (1999), probabilitas untuk mengamati nilai piksel $P(X_t)$ dapat dihitung dengan persamaan (3.95) :

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \eta(X_t; \mu_{i,t}, \sigma_{i,t}^2) \quad (3.95)$$

dengan probabilitas kerapatan fungsi η menurut Shahbaz dkk., (2016) yang dapat dihitung dengan persamaan (3.96):

$$\eta(X_t; \mu_{i,t}, \sigma_{i,t}^2) = \frac{1}{\sigma_{i,t} \sqrt{2\pi}} e^{-\left(\frac{X_t - \mu_{i,t}}{2\sigma_{i,t}^2}\right)} \quad (3.96)$$

dengan nilai K adalah banyaknya distribusi / fungsi. X_t adalah nilai piksel pada waktu ke-t. $\omega_{i,t}$ adalah estimasi dari *weight* dari Gaussian ke-i pada waktu ke-t. $\mu_{i,t}$ adalah nilai *mean* dari Gaussian ke-i pada waktu ke-t. $\sigma_{i,t}^2$ adalah *variance* dari Gaussian ke-i pada waktu ke-t.

Kriteria keputusan untuk menandai piksel tertentu pada waktu ke-t sebagai *background* (latar objek) dan *foreground* (objek) dapat dihitung pada persamaan (3.97):

$$|X_t - \mu_{i,t}| > \lambda \sigma_{i,t} \quad (3.97)$$

dengan λ adalah *threshold* (nilai ambang) konstan sama dengan 2,5. Setiap nilai piksel baru, X_t , diperiksa terhadap distribusi K dari Gaussian yang ada sampai ditemukan kecocokan. Kecocokan didefinisikan nilai piksel berada dalam 2,5 standar deviasi dari suatu distribusi. Karena itu, piksel-piksel yang nilainya berada di lebih dari 2,5 standar deviasi yang jauh dari nilai rata-rata ditandai sebagai objek. Jika kecocokan ditemukan dengan salah satu komponen Gaussian K, piksel ditandai sebagai latar objek. Parameter komponen yang cocok adalah pembaruan yang dapat dihitung dengan persamaan (3.98-3.101):

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha \quad (3.98)$$

$$\mu_{i,t+1} = (1 - \beta)\mu_{i,t} + \beta X_{t+1} \quad (3.99)$$

$$\sigma_{i,t+1}^2 = (1 - \beta)\sigma_{i,t}^2 + \beta(X_{t+1} - \mu_{i,t})(X_{t+1} - \mu_{i,t})^T \quad (3.100)$$

$$\beta = \alpha \eta(X_{t+1}; \mu_{i,t+1}, \sigma_{i,t+1}^2) \quad (3.101)$$

dengan α adalah nilai *learning rate* konstan dan β adalah nilai *learning rate* kedua.

Jika tidak ditemukan kecocokan dengan salah satu nilai Gaussian K, maka piksel ditandai sebagai objek. Sebelum deteksi objek, parameter perlu diperbarui. Untuk distribusi yang tidak cocok, hanya *weight* yang diperbarui dalam persamaan (3.102-3.104):

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} \quad (3.102)$$

$$\mu_{i,t+1} = \mu_{i,t} \quad (3.103)$$

$$\sigma_{i,t+1}^2 = \sigma_{i,t}^2 \quad (3.104)$$

Bila tidak ada nilai distribusi K yang cocok dengan nilai piksel terkini, maka distribusi yang paling tidak mungkin diganti distribusi dengan nilai terkini yaitu *mean* (nilai rata-rata), *variance* (varians) yang tinggi dan *weight* sebelumnya yang rendah.

Nilai K pada persamaan (3.94) menyatakan banyaknya distribusi / fungsi yang dibutuhkan untuk *background model*. Pemilihan nilai K yang sesuai bergantung pada kompleksitas dari *setting* latar objeknya. Peningkatan nilai K bisa memberi hasil yang lebih baik tetapi hal ini menyebabkan peningkatan biaya komputasi. *Range* nilai K adalah 2 sampai 5. Nilai Gaussian paling kecil adalah 2. Satu untuk *background* (latar objek), satu untuk *foreground* (objek). Hasil eksperimen yang dikerjakan oleh Shahbaz dkk., (2016) memperlihatkan bahwa nilai K = 2 lebih cocok untuk ruang tertutup dan nilai K ≥ 4 direkomendasikan untuk ruang terbuka.

Learning rate α mengontrol seberapa cepat atau lambat komponen *Gaussian* melakukan kebaruan contohnya *weight*, *mean* dan *variance*. Alternatif lainnya, *learning rate* dapat memastikan berapa lama komponen *Gaussian* tertentu akan dipertahankan modelnya. Nilai yang ditetapkan tergantung pada kompleksitas latar objeknya. Misalnya, latar objek yang berubah secara perlahan membutuhkan nilai α yang rendah dan latar objek yang berubah secara bertahap membutuhkan nilai α yang tinggi. Waktu hidup τ dari komponen Gaussian tertentu dalam model latar objek dihitung dengan persamaan (3.105):

$$\tau = \frac{\log_{1-\alpha} 1/2}{f} \quad (3.105)$$

dengan f adalah frame per detik. Sebagai contoh, untuk $\alpha = 0,005$ dengan $f = 30$, waktu hidup $\tau = 4,6$ detik. Artinya jika sebuah objek bergerak menjadi diam,

setelah 4,6 detik, objek akan diubah menjadi latar objek. Nilai kecil seperti itu mungkin permasalahannya karena kameranya bergerak sehingga *noise* yang ditandai sebagai objek akan membutuhkan lebih banyak waktu untuk diubah menjadi latar objek. Oleh karena itu, disarankan untuk memilih kamera statis ketika menginstal sistem. Selanjutnya, jika awalnya latar objek dimodelkan dengan beberapa objek bergerak maka objek tersebut akan tetap tinggal lebih lama sebelum diubah menjadi latar objek. Oleh karena itu, disarankan untuk memodelkan latar objek tanpa ada objek yang bergerak.

Alternatif lainnya, α dapat ditemukan dengan menggunakan waktu hidup τ yang dapat dihitung dengan persamaan (3.106):

$$\alpha = \frac{1}{\tau \times f} \quad (3.106)$$

dengan f adalah frame per detik. Sebagai contoh, jika diperlukan untuk menahan waktu hidup piksel tertentu lebih dari 30 detik, diberikan $f = 30$ fps, substitusikan ke dalam persamaan (3.106) dengan $\alpha = 0,0011$. Nilai ini akan sama hasilnya ketika dihitung dengan menggunakan persamaan (3.105).

3.11 Operasi Morfologi

Operasi Morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Inti operasi morfologi melibatkan dua larik piksel. *Array* (larik) pertama berupa citra yang akan dikenali operasi morfologi, sedangkan *array* (larik) kedua dinamakan kernel atau *structuring element* (elemen penstruktur). Dua operasi yang mendasari morfologi adalah dilasi dan erosi. Dua operasi lain yang sangat berguna dalam pemrosesan citra adalah *opening* dan *closing*, yang dibentuk melalui dua operasi dasar itu.

3.11.1 Operasi Dilasi

Operasi dilasi biasa dipakai untuk mendapatkan efek pelebaran terhadap piksel bernilai 1 (Gonzalez dkk., 2008). Operasi ini dihitung dengan persamaan (3.107):

$$A \oplus B = \{z | [(\hat{B})_z \cap A] \neq \emptyset\} \quad (3.107)$$

dengan,

- a. $\hat{B} = \{w | w = -b, \text{ untuk } b \in B\}$ dengan \hat{B} adalah refleksi dari himpunan B. Jika B adalah himpunan piksel-piksel yang merepresentasikan sebuah objek di dalam sebuah citra maka \hat{B} adalah himpunan titik-titik di dalam B yang koordinat (x, y) nya telah diubah menjadi $(-x, -y)$.
- b. $(B)_z = \{w | w = b + z, \text{ untuk } b \in B\}$ dengan $(B)_z$ adalah translasi dari himpunan B. Jika B adalah himpunan piksel-piksel yang merepresentasikan sebuah objek di dalam sebuah citra maka $(B)_z$ adalah himpunan titik-titik di dalam B yang koordinat (x, y) nya telah diubah menjadi $(x+z_1, y+z_2)$.
- c. $z = (z_1, z_2)$ dengan z adalah himpunan bilangan integer dan z_1, z_2 adalah anggota bilangan integer dari z.

atau dengan persamaan lain menurut (Gonzalez dkk., 2008), dilasi dapat dihitung dengan persamaan (3.108):

$$A \oplus B = \{z | [(B)_z \cap A] \subseteq A\} \quad (3.108)$$

dengan B adalah sebuah struktur elemen dan A adalah citra objek yang akan dilakukan dilasi.

3.11.2 Operasi Erosi

Operasi Erosi mempunyai efek memperkecil struktur objek. Operasi ini dirumuskan menurut Gonzalez dkk., (2008) operasi erosi dapat dihitung dengan persamaan (3.109):

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (3.109)$$

atau dalam persamaan lain menurut Gonzalez dkk., (2008), operasi erosi dapat dihitung dengan persamaan (3.110):

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\} \quad (3.110)$$

dengan A^c adalah komplemen dari A dan \emptyset adalah himpunan kosong.

3.11.3 Operasi Opening

Operasi *opening* adalah operasi erosi yang diikuti dengan dilasi dengan menggunakan elemen penstruktur yang sama. Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan segmen yang terlalu kecil untuk ditempati oleh elemen penstruktur. Dengan perkataan lain, semua struktur objek yang berukuran lebih kecil daripada elemen penstruktur akan tereliminasi oleh erosi dan kemudian penghalusan dilakukan melalui dilasi (Gonzalez dkk., 2008). Operasi *opening* dapat dihitung dengan persamaan (3.111):

$$A \circ B = (A \ominus B) \oplus B \quad (3.111)$$

3.11.4 Operasi *Closing*

Operasi *closing* berguna untuk menghaluskan kontur dan menghilangkan lubang-lubang kecil (Gonzalez dkk., 2008). Operasi *closing* dihitung dengan persamaan (3.112):

$$A \bullet B = (A \oplus B) \ominus B \quad (3.112)$$

Jadi, operasi *closing* dilaksanakan dengan melakukan operasi dilasi terlebih dahulu dan kemudian diikuti dengan operasi erosi.

3.12 *K-Nearest Neighbor* (KNN)

Algoritme *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut (Matos dan Souza, 2013; Anderson dkk., 2012; Obula dkk., 2014; Wang dkk., 2015). KNN termasuk algoritme *supervised learning* dimana *query instance* yang baru, diklasifikasikan berdasarkan mayoritas dari kategori pada KNN.

Tujuan algoritme ini menghasilkan objek baru berdasarkan atribut dan *training sample*. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah k objek atau (titik training) yang paling dekat dengan titik *query*. Klasifikasi menggunakan *vote* terbanyak diantara klas dari k objek. Algoritme *K-Nearest Neighbor* (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai

prediksi dari *query instance* yang baru.

Algoritme metode *K-Nearest Neighbor* (KNN) sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample*. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan ciri dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klas *training sample*. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klas yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan *Euclidean distance*. Jarak *Euclidean* paling sering digunakan untuk menghitung jarak. Jarak *Euclidean* berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua objek yang direpresentasikan dalam persamaan (3.113 - 3.116) :

$$D_{a,b} = k = d(X_k - Y_k)^2 \quad (3.113)$$

$$X^1 = [X_1, X_2, \dots, X_p] \text{ dan } Y^1 = [Y_1, Y_2, \dots, Y_p] \quad (3.114)$$

$$D_{(x,y)}^2 = (X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \dots + (X_p - Y_p)^2 \quad (3.115)$$

$$D_{(x,y)}^2 = (X - Y)(X - Y) = (Y - X)(Y - X) \quad (3.116)$$

Matrik $D(a,b)$ adalah jarak skalar dari kedua vektor a dan b dari matrik dengan ukuran d dimensi. Semakin besar nilai D akan semakin jauh tingkat keserupaan antara kedua individu dan sebaliknya jika nilai D semakin kecil maka akan semakin dekat tingkat keserupaan antar individu tersebut. Nilai k yang terbaik untuk algoritme ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klas menjadi semakin kabur. Nilai K yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus, dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, $K=1$) disebut algoritme *Nearest Neighbor*.

Ketepatan algoritme KNN sangat dipengaruhi oleh ada atau tidaknya ciri-ciri yang tidak relevan atau jika bobot ciri tersebut tidak setara relevansinya terhadap klas. Riset terhadap algoritme ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap ciri agar performa klasifikasi menjadi lebih

baik. Dengan menggunakan kategori *nearest neighbor* yang paling mayoritas maka dapat diprediksi nilai *query instance* yang telah dihitung.

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor (KNN)*:

- a. Menentukan parameter K (jumlah tetangga paling dekat).
- b. Menghitung kuadrat jarak Euclid (*query instance*) masing-masing objek terhadap data sampel yang diberikan.
- c. Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak Euclid terkecil.
- d. Mengumpulkan kategori Y (klasifikasi *nearest neighbor*).
- e. Menggunakan *nearest neighbor* yang paling mayoritas untuk memprediksi nilai *query instance* yang telah dihitung.

3.13 Kalman Filter

Pertama kali, Kalman Filter diperkenalkan oleh Greg Welch dan Gary Bishop (Bishop dan Welch, 2001; Welch dan Bishop, 2006). Kalman Filter (Srilekha dkk., 2015) memberikan perkiraan yang optimal dari *state* proses dengan meminimalkan varian dari kesalahan estimasi dan memberi pengukuran agar luarannya sesuai.

Metode *Kalman Filter* telah banyak digunakan oleh para peneliti. Ada peneliti yang menggunakan untuk sistem navigasi yang terintegrasi (Gao dkk., 2010). Ada peneliti yang menggunakan untuk mengestimasi posisi kendaraan di *channel* yang *noise*-nya ekstrim (Sharma, 2014). Ada pula peneliti yang menggunakan untuk mengestimasi suhu ruangan (Hasfjord, 2014). *Kalman Filter* juga digunakan untuk memprediksi lokasi kendaraan di frame video berikutnya (Teoh dan Bräunl, 2012). Selain itu, *Kalman Filter* digunakan untuk pelacakan kendaraan yang bergerak secara *real time* (Kocadag dan Demirkol, 2015). Metode ini juga digunakan untuk mendeteksi dan melacak kendaraan (Srilekha dkk., 2015). Metode ini juga digunakan untuk mengestimasi parameter kendaraan (Burkul dkk., 2014). Keuntungan menggunakan *Kalman Filter* adalah (Teoh dan Bräunl, 2012):

- a. Memberikan perkiraan lokasi yang terbaik untuk mencari kendaraan dalam frame video berikutnya sehingga tingkat deteksi ulang semakin baik.
- b. Mengurangi area pencarian untuk mendeteksi ulang kendaraan dan untuk mempersingkat waktu pengolahan.
- c. Mengurangi jumlah deteksi palsu karena daerah gambar yang tidak mengandung kendaraan telah dikeluarkan dari pencarian.

Dalam Lampiran C ditunjukkan kesesuaian nilai *centroid* dalam *Kalman Filter*. Lampiran C berisi contoh perhitungan metode *Kalman Filter*. Proses rekursif dari metode *Kalman Filter* diperlihatkan pada Gambar 3.12. Langkah-langkah pendekatan proses rekursif *Kalman Filter*.

1. Inisialisasi *state*

Pada langkah ini, sistem menginisialisasi \hat{x}_0 dan P_0 . \hat{x}_0 dan P_0 ditetapkan pada awal proses rekursif *Kalman Filter*. \hat{x}_0 dan P_0 hanya digunakan satu kali saat awal proses rekursif *Kalman Filter*. \hat{x}_0 adalah posisi objek yang terlacak saat inisialisasi di frame pertama. \hat{x}_0 menyatakan *state* matrik awal yang memiliki ordo 4×1 . Nilai \hat{x}_0 berasal dari ciri *centroid*. \hat{x}_0 dapat ditunjukkan pada persamaan (3.117).

$$\hat{x}_0 = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \quad (3.117)$$

dengan:

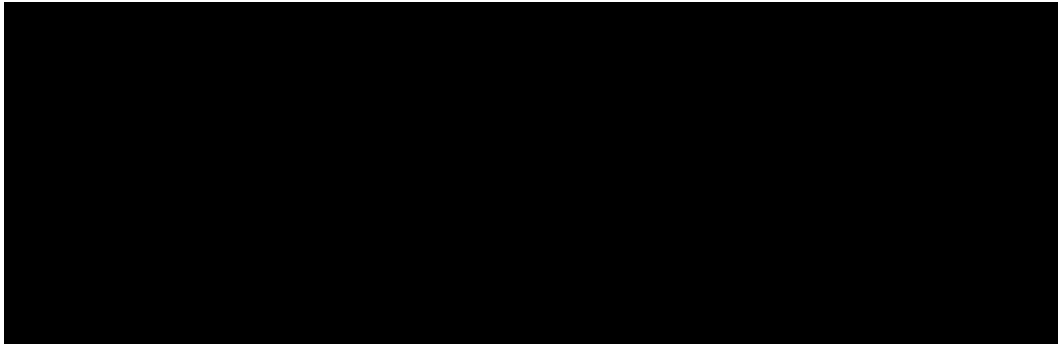
x = nilai posisi di arah x

\dot{x} = nilai *velocity* di arah x

y = nilai posisi di arah y

\dot{y} = nilai *velocity* di arah y

P_0 menyatakan *process covariance matrix* (matrik kovarian proses) yang memiliki ukuran 4×4 . P_0 diberikan secara acak. P_0 diisi dengan standar deviasi. P_0 memberikan estimasi kovarian kesalahan *state* yang mungkin terjadi.



Gambar 3.12 Proses rekursif *Kalman Filter* (Welch dan Bishop, 2006)

2. State sebelumnya

State sebelumnya adalah \hat{x}_{k-1} dan P_{k-1} .

Untuk awal proses rekursif *Kalman Filter*, nilai \hat{x}_0 dan P_0 diubah menjadi nilai state sebelumnya seperti yang ditunjukkan pada persamaan (3.118 dan (3.119).

Nilai awal adalah nilai centroid.

$$\hat{x}_0 \rightarrow \hat{x}_{k-1} \quad (3.118)$$

dengan:

$\hat{x}_0 =$ state matrik awal

$\hat{x}_{k-1} =$ state matrik sebelumnya

$$P_0 \rightarrow P_{k-1} \quad (3.119)$$

dengan:

$P_0 =$ matrik kovarian proses awal

$P_{k-1} =$ matrik kovarian proses sebelumnya

Untuk proses rekursif berikutnya pada *Kalman Filter*, nilai \hat{x}_{k-1} dan P_{k-1} berasal dari nilai \hat{x}_k dan P_k pada frame sebelumnya seperti yang ditunjukkan pada persamaan (3.133) dan (3.134) dengan $\hat{x}_k =$ state matrik terkini, $P_k =$ matrik kovarian proses terkini.

3. State Baru (Prediksi)

Untuk state baru (prediksi) digunakan persamaan (3.120):

$$\hat{x}_k^- = A\hat{x}_{k-1} + B_{u_k} \quad (3.120)$$

dengan:

\hat{x}_k^- = state matrik prediksi

A = state transition model

B = control model

u = matriks variabel kontrol (*control variable matrix*).

State transition model (A) yaitu model yang mendeskripsikan perubahan state yang terjadi diantara perubahan waktu yang ada. Nilai matrik A untuk matrik dua dimensi dibentuk seperti pada persamaan (3.121). *State transition model* (A) disesuaikan formatnya agar dapat dilakukan perkalian antara matrik A dan *state* matrik (Han dkk., 2008; Chen dan Park, 2013) sehingga diperoleh format yang benar untuk nilai *state* matrik berikutnya.

$$A = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.121)$$

ΔT adalah perpindahan state ke posisi baru dalam selang waktu yang sangat kecil yang dibutuhkan dalam satu siklus rekursif perputaran *Kalman Filter* (*time for 1 cycle*). Selang waktu antara satu frame ke frame yang lain sangatlah kecil. Oleh karena itu, nilai ΔT diasumsikan bernilai 1.

Control model (B) yaitu model yang mendeskripsikan kontrol masukan saat terjadi perubahan nilai *state*. Nilai B mengacu dari persamaan (3.122) yang merupakan gerak lurus berubah beraturan (Wu dkk., 2010).

$$X = X_0 + \dot{x}t + \frac{1}{2}\ddot{x}t^2 \quad (3.122)$$

dengan:

X = *state*

X_0 = *initial state*

t = *time lapse* = selang waktu

\dot{x} = *velocity* di arah x

\ddot{x} = *acceleration* = percepatan

Matriks B dalam dua dimensi untuk objek bergerak dapat dihitung dengan persamaan (3.123).

$$B = \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 \\ \Delta T & \frac{1}{2}\Delta T^2 \\ 0 & \Delta T \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}1^2 & 0 \\ 1 & \frac{1}{2}1^2 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (3.123)$$

Objek yang bergerak atau kendaraan bergerak, *control variable matrix* (u) diasumsikan memiliki percepatan dalam gerakannya. *Control variable matrix* (u) memiliki ordo 2×1 dihitung dengan persamaan (3.124).

$$u = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \quad (3.124)$$

dengan:

a_x = percepatan x

a_y = percepatan y

Percepatan pada kendaraan yang bergerak dari satu frame ke frame berikutnya sangat kecil. Karena nilainya yang sangat kecil maka pada penelitian ini diasumsikan nilai $a_x = 0$ dan $a_y = 0$.

Pada metode *Kalman Filter* diprediksi juga perhitungan matrik kovarian proses yang mengestimasi berapa besar kovarian kesalahan yang mungkin terjadi saat perubahan *state*. Matriks kovarian proses (P_k^-) dihitung dengan persamaan (3.125).

$$P_k^- = AP_{k-1}A^T + Q \quad (3.125)$$

dengan:

P_k^- = matrik kovarian proses prediksi

P_{k-1} = matrik kovarian proses sebelumnya

A = *state transition model*

A^T = matrik *state transition model* transpose

$$A^T = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Q = *process noise covariance matrix* (matrik kovarian *noise*)

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

Nilai Q memiliki ukuran 4x4 untuk kendaraan bergerak (Han dkk., 2008; Chen dan Park, 2013). Nilai Q diberikan secara random.

4. Perbarui dengan Pengukuran Baru dan *Kalman Gain*

Untuk memastikan kebenaran nilai prediksi matrik *state* baru dan nilai matrik kovarian proses maka keduanya perlu diperbarui dengan sebuah pengukuran baru dan *Kalman Gain*. *Kalman Gain* (K_k) dapat dihitung dengan persamaan (3.126).

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (3.126)$$

dengan:

K_k = *Kalman Gain*

H = *measurement model*

R = *process noise covariance matrix*

Measurement model (H) dibentuk dalam ordo 2 x 4 karena nilai ini bentuk format matrik identitasnya disesuaikan agar dapat dilakukan perkalian antara matriks H dan matrik lain (Han dkk., 2008; Chen dan Park, 2013). Sehingga, hasil dalam format yang benar untuk *state* matrik di waktu berikutnya dapat diperoleh.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

H^T = *measurement model* (model pengukuran) transpose

$$H^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Process noise covariance matrix (R) menspesifikasikan kovarian proses dengan ukuran 2 x 2. Nilai 10 pada nilai R karena karena nilai ini bentuk format matrik identitasnya disesuaikan agar dapat dilakukan penjumlahan antara matrik R dan matrik lain

$$R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Matriks pada persamaan (3.126) dapat dihitung dengan menggunakan pembagian matriks. Pembagian matriks dihitung dengan cara melakukan

perkalian matrik A dengan invers matrik B seperti ditunjukkan persamaan (3.127).

$$\frac{A}{B} = AB^{-1} \quad (3.127)$$

Invers Matrik dapat dihitung dengan persamaan (3.128):

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (3.128)$$

Posisi *state* matrik juga perlu diperbarui. Persamaan pembaruan *state* matrix ditunjukkan pada persamaan (3.129).

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.129)$$

5. Pengukuran

Measurement (Pengukuran) digunakan untuk memperbaharui nilai *state* matrik prediksi. Pembaruan persamaan (3.129) dilakukan menggunakan *measurement* (pengukuran) yang dapat dihitung dengan persamaan (3.130).

$$z_k = H\hat{x}_k + v_k \quad (3.130)$$

dengan:

\hat{x}_k = *state* matrik terkini

z_k = *measurement of the state*

v_k = *measurement noise (uncertainty)*.

v_k dipengaruhi oleh distribusi *Gaussian* (Chen dan Park, 2013). Pada penelitian ini v_k diasumsikan bernilai nol sehingga nilai pengukurannya menjadi persamaan (3.131).

$$z_k = H\hat{x}_k \quad (3.131)$$

Jika nilai z_k pada persamaan (3.131) dimasukkan ke persamaan (3.129) maka hasilnya menjadi:

$$\hat{x}_k = \hat{x}_k^- + K_k(H\hat{x}_k - H\hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^- + K_k H\hat{x}_k - K_k H\hat{x}_k^-$$

$$\hat{x}_k - K_k H\hat{x}_k = \hat{x}_k^- - K_k H\hat{x}_k^-$$

$$(1 - K_k H)\hat{x}_k = (1 - K_k H)\hat{x}_k^-$$

$$\hat{x}_k = \hat{x}_k^-$$

Dengan demikian diperoleh nilai \hat{x}_k untuk *state* matrik terkini.

6. Terkini menjadi sebelumnya

Persamaan matrik kovarian proses terkini (P_k) dapat dihitung dengan persamaan (3.132).

$$P_k = (I - K_k H) P_k^- \quad (3.132)$$

dengan:

P_k = matrik kovarian proses terkini

I = matrik identitas dengan ordo 4×4

Jika nilai *state* matrik terkini dan matrik kovarian proses terkini telah diperoleh maka nilai k diubah menjadi $k-1$. Artinya nilai \hat{x}_k diberikan ke \hat{x}_{k-1} dan nilai P_k diberikan ke P_{k-1} yang dapat dihitung dengan persamaan (3.133) dan (3.134).

$$\hat{x}_k \rightarrow \hat{x}_{k-1} \quad (3.133)$$

$$P_k \rightarrow P_{k-1} \quad (3.134)$$

7. Keluaran *state* terkini

Keluaran *state* terkini adalah *state* matrik terkini (\hat{x}_k) dan matrik kovarian proses terkini (P_k).

BAB 4

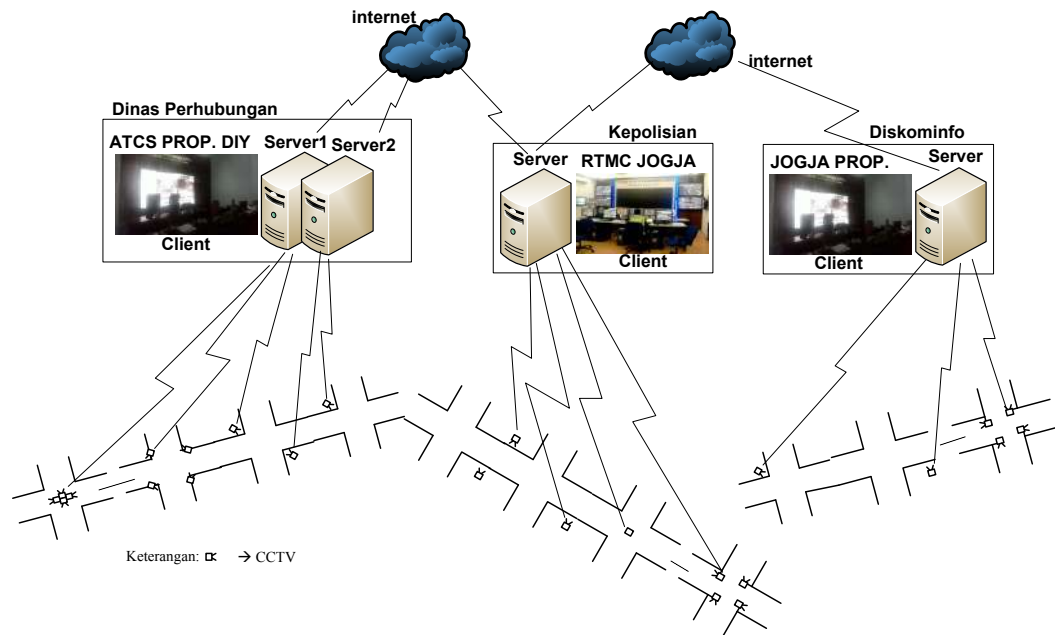
METODE PENELITIAN

4.1 Gambaran Umum Sistem Pemantauan dan Pengaturan Lalu Lintas

Sistem Pemantauan dan Pengaturan Lalu Lintas diperlukan oleh pihak Kepolisian agar permasalahan lalu lintas dapat segera ditangani dengan baik. Gambar 4.1 menunjukkan Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang berjalan saat ini. Di ruang *Regional Traffic Management Center* (RTMC), pihak Kepolisian dapat memonitor lalu lintas dari berbagai CCTV milik Dinas Perhubungan (Dishub), Dinas Komunikasi dan Informatika (Diskominfo) dan Dinas Lalu Lintas (Dislantas) karena datanya terintegrasi melalui internet. Di ruang RTMC Dislantas, *Area Traffic Control System* (ATCS) Dishub dan JOGJA PROP Diskominfo terdapat banyak hasil rekaman video yang di lihat bersamaan. Di ruang-ruang tersebut, terdapat monitor besar beberapa Server dan beberapa *Personal Computer* (PC) / *Client*. CCTV secara *real time* mengirim data video ke Server.

Kepolisian Yogyakarta masih memonitor lalu lintas secara manual sejak tahun 2007 (Munawar, 2007, 2008b, 2008a) sampai saat ini. Monitor secara manual artinya operator secara langsung memantau lalu lintas dari semua *Closed Circuit Television* (CCTV) yang ada di Yogyakarta. Bila pihak Kepolisian membutuhkan data video dari CCTV milik Dishub dan Diskominfo Yogyakarta untuk bukti pelanggaran lalu lintas, bukti peradilan atau bukti kecelakaan maka mereka harus memintanya ke Dishub dan Diskominfo sesuai prosedur kebijakan pimpinan Dishub dan Diskominfo karena kewenangan data video milik masing-masing Dinas yang bersangkutan dan data bersifat rahasia. Dishub dan Diskominfo akan mengecek ketersediaan data videonya karena masa simpan video hanya 30 hari saja.

Pada tahun 2012 Dinas Perhubungan menyatakan bahwa masalah yang dihadapi adalah data video sering diterima tidak lengkap/*discontinue* karena banyak frame video yang hilang. Akibatnya data video tidak dapat di olah lebih lanjut.



Gambar 4.1 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang berjalan saat ini

Kegagalan menghadirkan frame-frame penting ini menghambat proses analisis video. Kegagalan ini juga menghambat proses mendapatkan informasi lalu lintas mengenai kondisi lalu lintas terkini, penyebab kecelakaan dan permasalahan lalu lintas yang sering terjadi bersamaan di beberapa tempat yang berbeda. Bukti otentik berupa data video tidak dapat diperoleh secara maksimal untuk proses di pengadilan apabila Kepolisian dan Dinas yang terkait tidak dapat memanfaatkan data video secara maksimal karena data video tidak dapat diolah lebih lanjut. Data video diperlukan untuk mendapatkan informasi yang akurat mengenai gambaran yang menyeluruh di tempat kejadian perkara. Misalnya, kecepatan kendaraan sesaat sebelum dan sesudah terjadi kecelakaan. Proses ini merupakan proses audit yang memerlukan pelacakan kendaraan.

Penyelesaian masalah yang diusulkan adalah dengan memodifikasi konsep Arsitektur *Smart Video Sensor* (SVS) dari penelitian yang telah dilakukan oleh Sánchez dkk., (2012) dan Ripolles dkk., (2012). SVS yang diusulkan adalah sensor video yang mengekstrak ciri - ciri objek lalu lintas dan memampatkan data video. SVS mengekstrak ciri-ciri objek dalam *memory* dan membentuk ciri-ciri

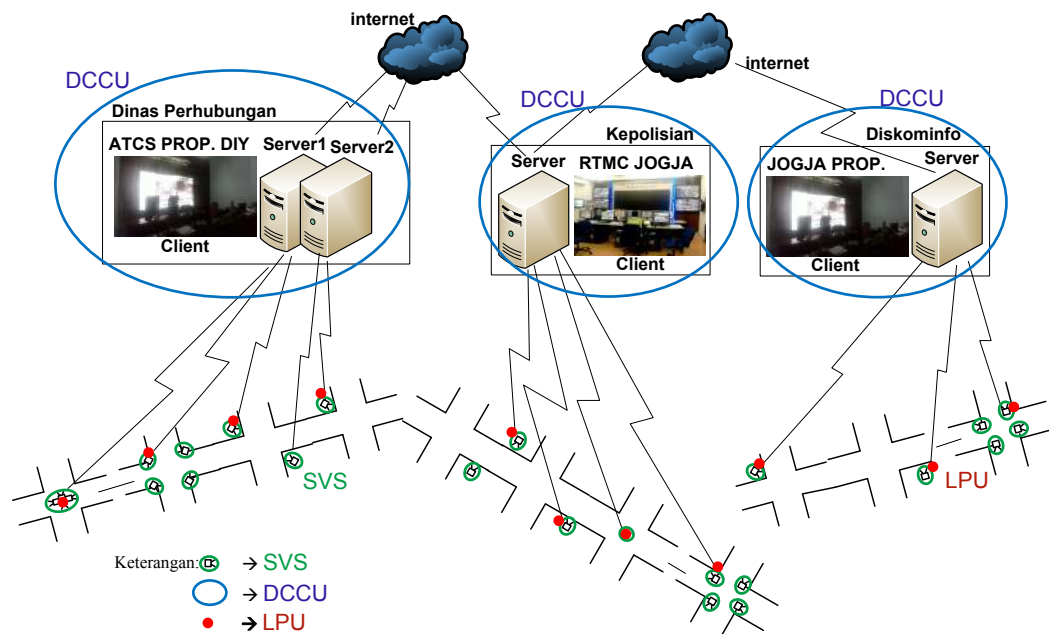
objek dalam format XML. Gambar 4.2 menunjukkan ilustrasi kemanfaatan SVS pada pelacakan kendaraan yang terklasifikasi. Ilustrasi ini menjelaskan cara sederhana ciri dalam format XML dikirim dari SVS ke DCCU dan LPU.



Gambar 4.2 Ilustrasi kemanfaatan SVS pada pelacakan kendaraan yang terklasifikasi

SVS diletakkan di setiap perempatan wilayah Yogyakarta sesuai dengan jumlah kamera CCTV yang ada di wilayah Yogyakarta. Karena sesuai dengan tujuan awal yaitu untuk mengatasi permasalahan jumlah frame video yang hilang di setiap daerah yang memiliki CCTV. SVS tidak menyimpan ciri objek di dalam file. SVS langsung mengirimkan data video yang sudah dimampatkan dan ciri-ciri objek lalu lintas dalam format XML ke *Local Processing Unit* (LPU) dan *Data Center and Control Unit* (DCCU). Gambar 4.3 menunjukkan Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang diusulkan. Hasil SVS dikirim ke LPU untuk pemrosesan secara lokal. Sedangkan hasil SVS dikirim ke DCCU agar dapat digunakan untuk analisa data lalu lintas selanjutnya. LPU merupakan unit pemrosesan pengaturan lalu lintas secara lokal. LPU tidak menyimpan hasil SVS. LPU melakukan pengolahan ciri dan hasilnya langsung digunakan untuk melakukan pemantauan dan pengaturan lampu lalu lintas. LPU diletakkan di tempat yang membutuhkan pengontrolan lalu lintas setempat. LPU diletakkan di setiap perempatan Yogyakarta. Setiap perempatan wilayah Yogyakarta cukup sebuah LPU karena LPU hanya memproses pengaturan lalu lintas secara lokal. DCCU menyimpan hasil SVS di dalam struktur file. DCCU berisi data dan aplikasi-aplikasi yang disimpan di dalam *server* yang ada di suatu organisasi. Aplikasi-aplikasi yang dibuat di DCCU dibutuhkan oleh pihak Kepolisian, Dishub dan Diskominfo Jogja. DCCU dapat melakukan pengontrolan lalu lintas dari jarak

jauh. DCCU dapat melakukan pelacakan, klasifikasi, dan pengaturan lalu lintas dari *Area Traffic Control System* (ATCS) Dinas Perhubungan Yogyakarta, Jogja Prop. Diskominfo dan RTMC Jogja. DCCU dapat berupa *dashboard* yang memonitor semua kegiatan yang ada di masing-masing ATCS JOGJA, RTMC JOGJA dan JOGJA PROP. DCCU diletakkan di setiap Dinas terkait. Dalam penelitian ini, DCCU diletakkan di ATCS JOGJA, RTMC JOGJA dan JOGJA PROP. Karena setiap Dinas terkait membutuhkan pengontrolan lalu lintas dari jarak jauh. Bila *Green Wave* di *setting* di DCCU maka DCCU dapat mengatur agar kendaraan yang melalui jalan besar dapat mendapat prioritas lampu hijau secara berurutan.



Gambar 4.3 Sistem Pemantauan dan Pengaturan Lalu Lintas Yogyakarta yang diusulkan

Dari aplikasi-aplikasi di DCCU dapat diperoleh *flow* (arus), *speed* (kecepatan) dan *density* (kerapatan) yang merupakan *traffic parameter*. Karakteristik mendasar arus lalu lintas adalah *flow* (arus), *speed* (kecepatan) dan *density* (kerapatan) (May, 1990; Mathew dan Rao, 2006). Persamaan-persamaan yang berkaitan dengan karakteristik arus lalu lintas dapat dilihat pada persamaan (3.83-3.92). Klein dan Kelley (1996) dan Board (2000) menyatakan *traffic*

parameter atau parameter lalu lintas adalah suatu tolok ukur yang dipakai sebagai patokan dalam sistem transportasi lalu lintas. *Traffic parameter* merupakan karakteristik arus lalu lintas yang dapat diklasifikasikan berdasarkan *quantify measures*, *quality assessment measures*, *movement measures* dan *composition/classification measures*.

Traffic parameter diperlukan agar dapat membantu menyelesaikan permasalahan lalu lintas. Bila *traffic parameter* diketahui maka kepadatan kendaraan juga dapat diketahui berdasarkan *flow*/arus lalu lintas dan *density*/kepadatan lalu lintas kendaraan. Kemungkinan terjadinya kecelakaan dapat diketahui lebih dini melalui *track* atau jalur yang dilalui oleh suatu kendaraan dan kecepatan kendaraan yang melebihi standar kecepatan maksimal di suatu tempat. Indikasi kemungkinan terjadi suatu kecelakaan dapat dilihat dari kendaraan melaju dengan tidak terkendali atau berpindah dari satu jalur ke jalur yang lain dengan tidak beraturan dan kecepatan kendaraan itu melebihi batas kecepatan yang telah ditentukan. Bila LPU diletakkan di tiap persimpangan secara berurutan maka kendaraan dapat dilacak keberadaannya dengan memperhitungkan kecepatannya. Keberadaan kendaraan diketahui dengan cara mengidentifikasi ciri kendaraan. Ciri kendaraan yang terdeteksi di suatu LPU maka ciri itu akan dibandingkan dengan ciri kendaraan yang terdeteksi di LPU berikutnya. Bila ciri kendaraan itu sama maka kendaraan berhasil diidentifikasi dan berhasil terlacak. Ciri kendaraan juga dapat digunakan untuk melacak siapa yang melakukan pelanggaran lalu lintas. Ciri kendaraan juga dapat digunakan untuk memantau kondisi real lalu lintas jalan raya yang terkini.

4.2 Arsitektur *Smart Video Sensor* (SVS)

Smart Video Sensor (SVS) yaitu sebuah sensor video yang dapat mengekstrak ciri dan menghasilkan ciri objek lalu lintas. Ciri yang diperoleh dari SVS dikirim ke DCCU dan LPU. Dengan adanya SVS maka ciri objek lalu lintas dapat langsung digunakan oleh DCCU atau LPU sesuai kebutuhan. Tahapan pengolahan data seperti pra pemrosesan, segmentasi dan ekstraksi ciri sudah dikerjakan oleh *SVS* sehingga hasil yang dibutuhkan dapat diperoleh dengan cepat.

Pihak Kepolisian dan Dinas Perhubungan yang bertugas dapat segera melihat data video asli agar dapat melakukan tindakan yang tepat karena data video aslinya juga ikut dikirimkan dalam bentuk yang sudah dimampatkan. Pihak Kepolisian dan Dinas Perhubungan yang bertugas juga dapat melihat kesesuaian data asli dan ciri objek lalu lintas karena sudah dilakukan sinkronisasi menggunakan *timestamp*. Bila jumlah *Smart Video Sensor* banyak maka *time stamp datetime* menjadi solusi untuk penyeragaman waktu.

Gambar 4.4 menunjukkan Arsitektur SVS yang diusulkan. Arsitektur yang baik adalah arsitektur yang dapat menunjukkan komponen dan fungsi, interkoneksi dan semantik. Komponen arsitektur SVS adalah Modul Sensor Video, Modul Pemampatan Video, Modul Ekstraksi Ciri, Modul Representasi Ciri dan Modul Transmisi Data. Fungsi SVS yang dikembangkan adalah mengekstrak ciri-ciri objek lalu lintas dan memberikan hasil pengolahannya untuk aplikasi di DCCU atau LPU. Data masukan SVS adalah *scene*. Data keluaran SVS adalah ciri-ciri objek lalu lintas dan data video yang dimampatkan. Modul Sensor Video berfungsi mengakuisisi *scene* menjadi sekumpulan frame yang berurutan. Modul Pemampatan Video dan Modul Ekstraksi Ciri menerima frame-frame. Modul Pemampatan Video berfungsi membentuk kumpulan frame menjadi data video dan memampatkan data video untuk meminimalkan ukuran video. Modul Ekstraksi Ciri berfungsi mengekstrak ciri-ciri tiap frame. Modul Representasi Ciri berfungsi merepresentasikan ciri-ciri objek yang dibentuk dalam format XML. Modul Transmisi Data berfungsi mentransmisikan data video yang dimampatkan dan ciri-ciri objek dalam format XML ke DCCU atau LPU. Fokus penelitian ini pada Modul Ekstraksi Ciri dan Modul Representasi Ciri yang ada di SVS.



Gambar 4.4 Arsitektur *Smart Video Sensor* yang diusulkan

Gambar 4.5 menunjukkan tahapan pada *Smart Video Sensor (SVS)*. Pertama, *scene* di akuisisi dalam sensor video menggunakan metode *sampling* dan kuantisasi kemudian diterima dalam bentuk sinyal. Alasan menggunakan sensor video karena pihak kepolisian dan Dinas Perhubungan memerlukan bukti otentik berupa data video sebagai bukti di pengadilan bila terjadi permasalahan lalu lintas dan untuk memantau lalu lintas terkini. Sinyal ini diubah menjadi frame-frame. Frame-frame ini diterima oleh Modul Ekstraksi Ciri dan Modul Pemampatan Video. Kedua, Modul Ekstraksi Ciri mengekstrak ciri tiap objek dengan tujuan ciri-ciri objek dapat dibedakan antara objek yang satu dengan objek yang lain. Metode yang digunakan pada Modul Ekstraksi Ciri adalah normalisasi ukuran dan normalisasi intensitas menggunakan *Histogram Equalization (HE)*, *Gaussian Mixture Model (GMM)*, morfologi *opening* dan *closing*. Alasan pemilihan metode *Histogram Equalization* karena metode ini dapat meningkatkan kualitas citra secara menyeluruh (Hawlder dan Hossain, 2014; Xu dkk., 2014; Diego, 2014; Yadav dkk., 2014; Kadir dan Susanto, 2013). Alasan pemilihan metode GMM karena metode ini lebih adaptif terhadap perubahan latar objek (Gonzalez dan Woods, 2012; Santosh dkk., 2013; Ping dkk., 2015), akurasinya bagus dan biaya komputasinya rendah (Shahbaz dkk., 2016) dan pembaruan latar objek sangat cepat dan real-time, mudah menganggap kendaraan berhenti sebagai latar objek. Alasan pemilihan morfologi *opening* dan *closing* karena *opening* dapat menebalkan bentuk objek dan *closing* dapat menghilangkan lubang kecil dan mengisi gap pada garis bentuk (*contour*) objek (Gonzalez dkk., 2008). Objek mengekstrak ciri menggunakan rumus untuk menghasilkan 39 ciri. Ketiga, Modul Representasi Ciri menyusun dan menyajikan informasi dengan format yang dinamis. Hal ini diperlukan karena ciri-ciri obyek memiliki banyak variasi jumlah dan panjang nilai ciri. Oleh karena itu, metode yang sesuai digunakan pada Modul Representasi Ciri ini adalah XML. Alasan pemilihan XML karena XML tidak memerlukan format baku dalam membuat dokumen *markup*, XML memiliki kemudahan dalam mempertukarkan informasi di platform yang berbeda, dan XML dapat menyimpan informasi tag waktu pada setiap frame yang berfungsi sebagai *time stamp*. *Time stamp* dapat menghubungkan frame terkini dengan

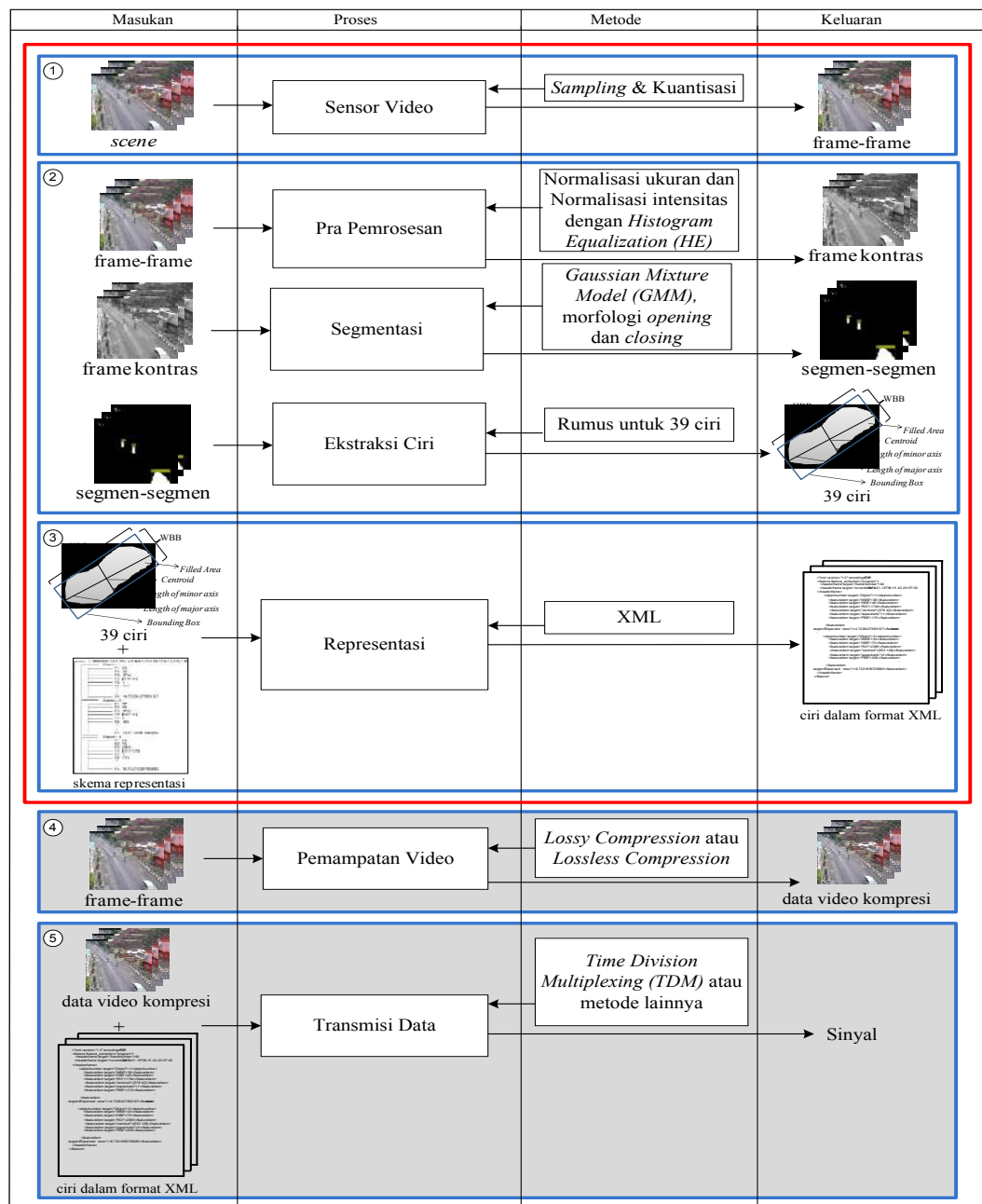
frame berikutnya sehingga mempermudah dalam melacak objek. Masukan Modul Representasi Ciri adalah skema representasi dan ciri-ciri objek. Hasil pada Modul Representasi Ciri adalah ciri-ciri objek dalam bentuk representasi XML. Keempat, Modul Pemampatan Video memampatkan data video / kumpulan frame untuk meminimalkan ukuran *video*. Beberapa alternatif metode yang dapat digunakan pada Modul Pemampatan Video adalah metode *Lossy Compression* dan *Lossless Compression*. Kelima, Modul Transmisi Data mengirimkan data video yang telah dimampatkan dan ciri-ciri objek dalam representasi XML ke DCCU atau LPU. Salah satu metode yang dapat digunakan pada transmisi data yaitu *Time Division Multiplexing (TDM)*, namun tidak menutup kemungkinan untuk menggunakan metode yang lain. Pada penelitian ini hanya blok yang ada di dalam kotak berwarna merah pada Gambar 4.5 yang dikerjakan. Blok yang dikerjakan hanya 3 tahap pertama. Blok yang berwarna abu-abu pada Gambar 4.5 adalah blok yang tidak dikerjakan. Blok yang tidak dikerjakan adalah tahap keempat dan kelima.

4.1.1 Modul Sensor Video

Prinsip kerja sensor ini adalah sensor merekam *scene*. Data masukan dari sensor video adalah *scene*. Keluaran sensor video adalah frame-frame. Alasan menggunakan sensor video karena pihak kepolisian dan Dinas Perhubungan memerlukan bukti otentik berupa data video sebagai bukti di pengadilan bila terjadi permasalahan lalu lintas dan untuk memantau lalu lintas terkini.

Sensor video digunakan untuk menangkap *scene* yang direkam oleh kamera video. Kamera menangkap cahaya yang dipantulkan oleh objek, yang berada di dalam *scene*. Cahaya ini diteruskan oleh sensor. Di dalam sensor, ada *filter array* yang memisahkan cahaya menjadi 3 warna dasar. Warna dasar itu adalah RGB (*Red, Green, Blue*). Dari 3 warna ini, sensor mengubah cahaya menjadi sinyal elektronik. Sinyal elektronik ini dikirimkan ke *processor*. *Processor* mengolah sinyal elektronik menjadi citra dan audio. Kemudian citra dan audio dikirim ke media penyimpanan. Format penyimpan data di dalam media penyimpanan bisa dipilih formatnya mulai dari MPEG, 3GP, AVI, WMV (Nurchaya, 2014). Format penyimpanan untuk kamera CCTV yang bisa

digunakan adalah : 3GP, MP4, AVI, WMV. Pada penelitian ini format data video yang diperoleh dari CCTV adalah .3gp. Alasan pemilihan format 3GP karena ukuran gambar dan *bandwidth* dari 3GP sangat kecil sehingga dapat mengirimkan video dan audio dengan cepat. 3GP adalah format file multimedia yang didefinisikan oleh 3GPP (*3rd Generation Partnership Project*). 3GP dirancang untuk efisiensi tempat penyimpanan dan *bandwidth* (ETSI, 2011).

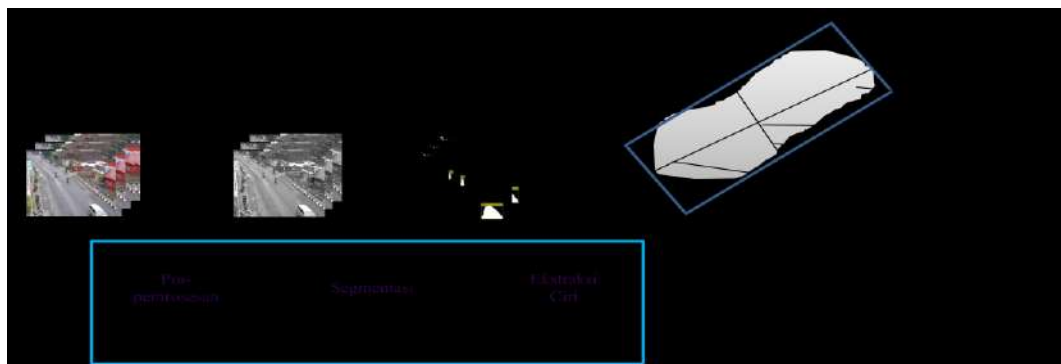


Gambar 4.5 Tahapan pada *Smart Video Sensor*

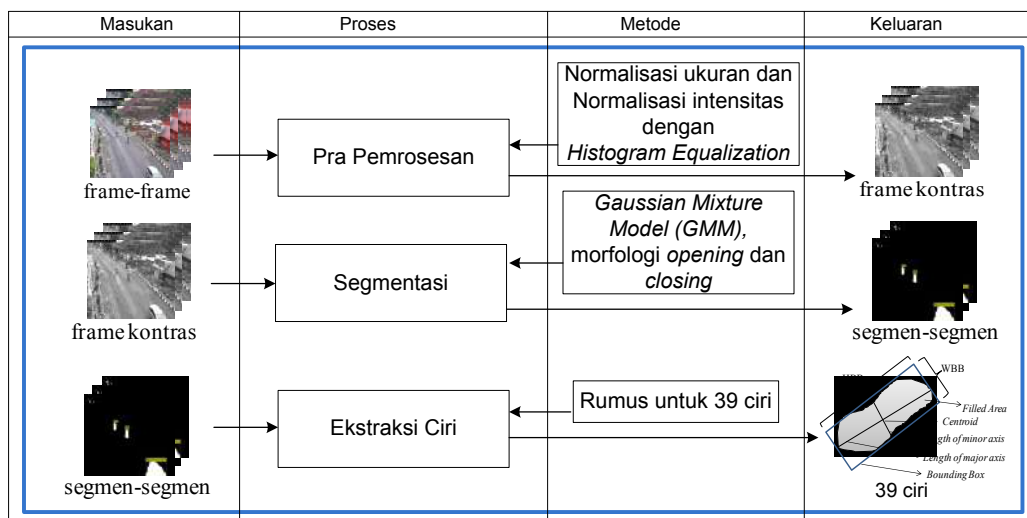
4.1.2 Modul Ekstraksi Ciri

Modul Ekstraksi Ciri berfungsi untuk mengekstrak nilai-nilai unik suatu objek yang membedakan dengan objek yang lain. Alasan diperlukannya Modul Ekstraksi Ciri karena objek yang satu perlu dibedakan dengan objek lainnya. Modul Ekstraksi Ciri ditunjukkan pada Gambar 4.6. Modul Ekstraksi Ciri memiliki proses Pra Pemrosesan, Segmentasi dan Ekstraksi Ciri. Tahapan pada Modul Ekstraksi Ciri dapat ditunjukkan pada Gambar 4.7.

Pada Pra Pemrosesan, frame-frame diproses sehingga menjadi frame-frame yang lebih kontras. Metode yang digunakan adalah *Histogram Equalization (HE)*. Langkah-langkah pembuatan *Histogram Equalization* ini berasal dari persamaan (3.93). Gambar A.2 pada Lampiran A menunjukkan Algoritme *Histogram Equalization*.



Gambar 4.6 Modul Ekstraksi Ciri

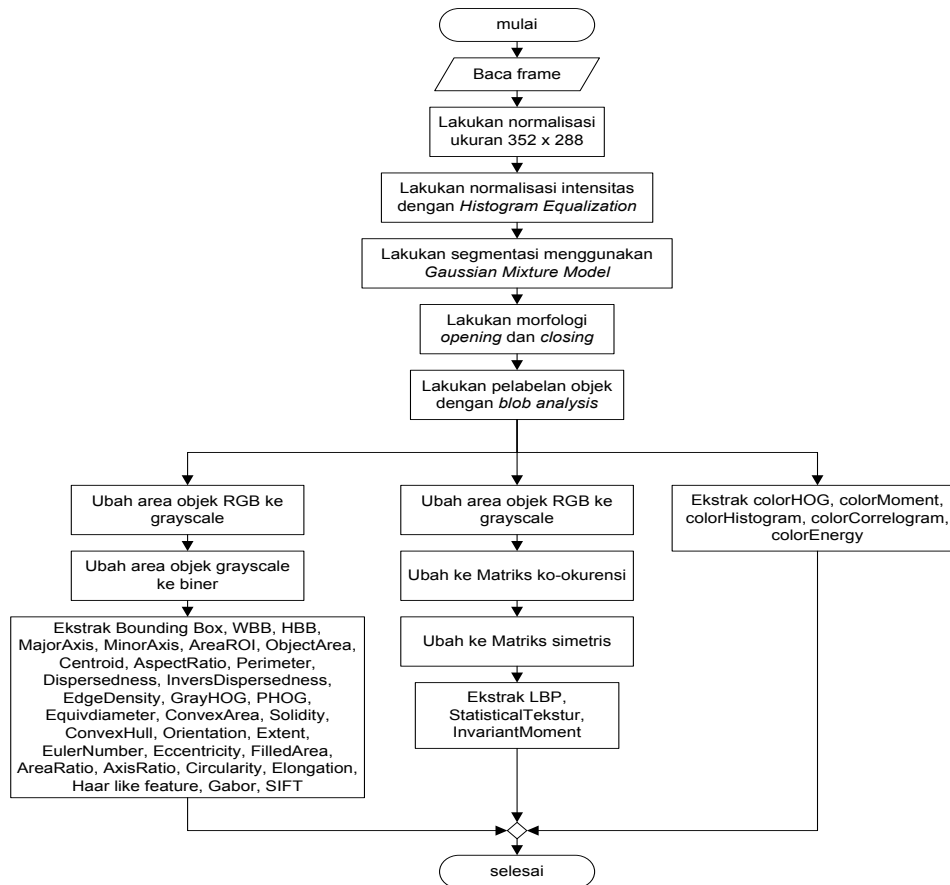


Gambar 4.7 Tahapan pada Modul Ekstraksi Ciri

Pada Segmentasi, frame-frame diproses sehingga menjadi segmen-segmen menggunakan *Gaussian Mixture Model (GMM)*. Langkah-langkah GMM ada pada persamaan (3.94-3.105). Segmentasi juga melakukan post-pemrosesan. Tujuan post-pemrosesan adalah untuk menghilangkan *noise*. Post-pemrosesan yang digunakan adalah morfologi *opening* dan *closing*. Morfologi *opening* menghaluskan kontur objek dan menghilangkan segmen yang terlalu kecil. Operasi *opening* adalah operasi erosi yang diikuti operasi dilasi dengan menggunakan elemen penstruktur yang sama. Rumus *opening* dapat dilihat pada persamaan (3.110). Gambar A.4 pada Lampiran A menunjukkan Algoritme erosi. Rumus erosi ada pada persamaan (3.108-3.109). Gambar A.5 pada Lampiran A menunjukkan Algoritme dilasi. Rumus dilasi ada pada persamaan (3.106-3.107). Morfologi *closing* menghaluskan kontur, menghilangkan lubang kecil dan mengisi gap pada garis-garis bentuk (*contour*). Operasi *closing* adalah operasi dilasi yang diikuti operasi erosi dengan menggunakan elemen penstruktur yang sama. Rumus *closing* ada pada persamaan (3.111).

Model Ekstraksi Ciri menunjukkan gambaran nyata proses ekstraksi ciri sederhana mulai dari proses awal sampai selesai proses ekstraksi ciri. Model Ekstraksi Ciri ini ditunjukkan pada Modul Ekstraksi Ciri pada Gambar 4.6. Gambar A.6 pada Lampiran A menunjukkan langkah-langkah Ekstraksi Ciri. Tiap segmen diberi label. Segmen yang diberi label adalah objek yang terdeteksi. Setelah itu, tiap objek diekstrak cirinya untuk menghasilkan ciri-ciri objek. Gambar 4.8 menunjukkan tahapan untuk mengekstrak ciri. Hasil model ekstraksi ciri adalah ciri-ciri objek lalu lintas. Ada 39 ciri dalam tiap objek. Rumus untuk 39 ciri yang digunakan ada pada persamaan (3.1-3.82). Jumlah ciri dapat bertambah sesuai dengan kebutuhan dari Pihak Kepolisian dan Dinas Perhubungan. Saat ini, ciri diperlukan sebanyak 39 ciri karena ciri-ciri ini akan digunakan untuk berbagai kebutuhan seperti identifikasi objek, deteksi objek, pengenalan objek, klasifikasi objek, pelacakan objek, perhitungan jumlah objek, perhitungan kecepatan objek dan estimasi kemacetan lalu lintas yang diperlukan oleh pihak Kepolisian dan Dinas Perhubungan. Lampiran B menunjukkan 39 ciri yang ada dapat dimanfaatkan untuk berbagai kebutuhan Dinas terkait. Lampiran

ini menunjukkan beberapa manfaat ciri, walaupun tidak menutup kemungkinan untuk dapat dimanfaatkan ke hal yang lain.



Gambar 4.8 Tahapan Ekstraksi Ciri

4.1.3 Modul Representasi Ciri

Modul Representasi Ciri yang ditunjukkan pada Gambar 4.9 berfungsi untuk menyusun dan menyajikan informasi yang formatnya tidak baku. Pada penelitian ini, representasi ciri yang dipilih adalah representasi dalam format XML. XML digunakan dalam membuat dokumen *markup* untuk keperluan pertukaran data antar sistem yang beraneka ragam.

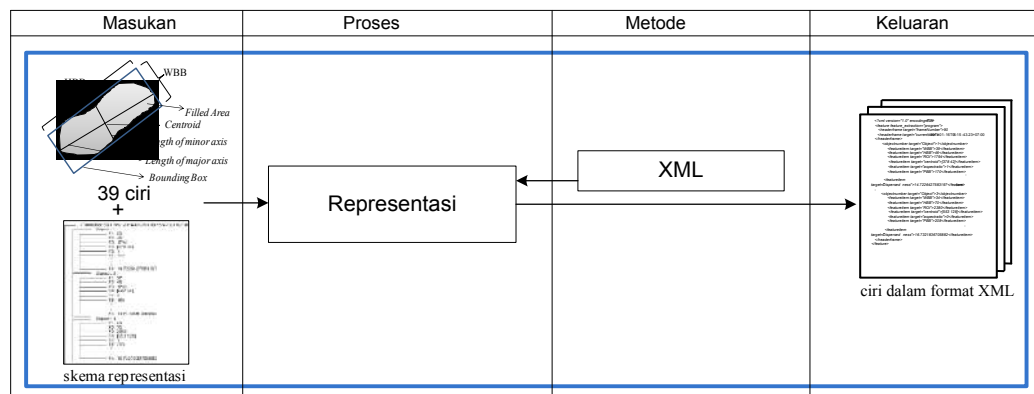
Alasan pemilihan format XML karena XML memiliki kemudahan dalam mempertukarkan informasi di platform yang berbeda (Carey, 2007; Carey, 2010; Parsons dan Oja, 2012). Selain itu, format XML digunakan karena XML dapat menyimpan informasi tag waktu. Informasi tag waktu disimpan untuk

menyelesaikan masalah data video lalu lintas yang tidak lengkap. Dengan menyimpan informasi tag waktu berarti setiap frame memiliki time stamp. Dengan time stamp, frame terkini akan terhubung dengan frame berikutnya. Hasilnya akan mempermudah untuk melacak objek.

Tahapan pada Modul Representasi Ciri ditunjukkan pada Gambar 4.10. Pada tahap ini, peneliti membuat skema representasi. Skema representasi ini dapat dilihat lebih jelas pada Gambar 5.6. Skema ini diperlukan pada proses representasi. Skema representasi ciri dan ciri-ciri objek menjadi masukan pada proses representasi untuk menghasilkan ciri dalam format XML. Gambar A.7. pada Lampiran A menunjukkan ciri yang dibentuk ke dalam format XML.



Gambar 4.9 Modul Representasi Ciri

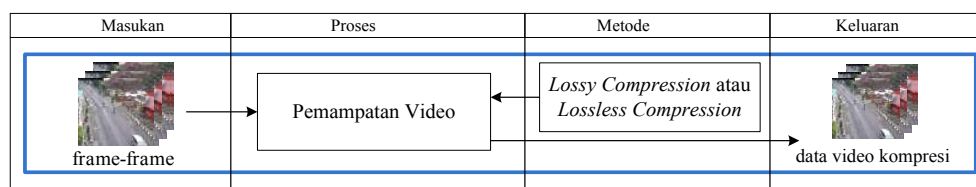


Gambar 4.10 Tahapan pada Modul Representasi Ciri

4.1.4 Modul Pemampatan Video

Pemampatan Video dapat mengatasi permasalahan keterbatasan *bandwidth* dan *processing power* CPU (Sutoyo dkk., 2009; Gibson, 2000). Data video dimampatkan untuk meminimalkan ukuran video. Tujuannya agar mempercepat proses pengiriman video. Metode yang digunakan untuk pemampatan video adalah *Lossy Compression*.

Gambar 4.11 menunjukkan tahapan pada Modul Pemampatan Video. Frame-frame yang diterima dari Modul Sensor Video menjadi masukan pada Modul Pemampatan Video. Modul Pemampatan Video memampatkan data video untuk meminimalkan ukuran *video*. Beberapa alternatif metode yang dapat digunakan pada Modul Pemampatan Video adalah metode *Lossy Compression* dan *Lossless Compression*. Hasilnya diperoleh data video yang telah dimampatkan / data video kompresi. Hasil ini tidak disimpan melainkan langsung diberikan ke Modul Transmisi Data.



Gambar 4.11 Tahapan pada Modul Pemampatan Video

Format video hasil pemampatan yang baik adalah format video yang cocok dengan kebutuhan pengiriman dan penyimpanan data. Format video beragam yaitu *.AVI (Audio Video Interleave)*; *.FLV (Flash Video)*; *.MP4 (Moving Picture 4)*; *.MOV (Apple's QuickTime container format)*; *.VOB (DVD Video Object)*; *.MKV (Mastroska)*; *.MPEG (Moving Picture Expert Group)* dan *.3GP* atau *3GPP (3rd Generation Partnership Project)*. Format data video yang diperoleh dari CCTV adalah *.3gp*. Dengan digunakannya format *.3gp* berarti proses pemampatan video telah dikerjakan.

Alasan penggunaan modul pemampatan video karena ada permintaan polisi yang tidak bisa ditawar untuk bukti pelanggaran lalu lintas. Sedangkan alasan penggunaan format *.3gp* karena ukuran gambar dan *bandwidth* 3GP sangat

kecil sehingga dapat mengirimkan video dan audio dengan cepat (ETSI, 2011). 3GP termasuk *Lossy Compression*. Metode *Lossy Compression* dipilih karena rasio pemampatannya yang lebih tinggi dari metode *Lossless Compression*.

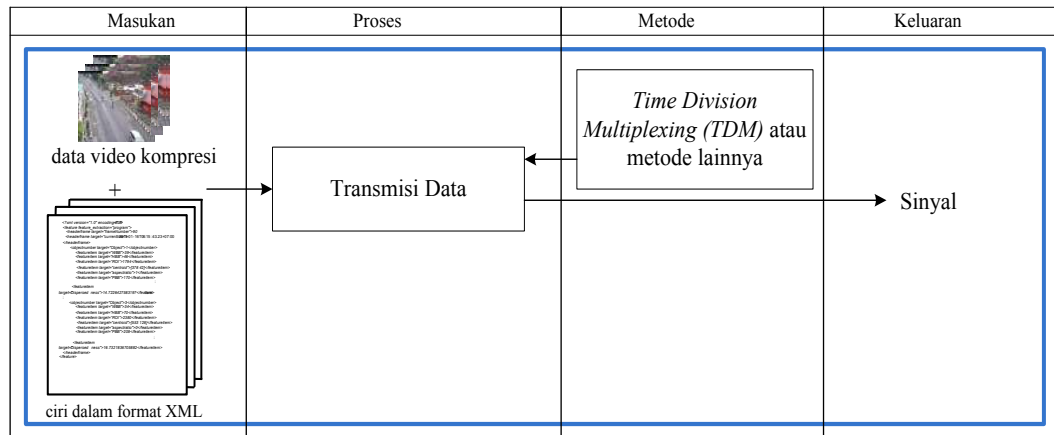
4.1.5 Modul Transmisi Data

Modul Transmisi Data berfungsi untuk mengirimkan ciri-ciri dan data video ke aplikasi yang membutuhkan (Bubenikova dkk., 2013). Modul Transmisi Data diperlukan untuk memastikan data berhasil dikirim dan diterima oleh pihak yang membutuhkan. Data yang dikirim Modul Transmisi Data adalah data video lalu lintas yang sudah dimampatkan dan ciri-ciri objek lalu lintas dalam bentuk XML. Data-data tidak disimpan melainkan langsung dikirimkan ke DCCU dan LPU. Salah satu metode yang dapat digunakan pada Modul Transmisi Data adalah *Time Division Multiplexing (TDM)*, namun tidak menutup kemungkinan untuk menggunakan metode yang lain.

Alasan diperlukannya Modul Transmisi Data adalah :

1. karena pentingnya memastikan ciri-ciri dan data video yang telah dimampatkan berhasil dikirim.
2. karena Pihak Kepolisian dan Dinas Perhubungan sangat membutuhkan ciri-ciri dan data video sehingga mereka harus yakin ciri-ciri dan data video telah diterima dengan baik dan dapat dibaca dengan benar.

Gambar 4.12 menunjukkan tahapan pada Modul Transmisi Data. Masukan data pada Modul Transmisi Data adalah data video yang telah dimampatkan dan ciri-ciri objek dalam format XML. Modul Transmisi Data mengubah data video yang telah dimampatkan dan ciri-ciri objek dalam format XML menjadi sinyal. Modul Transmisi Data mengirimkannya ke DCCU atau LPU. Salah satu metode yang dapat digunakan pada transmisi data yaitu *Time Division Multiplexing (TDM)*, namun tidak menutup kemungkinan untuk menggunakan metode yang lain. Modul Transmisi Data diasumsikan sudah dikerjakan sehingga pada penelitian ini tidak dibahas lebih lanjut.



Gambar 4.12 Tahapan pada Modul Transmisi Data

4.3 Alat Penelitian

Pada penelitian ini digunakan peralatan untuk menunjang pelaksanaan penelitian. Lampiran F menunjukkan rekomendasi spesifikasi perangkat keras untuk *Smart Video Sensor* (SVS) bila diimplementasikan. Penelitian ini menggunakan peralatan yang terdiri dari perangkat keras dan perangkat lunak. Spesifikasi perangkat keras yang digunakan sebagai berikut:

1. Prosesor bertipe Intel Core i3
2. RAM dengan kapasitas 8 GB
3. Harddisk dengan kapasitas 500 GB

Spesifikasi perangkat lunak yang digunakan sebagai berikut:

1. Sistem operasi Windows 8 64bit
2. *Matlab* 2015a sebagai perangkat lunak untuk membangun dan mengembangkan pemodelan ekstraksi ciri.

4.4 Data Penelitian

Penelitian ini menggunakan data primer yang diperoleh dari Dinas Perhubungan Propinsi Daerah Istimewa Yogyakarta. Data primer ini berupa data video lalu lintas. Lokasi yang diambil untuk penelitian adalah Propinsi Daerah

Istimewa Yogyakarta. Hasil pengumpulan data ini digunakan untuk pengembangan Arsitektur *Smart Video Sensor* (SVS). Sampel data primer ini kualitasnya baik dan secara representatif dapat digunakan sebagai data penelitian. Data video yang digunakan berasal dari 3 titik ATCS (*Area Traffic Control System*) Pemprov Yogyakarta yaitu Galeria, Condong Catur dan Babar Sari. Data video yang digunakan adalah data video yang direkam pada pagi, siang dan sore hari. Data video yang digunakan hanya kendaraan bermotor yang menghadap depan dan menghadap belakang. Data video berukuran 352 x 288 dan kerapatan video 30 fps. Jumlah frame untuk setiap video yang digunakan adalah 1000 frame. Data set yang digunakan dibagi 2 yaitu untuk data pelatihan dan data pengujian. Data video yang digunakan untuk menentukan nilai K pada K Nearest Neighbor (KNN) adalah Galeria_8, CondongCatur_11, Babarsari_8.

4.5 Rencana Pengujian

Rencana pengujian dari metode yang diusulkan bertujuan untuk mendemonstrasikan bagaimana Smart Video Sensor bekerja. Rencana pengujian dilakukan untuk memastikan bahwa ciri objek lalu lintas dapat digunakan oleh aplikasi lain sesuai kebutuhan di DCCU atau LPU. Oleh karena itu, ciri-ciri objek yang telah di terima, perlu diuji.

Rencana pengujian ini terdiri dari:

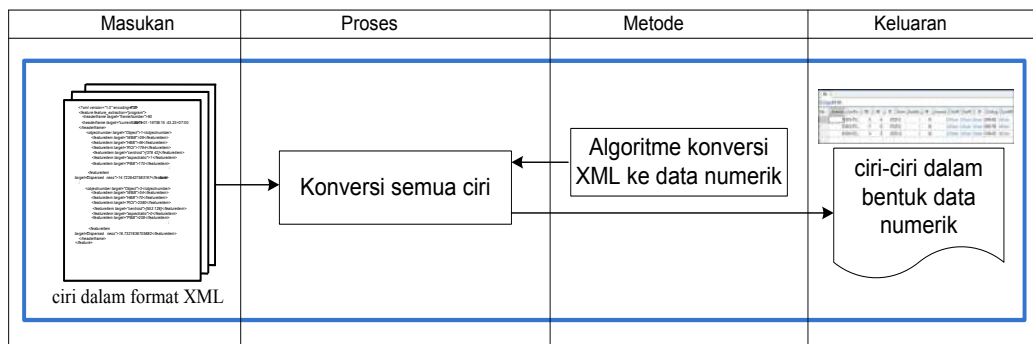
- a. rencana pengujian pada representasi ciri dalam format XML,
- b. rencana pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS yaitu untuk mengetahui akurasi jenis kendaraan,
- c. rencana pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS yaitu untuk mengetahui selisih dari dua posisi centroid dari Kalman Filter dan centroid dari ciri objek yang berasal dari SVS.

4.5.1 Rencana pengujian pada representasi ciri dalam format XML

Rencana pengujian pertama dengan merepresentasikan ciri untuk membuktikan bahwa ciri-ciri objek yang telah dikirim sama dengan ciri yang

telah diterima. Rencana pengujian kedua dengan membuat model klasifikasi kendaraan. Rencana pengujian ketiga dengan membuat model pelacakan kendaraan sesuai jenisnya. Dengan berhasilnya pengujian-pengujian ini maka ciri-ciri objek lalu lintas dalam format XML dapat digunakan untuk aplikasi lain sesuai kebutuhan.

Gambar 4.13 menunjukkan tahapan pengujian pada representasi ciri. Masukan pada tahap pengujian representasi ciri adalah ciri dalam format XML. Ciri dalam format XML di konversi menggunakan algoritme konversi XML ke data numerik. Keluaran pada tahap pengujian representasi ciri adalah ciri-ciri dalam bentuk data numerik.



Gambar 4.13 Tahapan pengujian pada representasi ciri

Tujuan pengujian representasi ciri untuk membuktikan bahwa ciri yang dikirim dalam bentuk XML itu dapat dibaca dan dapat dilihat kesesuaiannya dengan video framenya. Pengujian representasi ciri dilakukan dengan memastikan bahwa ciri yang dikirim sama dengan ciri yang diterima. Jika ciri-ciri yang dikirim sama dengan ciri yang diterima maka skema representasinya betul. Ciri yang telah dikirim oleh SVS dalam format XML ke DCCU perlu di konversi ke bentuk data numerik.

Langkah-langkah yang harus dilakukan untuk konversi ciri dalam format XML adalah:

1. konversi ciri dari format XML ke struktur array. Gambar A.8. pada Lampiran A menunjukkan langkah-langkah konversi ciri dari format XML ke dalam struktur array.

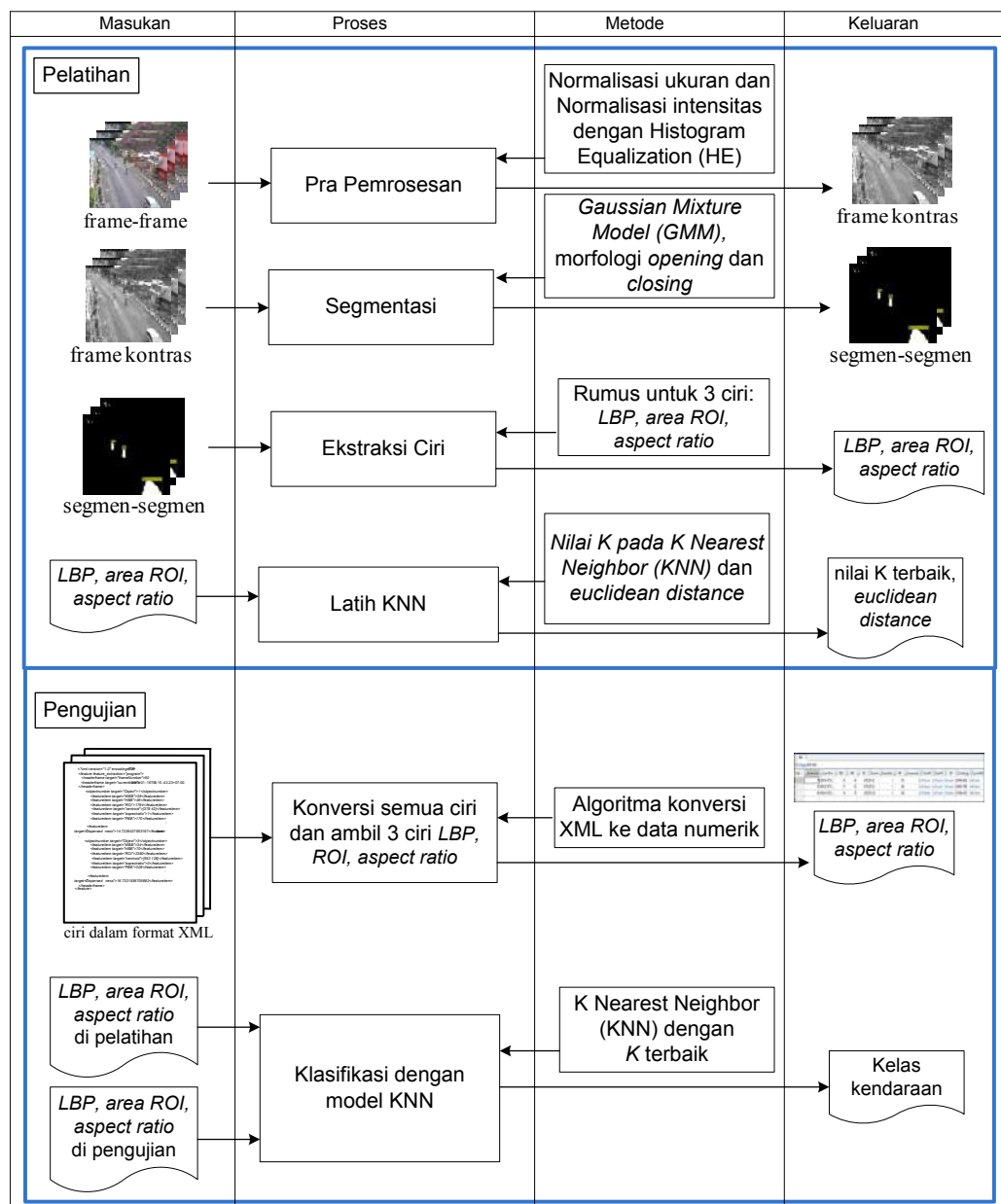
2. ubah variabel *string* ke variabel *number*. Gambar A.9. pada Lampiran A menunjukkan cara mengubah variabel *string* ke variabel *number*.
3. ubah dalam bentuk tabel. Gambar A.10. pada Lampiran A menunjukkan cara mengubah dalam bentuk tabel.

4.5.2 Rencana pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS

Gambar 4.14 menunjukkan tahapan pada model klasifikasi kendaraan mengikuti Arsitektur *Smart Video Sensor*. Tahapan ini meliputi pelatihan dan pengujian. Pada tahap pelatihan dilakukan proses pra pemrosesan, segmentasi, ekstraksi ciri dan latih KNN. Pada tahap pengujian dilakukan konversi ciri dalam format XML dan klasifikasi dengan model KNN menggunakan K terbaik.

Langkah-langkah yang dilakukan pada pelatihan. Langkah pertama, frame-frame pada pra pemrosesan dilakukan normalisasi ukuran dan normalisasi intensitas menggunakan metode *Histogram Equalization (HE)* untuk mendapatkan frame yang lebih kontras. Kedua, frame yang lebih kontras ini di segmentasi menggunakan metode *Gaussian Mixture Model (GMM)*. Segmentasi ini juga menggunakan morfologi *opening* dan *closing*. Hasil segmentasi adalah segmen-segmen. Ketiga, segmen-segmen di beri label. Satu label menyatakan satu objek. Objek di ekstrak cirinya menggunakan rumus ciri. Ciri yang diekstrak adalah *Local Binary Pattern (LBP)*, area *Region of Interest (ROI)* dan *aspect ratio*. Alasan pemilihan tiga ciri ini karena *Local Binary Pattern (LBP)* memiliki 59 dimensi yang telah diseleksi dari 256 dimensi sehingga dapat digunakan untuk membedakan obyek dalam klasifikasi (Rabiu, 2013; Firdaus dkk., 2016), area ROI merupakan area objek berbentuk segi empat yang dapat digunakan untuk klasifikasi (Mithun dkk., 2012), *aspect ratio* dapat membedakan jenis kendaraan berdasarkan besar rasio yang dimilikinya (Kota dan Rao, 2013). Keempat, ciri-ciri yang telah diekstrak kemudian dipakai untuk melatih *K-Nearest Neighbor (KNN)* menggunakan alat ukur *Euclidean distance* sehingga menghasilkan nilai K dan *Euclidean distance*. Rumus *Euclidean distance* yang digunakan ada pada persamaan (3.112-3.115).

Pada tahap pengujian dilakukan proses konversi dan klasifikasi. Semua ciri dalam format XML di konversi menggunakan algoritme konversi XML ke data numerik. Ciri yang diambil adalah *Local Binary Pattern (LBP)*, area *Region of Interest (ROI)* dan *aspect ratio*. Dari LBP, area ROI, *aspect ratio* di pelatihan dan LBP, area ROI, *aspect ratio* di pengujian serta nilai K terbaik dari KNN maka dilakukan klasifikasi dengan model KNN sehingga diperoleh kelas kendaraan.



Gambar 4.14 Tahapan pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS

Pengujian pada klasifikasi kendaraan bertujuan untuk mengetahui seberapa baik ciri objek lalu lintas yang di rekonstruksi dalam ciri dalam format XML. Parameter keberhasilan diukur dengan seberapa baik ciri dapat dikenali sesuai kelas kendaraannya. Pada pengujian klasifikasi kendaraan ini, kendaraan akan diklasifikasi menggunakan ciri objek yang sudah diperoleh dari SVS. Ciri yang dipilih untuk klasifikasi kendaraan adalah: *Local Binary Pattern* (LBP), area *Region of Interest* (ROI) atau *area*, dan *aspect ratio*. Algoritme yang digunakan untuk mengelompokkan kendaraan adalah: *K-Nearest Neighbor* (KNN).

Alat ukur yang digunakan untuk pengujian pada klasifikasi kendaraan adalah *confusion matrix*. Sebuah *confusion matrix* menunjukkan klasifikasi yang memprediksi kelas secara aktual. *Confusion matrix* menurut Santra dan Christy, (2012) dan Visa dkk., (2011) dapat ditunjukkan pada Tabel 4.1.

Tabel 4.1 *Confusion Matrix* (Santra dan Christy, 2012; Visa dkk., 2011)

	<i>Predicted Negative</i>	<i>Predicted Positive</i>
<i>Actual Negative</i>	a	b
<i>Actual Positive</i>	c	d

dengan:

a adalah jumlah prediksi negatif yang benar.

b adalah jumlah prediksi positif yang salah.

c adalah jumlah prediksi negatif yang tidak benar.

d adalah jumlah prediksi positif yang benar.

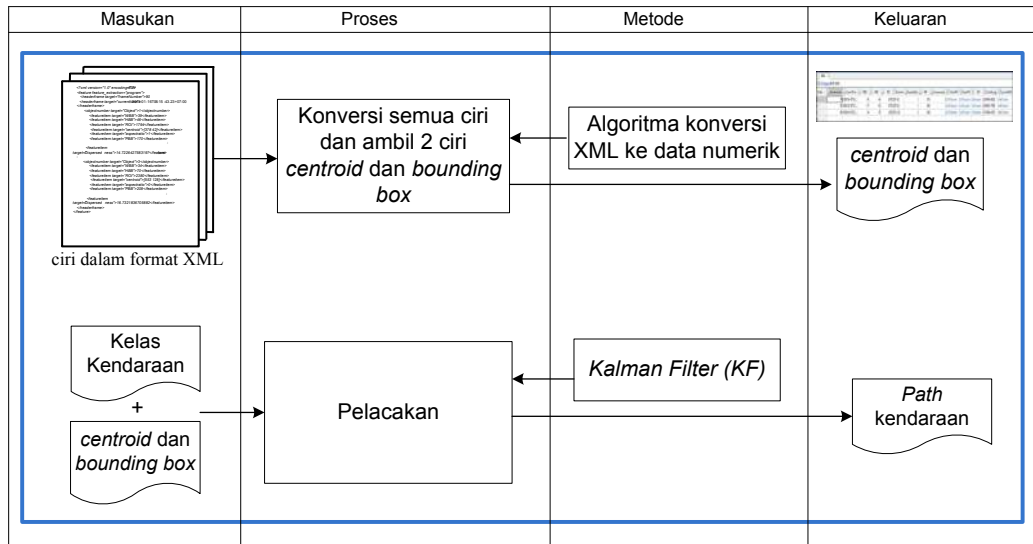
Pengukuran yang digunakan dalam klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS adalah *accuracy* (akurasi), seperti yang didefinisikan pada persamaan (4.1). Akurasi adalah tingkat kedekatan pengukuran terhadap nilai yang sebenarnya. Alasan pemilihan pengukuran menggunakan akurasi karena pengukuran ini juga digunakan oleh Chen dkk., (2012) dalam mengklasifikasi kendaraan.

$$Akurasi = \frac{a + d}{a + b + c + d} \quad (4.1)$$

4.5.3 Rencana pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS

Pada pengujian pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS, kendaraan yang telah diklasifikasi lalu dilacak berdasarkan jenis kendaraannya menggunakan ciri objek yang sudah diperoleh dari SVS. Alasan pengujian SVS dengan membuat pelacakan kendaraan berdasarkan jenisnya adalah karena pihak kepolisian dan Dinas Perhubungan membutuhkan informasi mengenai apa dan siapa yang menjadi penyebab kecelakaan serta informasi mengenai kondisi lalu lintas terkini. Ciri yang dipilih untuk pelacakan kendaraan berdasarkan jenis kendaraan adalah: *centroid* dan *bounding box* (Teoh dan Bräunl, 2012; Huang dkk., 2012). Algoritme yang digunakan adalah *kalman filter algorithm* (Teoh dan Bräunl, 2012).

Parameter keberhasilan pengujian diukur dengan mengetahui seberapa kecil perbedaan antara nilai ciri *centroid* dari representasi XML dengan nilai ciri *centroid* dari metode *Kalman Filter*. Gambar 4.15 menunjukkan tahapan pengujian pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari *Smart Video Sensor (SVS)*. Pada pelacakan kendaraan ini, semua ciri dalam format XML dikonversi menggunakan algoritme konversi XML ke data numerik. Selanjutnya, ciri yang diambil adalah *centroid* dan *bounding box*. Masukan model pelacakan kendaraan ini adalah kelas kendaraan, *centroid* dan *bounding box*. Model pelacakan kendaraan ini menggunakan metode *Kalman Filter* (KF). Rumus *Kalman Filter* yang digunakan ada pada persamaan (3.116-3.133). Metode *Kalman Filter* dipilih karena metode ini dapat memprediksi lokasi terbaik untuk kendaraan dalam frame video berikutnya (Liu dan Yang, 2010; Purwanto dkk., 2013; Li dkk., 2014; Teoh dan Bräunl, 2012). Keluaran dari model pelacakan kendaraan ini adalah *path* kendaraan.



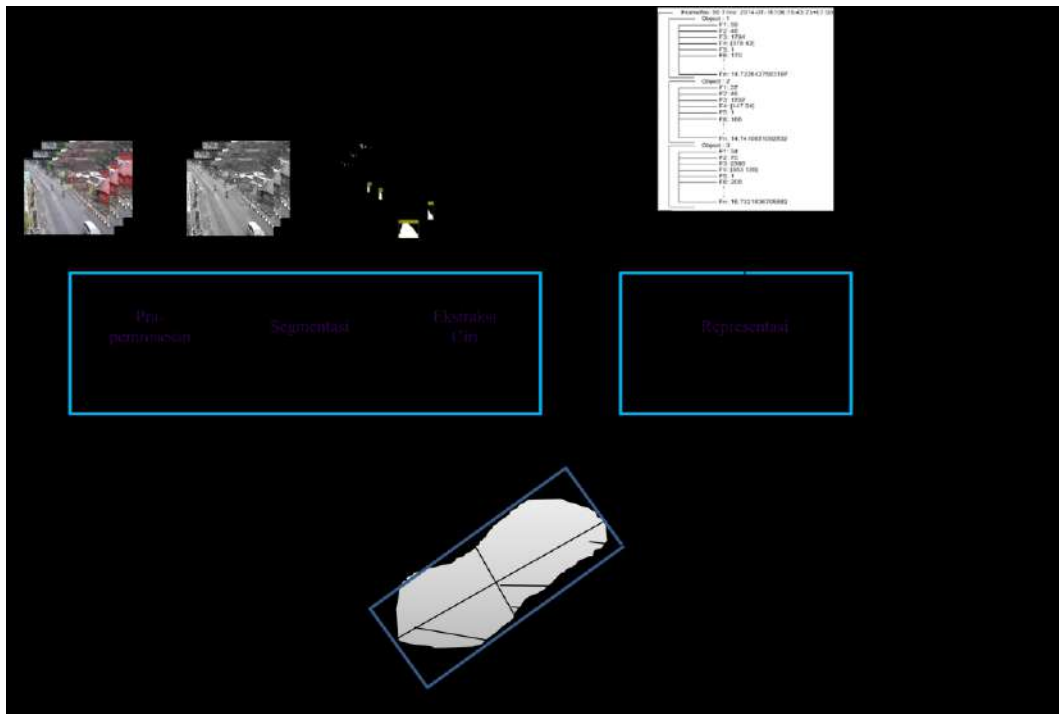
Gambar 4.15 Tahapan pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS

BAB 5

HASIL DAN PEMBAHASAN

5.1 Ekstraksi Ciri dan Representasi Ciri dalam SVS

Fokus SVS penelitian ini adalah Ekstraksi Ciri dan Representasi Ciri. Gambar 5.1 menunjukkan tahapan pada Ekstraksi Ciri dan Representasi Ciri. Modul Ekstraksi Ciri melakukan proses Pra Pemrosesan, Segmentasi dan Ekstraksi Ciri. Modul Ekstraksi Ciri menghasilkan ciri-ciri objek. Skema representasi dibentuk satu kali. Ciri-ciri objek dan skema representasi menjadi masukan Modul Representasi Ciri. Modul Representasi Ciri melakukan representasi. Hasilnya adalah ciri dalam format XML.



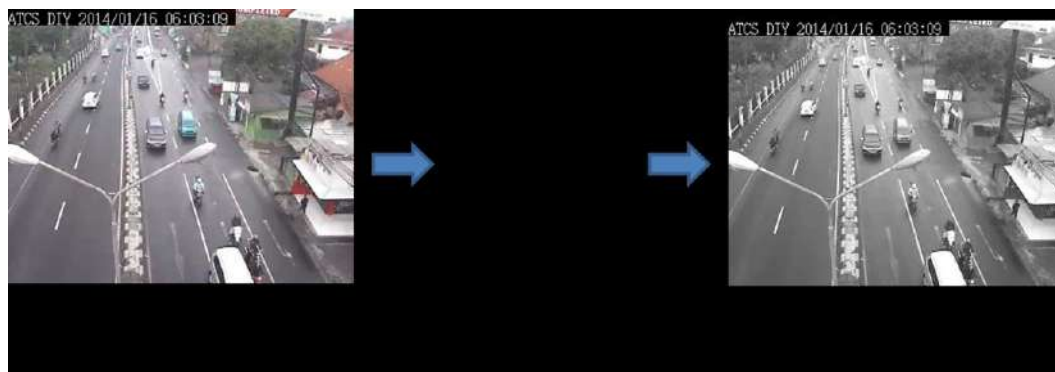
Gambar 5.1 Model Ekstraksi Ciri dan Representasi Ciri

Alur kerja SVS sesuai Gambar 5.1 dimulai dari Pra Pemrosesan, Segmentasi, Ekstraksi Ciri dan Representasi. Proses dimulai dari *scene* sebagai masukan SVS. SVS merekam *scene*. Data masukan dari sensor video adalah *scene*.

Keluaran sensor video adalah frame-frame. Frame-frame ini ditingkatkan kualitas citranya menggunakan *Histogram Equalization* sehingga frame-frame ini lebih kontras. Lalu, *frame-frame* disegmentasi menjadi kelompok segmen. Tiap segmen diberi label. Ada 39 ciri diekstrak dari tiap label. Pemrosesan data dari Gambar 5.1 dibahas lebih detail di sub bab 5.1.1.

5.1.1 Pra pemrosesan

Tujuan Pra-pemrosesan adalah untuk mendapatkan citra yang lebih berkualitas. Ada dua tahap pra pemrosesan yang dilakukan yaitu normalisasi ukuran dan normalisasi intensitas. Pada normalisasi ukuran, data video dari semua lokasi yang digunakan untuk dataset disamakan ukurannya menjadi 352 x 288. Alasan pemilihan ukuran ini karena ukuran itulah yang terkecil dari kumpulan video dataset yang ada. Pada pra pemrosesan data video yang sudah dinormalisasi ukurannya kemudian data video itu diekstrak menjadi frame-frame. Gambar A.1 pada Lampiran A menunjukkan Algoritme Ekstrak video ke frame-frame. Hasilnya, frame dibentuk format .jpeg. Alasan format jpeg karena jpeg sudah termampatkan dengan metode *Lossy Compression*. Frame diberi nama sesuai dengan nomor frame pada video. Frame tidak disimpan ke file tetapi langsung diolah di SVS kemudian langsung dikirim ke DCCU. Pada normalisasi intensitas, metode yang digunakan adalah *Histogram Equalization (HE)*. *Histogram Equalization* adalah suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Alasan pemilihan metode *Histogram Equalization* karena metode ini dapat meningkatkan kualitas citra secara menyeluruh (Hawlder dan Hossain, 2014; Xu dkk., 2014; Diego, 2014; Yadav dkk., 2014; Kadir dan Susanto, 2013). *Histogram Equalization* ini memerlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari histogram. Gambar 5.2 menunjukkan citra sebelum dan sesudah menggunakan *Histogram Equalization*.



Gambar 5.2 Citra sebelum dan sesudah *Histogram Equalization*

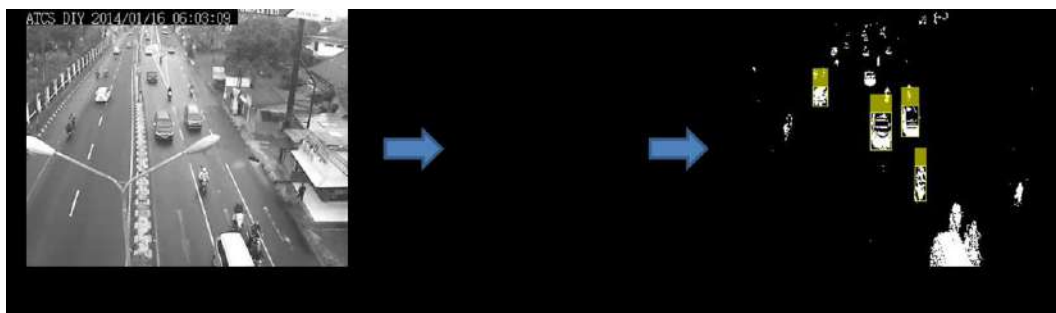
5.1.2 Segmentasi

Ada dua kegunaan segmentasi. Pertama, segmentasi berguna untuk memisahkan objek dengan latar objeknya. Kedua, segmentasi juga berguna untuk memisahkan antara objek yang satu dengan objek yang lain. Metode yang digunakan untuk segmentasi adalah *Gaussian Mixture Model (GMM)*. Data masukan segmentasi adalah citra yang kontras. Hasil keluarannya adalah segmen-segmen.

Alasan pemilihan metode GMM karena:

1. metode ini lebih adaptif terhadap perubahan latar objek (Gonzalez dan Woods, 2012; Santosh dkk., 2013; Ping dkk., 2015).
2. akurasiya bagus dan biaya komputasinya rendah (Shahbaz dkk., 2016).
3. pembaruan latar objek sangat cepat dan real-time, mudah menganggap kendaraan berhenti sebagai latar objek.

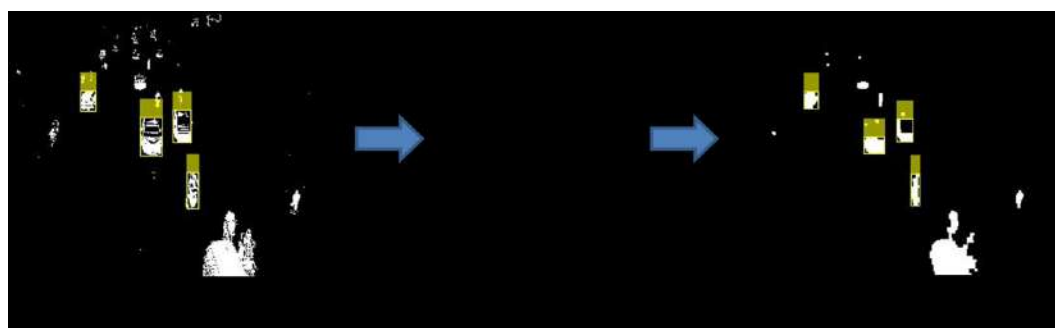
Pada tahap awal, *background model* diciptakan untuk membuat latar objek menggunakan GMM. *Background model* diciptakan dengan meniadakan objek yang bergerak. Setelah itu, sistem akan mengamati objek yang bergerak. Objek yang bergerak atau kecepatan $\neq 0$ berarti *foreground* (objek). Sedangkan objek yang tidak bergerak atau kecepataannya = 0, itulah *background* (latar objek). Setelah latar objek terpisah dengan objek, maka terbentuk segmen-segmen. Segmen adalah kelompok piksel yang memiliki ciri tertentu. Gambar 5.3 menunjukkan frame sebelum dan sesudah menggunakan GMM.



Gambar 5.3 Citra sebelum dan sesudah segmentasi menggunakan *Gaussian Mixture Model*

Segmentasi juga melakukan post-pemrosesan. Tujuan post-pemrosesan adalah untuk menghilangkan *noise*. Post-pemrosesan yang digunakan adalah morfologi *opening* dan *closing*. Morfologi *opening* secara umum menebalkan garis-garis bentuk. Operasi *opening* adalah operasi erosi yang diikuti operasi dilasi dengan menggunakan elemen penstruktur yang sama. Morfologi *closing* menipiskan garis-garis bentuk. Morfologi *closing* menghilangkan lubang kecil dan mengisi gap pada garis-garis bentuk (*contour*). Operasi *closing* dilaksanakan dengan melakukan operasi dilasi kemudian diikuti dengan operasi erosi (Gonzalez dkk., 2008).

Data masukan *post pemrosesan* adalah segmen-segmen. Hasil post pemrosesan adalah segmen-segmen yang telah terisi penuh. Gambar 5.4 menunjukkan sebelum dan sesudah menggunakan morfologi *opening* dan *closing*.



Gambar 5.4 Citra sebelum dan sesudah morfologi *opening* dan *closing*

Parameter GMM yang digunakan pada penelitian ini adalah $K = 4$ karena cocok untuk lingkungan terbuka (Shahbaz dkk., 2016). Jumlah training frame = 30 fps, *minimum background ratio* = 0,25, $\alpha = \text{learning rate} = 0,001$. Pada





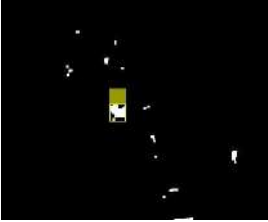




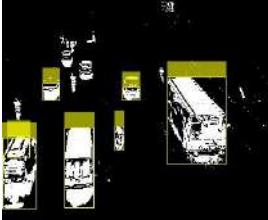





awalnya, beberapa nilai K , α dan τ diubah secara *Learning rate* α berguna untuk mengontrol seberapa cepat atau lambat komponen *Gaussian* melakukan kebaruan contohnya *weight*, *mean* dan *variance*. *Learning rate* juga digunakan untuk memastikan berapa lama komponen *Gaussian* tertentu akan dipertahankan modelnya. Nilai yang ditetapkan tergantung pada kompleksitas latar objeknya. Misalnya, latar objek yang berubah secara perlahan membutuhkan nilai α yang rendah dan latar objek yang berubah secara bertahap membutuhkan nilai α yang tinggi. Alasan pemilihan $\alpha = 0,001$ karena $\alpha = 0,001$ dapat mendeteksi objek yang bergerak dari yang sangat cepat sampai yang sangat lambat sehingga objek mudah disegmentasi.

Penentuan nilai K , α , τ pada penelitian ini dari uji coba dan diperkuat dari Penelitian Shahbaz, dkk. (2016). Pada awal penelitian, nilai K , α , τ di uji coba dengan nilai $K=3$, $\alpha = 0,01$ dan $\tau = 0,7$ namun hasilnya hanya beberapa objek yang terdeteksi dari tiap frame yang ada. Oleh karena itu, dilakukan pula beberapa percobaan, yaitu:

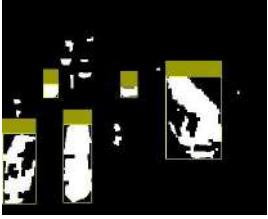





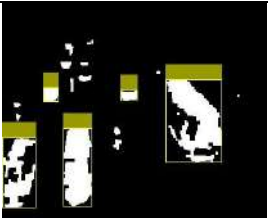

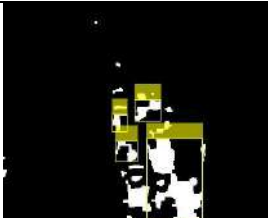
- a. $K=3$, $\alpha = 0,001$ dan $\tau = 0,25$ tanpa morfologi,
- b. $K=4$, $\alpha = 0,001$ dan $\tau = 0,25$ tanpa morfologi,
- c. $K=4$, $\alpha = 0,001$ dan $\tau = 0,5$ dengan morfologi,
- d. $K=2$, $\alpha = 0,001$ dan $\tau = 0,25$ dengan morfologi,
- e. $K=3$, $\alpha = 0,001$ dan $\tau = 0,25$ dengan morfologi dan
- f. $K=4$, $\alpha = 0,001$ dan $\tau = 0,25$ dengan morfologi.

Dari beberapa percobaan diperoleh hasil terbaik adalah $K=4$, $\alpha = 0,001$ dan $\tau = 0,25$ dengan morfologi. Penentuan ini dipilih karena pada setiap frame, objek yang terdeteksi lebih banyak dan lebih bagus dan hal ini juga diperkuat dari penelitian Shahbaz, dkk. (2016) yang menyatakan bahwa $K=4$ lebih cocok untuk lingkungan terbuka. Tabel 5.1 menunjukkan hasil segmentasi beberapa frame di Babarsari.

Tabel 5.1 Hasil segmentasi beberapa frame di Babarsari

	Babarsari_3, frame ke 800	Babarsari_4, frame ke 992	Babarsari_8, frame ke 412
basic frame			
Minimum background ratio = 0,7 $\alpha = 0,01$ K=3 Number of training frame=100 fps,			
Minimum background ratio = 0,25 $\alpha = 0,001$ K=3 Number of training frame=30 fps, tanpa morphology			
Minimum background ratio = 0,25 $\alpha = 0,001$ K=4 Number of training frame=30 fps, tanpa morphology			
Minimum background ratio = 0,5 $\alpha = 0,001$ K=4 Number of training frame=30 fps			

Tabel 5.1 (lanjutan)

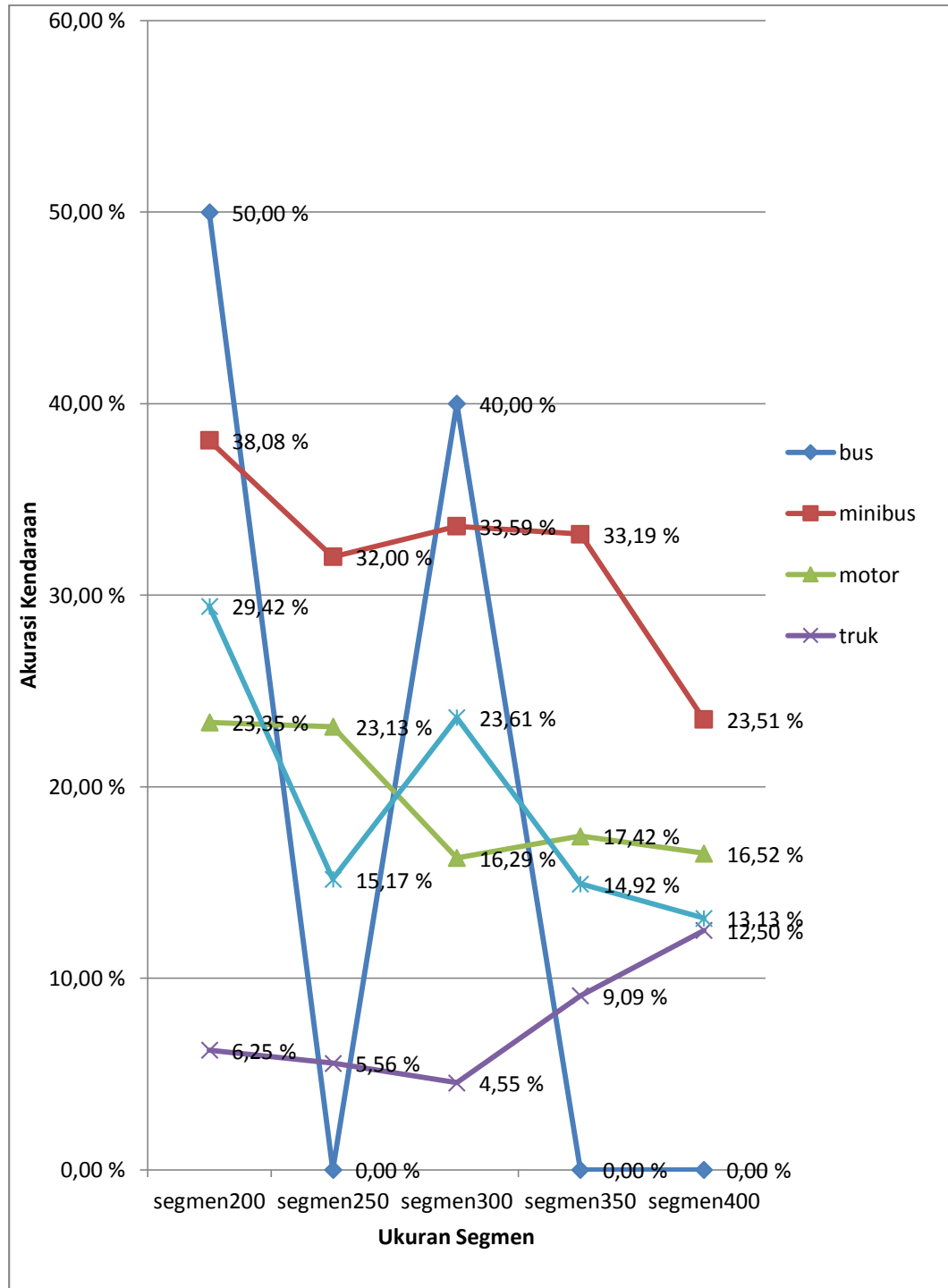
	Babarsari_3, frame ke 800	Babarsari_4, frame ke 992	Babarsari_8, frame ke 412
Minimum background ratio = 0,25 $\alpha = 0,001$ K=2 Number of training frame=30 fps			
Minimum background ratio = 0,25 $\alpha = 0,001$ K=3 Number of training frame=30 fps			
Minimum background ratio = 0,25 $\alpha = 0,001$ K=4 Number of training frame=30 fps			

5.1.3 Ekstraksi Ciri

Ciri adalah sifat yang dimiliki oleh suatu objek untuk membedakan dari objek yang lain. Ciri didefinisikan sebagai fungsi dari satu atau lebih pengukuran, yang masing-masing menentukan beberapa properti kuantitatif dari suatu objek, dan mengkuantifikasi beberapa karakteristik yang signifikan dari objek (Choras, 2007b). Tujuan ekstraksi adalah untuk mengambil atau mengekstraksi nilai – nilai unik dari suatu objek yang membedakan dengan objek yang lain.

Tiap segmen diberi label. Segmen yang diberi label adalah objek yang terdeteksi. Penentuan segmen ini dilakukan dari beberapa eksperimen untuk beberapa ukuran segmen yaitu 200, 250, 300, 350, 400. Caranya dengan menghitung rata-rata *confusion matrix* tiap segmen. Daerah yang digunakan

sebagai sampel adalah babarsari dengan nama video babarsari_8 dan jumlah frame yang digunakan 1814 frame. Gambar 5.5 memperlihatkan nilai rata-rata *confusion matrix* tertinggi diperoleh pada segmen 200.



Gambar 5.5 Hubungan antara ukuran segmen dengan akurasi kendaraan di Babarsari

Oleh karena itu, pelabelan diberikan bila segmen berukuran ≥ 200 . Alasan pemilihan ukuran segmen ≥ 200 karena meminimalkan kesalahan objek yang terdeteksi dan berdasarkan eksperimen diperoleh hasil bahwa segmen 200 memiliki nilai rata-rata tertinggi.

Segmen-segmen yang diberi label di ekstrak cirinya. Data masukan dari ekstraksi ciri adalah segmen-segmen. Hasil keluaran dari ekstraksi ciri adalah ciri-ciri objek-objek lalu lintas. Ciri yang dikumpulkan berjumlah 39 ciri seperti yang telah disebutkan pada Gambar 4.8. Alasan objek di ekstrak sebanyak 39 ciri karena ciri-ciri ini akan digunakan untuk identifikasi objek, deteksi objek, pengenalan objek, klasifikasi objek, pelacakan objek, perhitungan jumlah objek, perhitungan kecepatan objek, estimasi kemacetan lalu lintas, dan masukan *traffic parameter* lainnya yang diperlukan oleh pihak Kepolisian dan Dinas Perhubungan.

5.1.4 Representasi

Ciri-ciri objek lalu lintas yang telah diperoleh perlu direpresentasikan agar mudah dibaca. Tujuan representasi adalah mendeskripsikan informasi citra yang sesuai untuk digunakan (Gonzalez dan Woods, 2012). Representasi adalah penyajian citra dalam bentuk terstruktur. Representasi ciri obyek lalu lintas adalah kegiatan menyusun ciri-ciri objek lalu lintas dalam format XML. Format XML memiliki struktur yang dinamis.

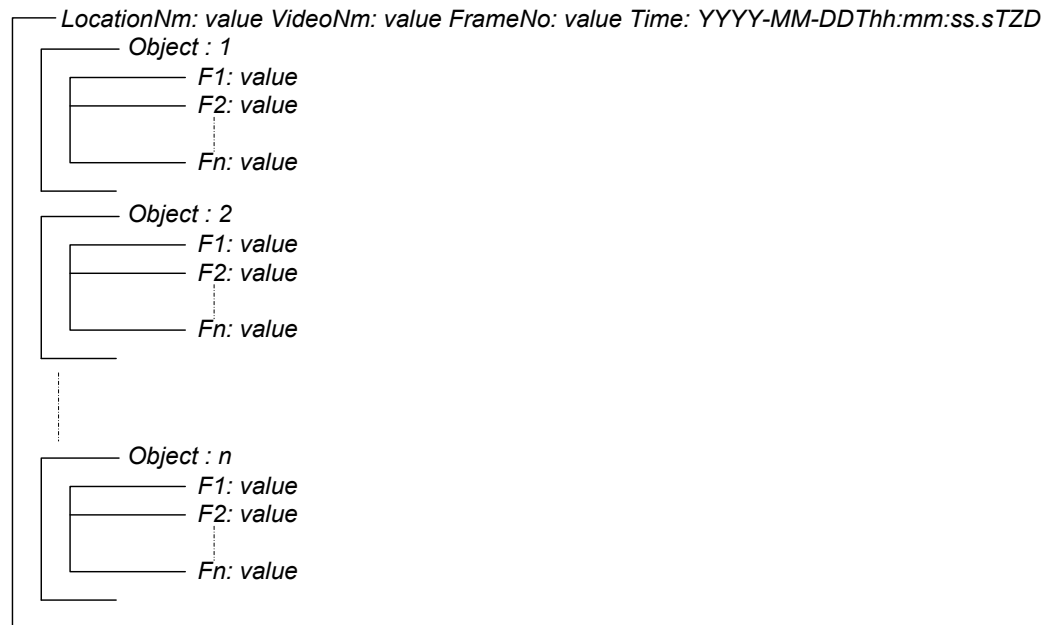
Alasan pemilihan representasi ciri dalam format XML karena:

1. Struktur XML berisi informasi yang tidak memerlukan format standard.
2. jumlah objek dalam tiap frame berbeda-beda sehingga format yang dinamis sangat dibutuhkan.
3. File XML berbentuk teks sehingga file XML dapat dibaca tanpa memerlukan bantuan software khusus (Bray dkk., 2013).
4. struktur XML memiliki kemudahan dalam mempertukarkan informasi di platform yang berbeda (Carey, 2007; Carey, 2010; Parsons dan Oja, 2012).
5. XML dapat menyimpan informasi tag waktu. Informasi tag waktu disimpan untuk menyelesaikan masalah data video lalu lintas yang tidak lengkap. Dengan menyimpan informasi tag waktu berarti setiap frame memiliki time

stamp. Dengan time stamp, frame terkini akan terhubung dengan frame berikutnya. Hasilnya akan mempermudah untuk melacak objek.

Alasan dibuatnya skema representasi adalah untuk memudahkan penyajian. Skema representasi dibuat dalam bentuk bagan terstruktur. Skema representasi dibuat hanya sekali di awal. Skema representasi ini memiliki 3 level. Pada level 1, skema representasi berisi informasi nama lokasi, nama video, nomor frame, tanggal dan waktu frame. Pada level 2, skema representasi berisi informasi objek di dalam frame. Pada level 3, skema representasi berisi informasi ciri-ciri yang dimiliki tiap objek. Skema representasi dari ciri-ciri objek lalu lintas ditunjukkan pada Gambar 5.6. Nama lokasi diperlukan untuk mengetahui daerah lokasi. Nama video diperlukan untuk mengetahui dengan pasti video yang direkam. Nomor frame diperlukan untuk mengetahui urutan frame itu. Tanggal dan waktu frame menggunakan format *Universal Time Format* (UTF). Time format ini sesuai dengan format internasional yang dispesifikasi dalam ISO 8601 (Wolf dan Wicksteed, 1997). Standar format yang digunakan adalah format tanggal lengkap, jam, menit, detik dan milidetik. Bentuknya: YYYY-MM-DDThh:mm:ss.sTZD. Contoh: 2015-10-28T12:05:45.55+07:00. Artinya frame ini direkam pada tanggal 28 Oktober 2015 jam 12:05:45:55 berdasarkan zona waktu GMT+7. Time format ini dipilih karena digunakan oleh World Wide Web. Time format ini digunakan untuk mempermudah penggunaan format waktu dalam satu bilangan. Dalam skema pada Gambar 5.6 dapat dilihat setiap satu frame terdiri dari beberapa objek. Tiap objek memiliki beberapa ciri.

Skema representasi menjadi dasar untuk pembentukan format XML. Skema representasi memiliki 3 (tiga) level untuk membentuk ciri dalam format XML yang terdiri dari *header frame*, jumlah objek dan ciri-ciri objek. Skema ini formatnya tidak standar karena jumlah dan panjang nilai ciri objek bervariasi. *Header frame* yang berisi nama lokasi, nama video, nomor frame dan tanggal frame beserta atributnya. Jumlah objek yang terdeteksi dalam frame itu dihitung berdasarkan jumlah label yang ada dalam 1 (satu) frame. Tiap objek yang terdeteksi di ekstrak ciri-cirinya. Setelah itu, tiap frame diberi nama file sesuai nomor framenya dengan *exention* .xml.



Gambar 5.6 Skema representasi dari ciri-ciri objek lalu lintas yang diusulkan

Skema representasi ciri pada Gambar 5.6 ditunjukkan pada Gambar 5.7 dalam struktur XML. Tag `<?xml version="1.0" encoding="utf-8"?>` menunjukkan versi XML yang dipergunakan. Tag `<Z feature_XML="list">` merupakan tanda awal dari pembacaan satu frame. Lokasi video ditunjukkan dengan `<HFrame Value="LocationNm">Location Name</HFrame>`. Nama video ditunjukkan dengan `<HFrame Value="VideoNm">Video Name</HFrame>`. Nomor frame ditunjukkan dengan `<HFrame Value="FrameNo"> Frame Number value.</HFrame>` Waktu rekam frame ditunjukkan dengan `<HFrame Value="FTimes">Date Time value </HFrame>`. Dalam satu frame bisa terdapat beberapa objek. Nomor objek ditunjukkan dengan `<ObjectNo Value="Object"> n </ObjectNo>`. Jumlah objek ditunjukkan dengan n. Tag `<Feature Value=" Featuren">value</Feature>` menunjukkan Featuren sebagai ciri ke-n dan value sebagai nilai ciri. Jumlah ciri ditunjukkan dengan n. Untuk menandai bahwa satu frame telah selesai maka diberi simbol `</Z>`.

```

<?xml version="1.0" encoding="utf-8"?>
<Z feature_XML="list">
  <HFrame Value="LocationNm">Location Name </HFrame>
  <HFrame Value="VideoNm">Video Name</HFrame>
  <HFrame Value="FrameNo">Frame Number value
  <HFrame Value="FTimes">Date Time value </HFrame>
  <ObjectNo Value="Object1"> value </ObjectNo>
    <Feature Value="Feature1"> value </Feature>
    <Feature Value="Feature2"> value </Feature>
    :
    <Feature Value="Feature39"> value </Feature>
  :
  <ObjectNo Value="Objectn"> value </ObjectNo>
    <Feature Value="Feature1">value</Feature>
    <Feature Value="Feature2">value</Feature>
    :
    <Feature Value="Feature39">value</Feature>
  </HFrame>
</Z>

```

Gambar 5.7 Struktur XML

5.2 Hasil Pengujian

Hasil pengujian dilakukan untuk memberi kepastian bahwa ciri-ciri objek lalu lintas yang telah dihasilkan oleh SVS dapat digunakan pada aplikasi lain. Hasil pengujian ini dilakukan setelah representasi ciri dikirim ke DCCU atau LPU. Hasil pengujian ini meliputi:

- hasil pengujian pada representasi ciri dalam format XML,
- hasil pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS,
- hasil pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS.

Pengujian dilakukan pada 3 lokasi data yang berbeda di Yogyakarta. Lokasinya meliputi: Babar Sari, Galeria, dan Condong Catur. Pengujian menggunakan 1000 frame tiap lokasi dengan ukuran video = 352 x 288 dan kerapatan video = 30 fps.

5.2.1 Hasil pengujian pada representasi ciri dalam format XML

Representasi ciri dalam format XML dilakukan untuk membuktikan bahwa ciri dapat dibaca dan dapat dilihat kesesuaiannya dengan video framenya. Ini juga dilakukan untuk memastikan bahwa ciri yang berhasil dikirim dapat dibaca dengan benar. Jika ciri berhasil dikirim dan ciri dapat dibaca maka skema representasinya betul.

Representasi ciri pada penelitian ini menggunakan 2 sampel frame data yang berbeda. Awalnya, SVS menghitung jumlah objek yang ada di dalam area itu. Kedua, SVS menghitung ciri tiap objek. Ketiga, SVS membentuk representasi ciri dalam format XML. Keempat, SVS mengirimkan ciri dalam format XML ke DCCU atau LPU. Kelima, sistem lain menerima ciri dalam format XML. Keenam, sistem lain membaca ciri dalam format XML. Untuk membaca ciri dalam format XML, sistem lain perlu mengubah ciri dalam format XML ke bentuk data numerik.

Ciri yang direpresentasikan 39 ciri. Jumlah ciri yang dikirimkan adalah 39. Jumlah dimensi ciri = 1229. Adapun rincian ciri beserta jumlah dimensi cirinya ditunjukkan pada Lampiran D. Waktu pemrosesan setiap frame berkisar antara 0,12 menit per frame sampai 1,12 menit per frame. Waktu pemrosesan setiap frame berbeda, tergantung bentuk dan jumlah objek yang terdeteksi. Ukuran file XML bervariasi. Ukuran file juga tergantung pada bentuk objek dan banyaknya objek yang terdeteksi. Objek yang berjumlah 1 (satu), ukuran file XML berkisar antara 9 KB sampai 87 KB. Objek yang berjumlah 2 (dua), ukuran file XML sama dengan 102 KB. Objek yang berjumlah 3 (tiga) atau lebih, ukuran file XML lebih besar dari 120 KB.

Ciri-ciri yang direpresentasikan meliputi *Width of the Bounding Box (WBB)*, *Height of the Bounding Box (HBB)*, *area ROI*, *centroid*, *aspectratio*, *Perimeter Bounding Box (PBB)*, dan *Dispersedness*. Untuk memudahkan pembacaan skema representasi maka tiap ciri diberi simbol. Simbol F1 untuk WBB. Simbol F2 untuk HBB. Simbol F3 untuk area *ROI*. Simbol F4 untuk *centroid*. Simbol F5 untuk *aspectratio*. Simbol F6 untuk PBB. Simbol F39 untuk *Dispersedness*. Pada Gambar 5.8, terdapat 3 objek lalu lintas yaitu motor1,

motor2 dan motor3. Objek lalu lintas yang direpresentasikan hanyalah objek lalu lintas yang bergerak.

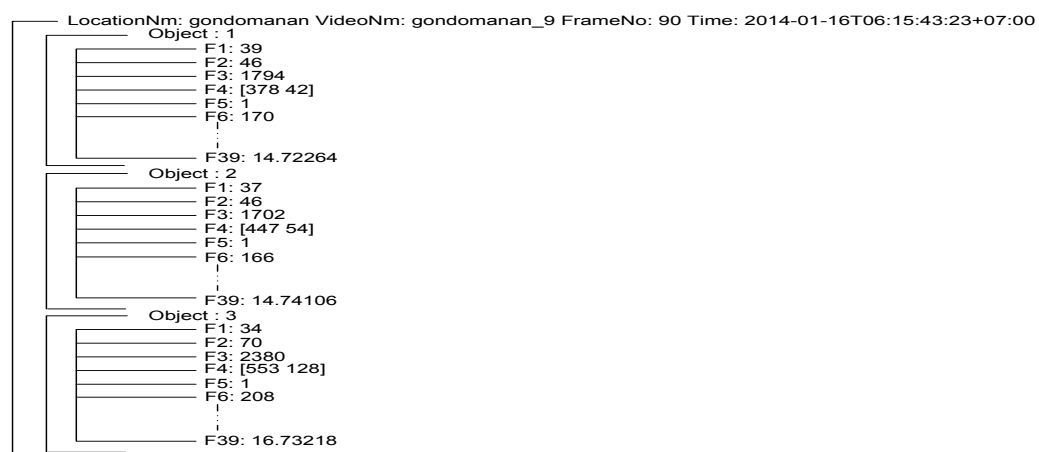


Gambar 5.8 Frame video 1

Gambar 5.9 menunjukkan skema representasi ciri dari frame video 1. LocationNm: gondomanan VideoNm: gondomanan_9 FrameNo: 90 Time: 2014-01-16T06:15:43.23+07:00 artinya nama lokasi gondomanan dengan nama video gondomanan_9 pada nomor frame 90 direkam pada tanggal 16 Januari 2014 jam 6 lebih 15 menit, 43 detik dan 23 milidetik berdasarkan zona waktu GMT + 7. Ada 3 objek yang direpresentasikan pada Gambar 5.9. Tiap objek ini memiliki 39 ciri. Nilai ciri objek pertama meliputi: WBB = 39, HBB = 46, area *ROI* = 1794, *centroid* = [378 42], *aspectratio* = 1, PBB = 170, dan *Dispersedness* = 14.72264. Nilai ciri objek kedua meliputi: WBB = 37, HBB = 46, area *ROI* = 1702, *centroid* = [447 54], *aspectratio* = 1, PBB = 166, dan *Dispersedness* = 14.74106. Nilai ciri objek ketiga meliputi: WBB = 34, HBB = 70, area *ROI* = 2380, *centroid* = [553 128], *aspectratio* = 0, PBB = 208, dan *Dispersedness* = 16.73218.

Gambar 5.10 menunjukkan ciri dalam format XML dari frame video 1 yang mewakili frame video yang dibentuk dalam XML. Tag `<?xml version="1.0" encoding="utf-8"?>` menunjukkan versi XML yang digunakan. Tag `<Z feature_XML="list">` merupakan tanda awal dari pembacaan satu frame. Lokasi frame ditunjukkan dengan `<HFrame Value="LocationNm"> gondomanan </HFrame>`. Nama video ditunjukkan dengan `<HFrame Value="VideoNm">`

gondomanan_9 </HFrame>. Nomor frame ditunjukkan dengan <HFrame Value="FrameNo"> 90 artinya *frame* ke 90 dari video yang direkam. Waktu rekam frame ditunjukkan dengan <HFrame Value="FTimes"> 2014-01-16T06:15:43.23+07:00 </HFrame> artinya frame video ini direkam pada tanggal 16 Januari 2014 pada jam 6 lebih 15 menit, 43 detik dan 23 milidetik berdasarkan zona waktu GMT+7. Dalam satu frame bisa terdapat beberapa objek. Nomor objek ditunjukkan dengan <ObjectNo Value="Object"> 1 </ObjectNo>. Jumlah objek yang di ekstrak adalah 39. Nama dan nilai ciri pertama dari objek pertama ditunjukkan dengan <Feature Value="WBB">39</Feature>. Nama dan nilai ciri kedua dari objek pertama ditunjukkan dengan <Feature Value="HBB">46</Feature>. Nama dan nilai ciri ketiga dari objek pertama ditunjukkan dengan <Feature Value="area ROI">1794</Feature>. Nama dan nilai ciri keempat dari objek pertama ditunjukkan dengan <Feature Value="centroid">[378 42]</Feature>. Nama dan nilai ciri kelima dari objek pertama ditunjukkan dengan <Feature Value="aspectratio">1</Feature>. Nama dan nilai ciri keenam dari objek pertama ditunjukkan dengan <Feature Value="PBB">170</Feature>. Nama dan nilai ciri terakhir dari objek pertama ditunjukkan dengan <Feature Value="Dispersedness">14.72264</Feature>. Pembacaan yang sama juga dilakukan untuk objek kedua dan ketiga. Semua objek lalu lintas bergerak dibaca cirinya sampai selesai. Tanda satu frame telah selesai maka diberi simbol </Z>.



Gambar 5.9 Skema representasi ciri dari frame video 1

```

<?xml version="1.0" encoding="utf-8"?>
<Z feature_XML="list">
<HFrame Value="LocationNm"> babarsari </HFrame>
<HFrame Value="VideoNm"> babarsari_4 </HFrame>
<HFrame Value="FrameNo"> 90
<HFrame Value="FTimes"> 2014-01-16T06:15:43.23+07:00 </HFrame>
  <ObjectNo Value="Object"> 1 </ObjectNo>
    <Feature Value="WBB">39</Feature>
    <Feature Value="HBB">46</Feature>
    <Feature Value="area ROI">1794</Feature>
    <Feature Value="centroid">[378 42]</Feature>
    <Feature Value="aspectratio">1</Feature>
    <Feature Value="PBB">170</Feature>
    :
    <<Feature Value="Dispersedness">14.72264</Feature>
    :
  <ObjectNo Value="Object"> 3 </ObjectNo>
    <Feature Value="WBB">34</Feature>
    <Feature Value="HBB">70</Feature>
    <Feature Value="area ROI">2380</Feature>
    <Feature Value="centroid">[553 128]</Feature>
    <Feature Value="aspectratio">0</Feature>
    <Feature Value="PBB">208</Feature>
    :
    <<Feature Value="Dispersedness">16.73218</Feature>
  </HFrame>
</Z>

```

Gambar 5.10 Ciri dalam format XML dari frame video 1

Gambar 5.11 menunjukkan keluaran ciri dalam format XML frame video 1 ke data numerik. Hasilnya menunjukkan bahwa ciri yang dikirim dapat dibaca dan nilai cirinya yang dikirim sama dengan nilai ciri yang diterima. Dengan demikian, pengujian representasi ciri telah berhasil dan skema representasinya sudah betul karena ciri yang dikirim sama dengan ciri yang diterima.

Gambar 5.12 menunjukkan *frame video 2*. LocationNm: galeria VideoNm: galeria_4 FrameNo: 194 Time 2014-01-16T06:10:15.21+07:00 artinya nama lokasi galeria dengan nama video galeria_4 nomor *frame* 194 direkam pada tanggal 16 Januari 2014 jam 6 lebih 10 menit 15 detik dan 21 milidetik berdasarkan zona waktu GMT + 7. Ada 3 objek yang direpresentasikan dari Gambar 5.12. Nilai ciri objek pertama meliputi: WBB = 156, HBB = 149, area ROI = 23244, *centroid* = [368 383], *aspectratio* = 1, PBB = 610, dan

Dispersedness = 15.154932. Nilai ciri objek meliputi: WBB = 42, HBB = 48, area *ROI* = 2016, *centroid* = [231 189], *aspectratio* = 1, PBB = 180, dan *Dispersedness* = 14.6944035. Nilai ciri objek ketiga meliputi: WBB = 22, HBB = 37, area *ROI* = 814, *centroid* = [166 156], *aspectratio* = 1, PBB = 610, dan *Dispersedness* = 15.233610.

Fields	framenumber	CurrentTime	WBB	HBB	ROI	Centroid	AspectRatio	PBB	Compactness	ColorHOG	GrayHOG	LBP	ColorEnergy	PyramidHOG
1	90	2016-03-05T13:...	39	46	1794	[378 42]	1	170		1x576 double	1x576 double	1x59 double	[3.6469 4.6833]	1x680 double
2	90	2016-03-05T13:...	37	46	1702	[447 54]	1	166		1x432 double	1x432 double	1x59 double	[4.8400 5.7389]	1x680 double
3	90	2016-03-05T13:...	34	70	2380	[553 128]	0	208		1x756 double	1x756 double	1x59 double	[3.5748 4.4787]	1x680 double
4														

Gambar 5.11 Keluaran ciri dalam format XML frame video 1 ke data numerik



Gambar 5.12 Frame Video 2

Gambar 5.13 menunjukkan ciri dalam format XML dari frame video 2. Tag `<?xml version="1.0" encoding="utf-8"?>` menunjukkan versi XML yang digunakan. Tag `<Z.feature_XML="list">` merupakan tanda awal dari pembacaan satu frame. Lokasi frame ditunjukkan dengan `<HFrame Value="LocationNm"> galeria </HFrame>`. Nama video ditunjukkan dengan `<HFrame Value="VideoNm"> galeria_4 </HFrame>`. Nomor frame ditunjukkan dengan

`<HFrame Value = "FrameNo"> 194`, artinya *frame* ke 194 dari video yang direkam. Waktu rekam frame ditunjukkan dengan `<HFrame Value = "FTimes"> 2014-01-16T06:10:15.21+07:00</HFrame>` artinya frame video ini direkam pada tanggal 16 Januari 2014 pada jam 6 lebih 10 menit, 15 detik dan 21 milidetik berdasarkan GMT+7. Dalam satu frame bisa terdapat beberapa objek. Nama objek ditunjukkan dengan `<ObjectNo Value = "Object"> 1 </ObjectNo>`. Jumlah objek ditunjukkan dengan `n`. Nama dan nilai ciri pertama dari objek pertama ditunjukkan dengan `<Feature Value = "WBB">156</Feature>`. Nama dan nilai ciri kedua dari objek pertama ditunjukkan dengan `<Feature Value = "HBB">149</Feature>`. Nama dan nilai ciri ketiga dari objek pertama ditunjukkan dengan `<Feature Value = "area ROI">23244</Feature>`. Nama dan nilai ciri keempat dari objek pertama ditunjukkan dengan `<Feature Value = "centroid">[368 383] </Feature>`. Nama dan nilai ciri kelima dari objek pertama ditunjukkan dengan `<Feature Value = "aspectratio">1</Feature>`. Nama dan nilai ciri keenam dari objek pertama ditunjukkan dengan `<Feature Value = "PBB">610</Feature>`. Nama dan nilai ciri terakhir dari objek pertama ditunjukkan dengan `<Feature Value = "Dispersedness"> 15.154932 </Feature>`. Pembacaan yang sama juga dilakukan untuk objek kedua dan selanjutnya. Semua objek lalu lintas bergerak dibaca cirinya sampai selesai. Terakhir diberi simbol `</Z>` untuk menunjukkan bahwa semua objek telah dibaca dan proses pada satu frame ini selesai.

Gambar 5.14 menunjukkan keluaran ciri dalam format XML ke data numerik. Dari hasil ini dapat dipastikan bahwa ciri yang dikirim sama dengan ciri yang diterima. Dengan demikian, pengujian representasi ciri telah berhasil dan skema representasinya sudah betul karena ciri yang dikirim sama dengan ciri yang diterima. Selanjutnya, skema representasi ciri ini dapat digunakan untuk merepresentasi ciri dari frame video lainnya. Pengujian representasi ciri juga dilakukan dengan menghapus 1 (satu) file XML yang telah diterima. Tujuannya untuk mengecek apakah file XML selanjutnya tetap dapat dibaca dan pengolahan data selanjutnya dapat tetap dilakukan. Hasil pengujian menunjukkan bahwa file XML selanjutnya tetap dapat dibaca dan pengolahan data selanjutnya dapat tetap

dilakukan. Artinya frame selanjutnya tetap dapat dibaca walau ada frame yang hilang. Dengan demikian, pengujian ini membuktikan Arsitektur SVS dapat bekerja dengan baik.

```

<?xml version="1.0" encoding="utf-8"?>
<Z feature_XML="list">
<HFrame Value="LocationNm"> galeria </HFrame>
<HFrame Value="VideoNm"> galeria_4 </HFrame>
  <HFrame Value="FrameNo">194
  <HFrame Value="FTime">2014-01-16T06:10:15.21+07:00
  </HFrame>
    <ObjectNo Value="Object"> 1 </ObjectNo>
      <Feature Value="WBB">156</Feature>
      <Feature Value="HBB">149</Feature>
      <Feature Value="area ROI">23244</Feature>
      <Feature Value="centroid">[368 383] </Feature>
      <Feature Value="aspectratio">1</Feature>
      <Feature Value="PBB">610</Feature>
      :
      <Feature Value="Dispersedness">15.154932</Feature>
      :
    <ObjectNo Value="Object"> 3 </ObjectNo>
      <Feature Value="WBB">22</Feature>
      <Feature Value="HBB">37</Feature>
      <Feature Value="area ROI">814</Feature>
      <Feature Value="centroid">[166 156] </Feature>
      <Feature Value="aspectratio">1</Feature>
      <Feature Value="PBB">118</Feature>
      :
      <Feature Value="Dispersedness">15.233610</Feature>
    </HFrame>
  </Z>

```

Gambar 5.13 Ciri dalam format XML dari frame video 2

Representasi ciri ini telah diuji untuk membuktikan bahwa sistem bekerja dengan baik dan skema yang diusulkan dalam format XML dapat dibaca dengan baik. Hasil pengujian menunjukkan bahwa sistem berhasil mengekstrak ciri, merepresentasikan ciri dalam format XML, mengirim representasi XML ke DCCU. Hasil pengujian juga menunjukkan bahwa representasi XML berhasil diterima di DCCU dan dibaca dengan baik oleh sistem lain. Pengujian ini dapat berhasil dengan baik karena telah ditambahkan algoritme untuk membuat XML

dapat membaca file/frame selanjutnya. Algoritme A.11 pada Lampiran A menunjukkan cara XML dapat membaca file/frame selanjutnya.

Fields	framenumber	CurrentTime	WBB	HBB	ROI	Centroid	AspectRatio	PBB	Compactness	ColorHOG	GrayHOG	LBP	ColorEnergy	PyramidHOG
1	194	2016-03-05T12:...	156	149	23244	[368 383]	1	610	1	1x11016 double	1x11016 dou...	1x59 double	[1.3581 1.5228]	1x680 double
2	194	2016-03-05T12:...	42	48	2016	[231 189]	1	180	1	1x720 double	1x720 double	1x59 double	[3.2352 3.9802]	1x680 double
3	194	2016-03-05T12:...	22	37	814	[166 156]	1	118	1	1x108 double	1x108 double	1x59 double	[4.1399 4.4130]	1x680 double
4														

Gambar 5.14 Keluaran ciri dalam format XML ke data numerik

Representasi ciri-ciri dari objek lalu lintas sangat berguna bagi pihak kepolisian dan Dinas Perhubungan. Dengan ciri ini, pihak kepolisian dan Dinas Perhubungan langsung dapat menggunakan ciri untuk kebutuhan yang lebih spesifik seperti: mengidentifikasi objek, mendeteksi objek, pengenalan objek, klasifikasi objek, pelacakan objek, perhitungan jumlah objek, perhitungan kecepatan objek, estimasi kemacetan lalu lintas, dan *traffic parameter* lalu lintas lainnya.

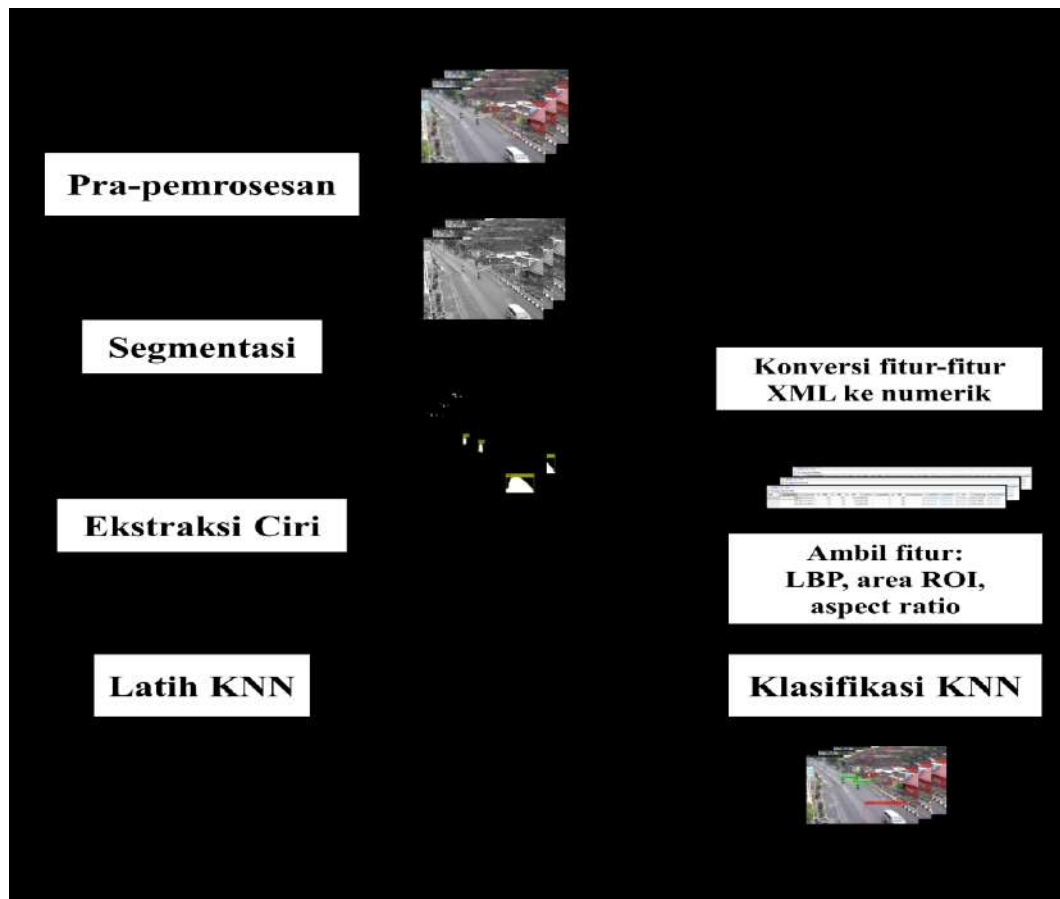
5.2.2 Hasil pengujian pada klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS

Pengujian klasifikasi kendaraan bertujuan untuk mengetahui seberapa baik ciri objek lalu lintas yang di rekonstruksi dari ciri dalam format XML. Parameter keberhasilan diukur dengan seberapa baik ciri dapat dipakai agar objek dapat dikelompokkan sesuai kelasnya. Kendaraan akan diklasifikasi menggunakan ciri objek yang sudah diperoleh dari SVS. Dengan adanya SVS, maka hasil klasifikasi kendaraan dapat lebih cepat diperoleh. Gambar 5.15 menunjukkan model klasifikasi kendaraan yang mengikuti Arsitektur SVS. Ciri yang dipilih untuk klasifikasi kendaraan adalah: *Local Binary Pattern (LBP)*, area *Region of Interest (ROI)* dan *aspect ratio*. Alasan memilih 3 ciri untuk klasifikasi kendaraan karena ketiga ciri ini cukup untuk menentukan kelas kendaraan. Algoritme yang

digunakan untuk mengelompokkan kendaraan adalah: *K-Nearest Neighbor (KNN)*. Alat ukur yang digunakan untuk mengukur kinerja sistem pengklasifikasi adalah *confusion matrix*.

5.2.2.1 Training (Pelatihan)

Awalnya, data *training* (pelatihan) dipersiapkan. Pada pelatihan dilakukan pra-pemrosesan, segmentasi dan ekstraksi ciri. Ciri yang diekstrak adalah *Local Binary Pattern (LBP)*, area *Region of Interest (ROI)*, dan *aspect ratio*. Kemudian data kendaraan dikumpulkan berdasarkan jenisnya (dataset). Kendaraan dibagi menjadi 4 klas yaitu motor, minibus, bus dan truk sesuai (Kemenhub, 2014). Gambar A.12 dan Gambar A.13 pada Lampiran A menunjukkan pembuatan dan penyatuan *dataset*.



Gambar 5.15 Model klasifikasi kendaraan berdasarkan Arsitektur SVS

5.2.2.1.1 Data video

Format awal data video berbentuk .3GP. Sebelum diproses, data ini dikonversi menjadi .AVI. Konversi ini dilakukan karena data format ini yang dikenali oleh Matlab. Pada proses awal pelatihan, data video diekstrak menjadi frame-frame.

5.2.2.1.2 Pra pemrosesan

Tiap frame dilakukan pra pemrosesan menggunakan *Histogram Equalization* untuk meningkatkan kualitas frame-nya. Metode *Histogram Equalization* dipilih karena keunggulannya yang dapat meningkatkan kualitas citra secara menyeluruh (Hawlater dan Hossain, 2014; Xu dkk., 2014; Diego, 2014; Yadav dkk., 2014; Kadir dan Susanto, 2013).

5.2.2.1.3 Segmentasi

Ada dua hal yang dilakukan pada proses segmentasi yaitu memisahkan latar objek dengan objeknya dan memisahkan antara objek yang satu dengan objek yang lain. Metode segmentasi yang digunakan adalah *Gaussian Mixture Model* (GMM). Data masukan segmentasi adalah frame yang sudah ditingkatkan kualitasnya. Hasil segmentasi adalah segmen-segmen. Segmen-segmen ini terpisah di beberapa tempat. Tiap segmen diberi label. Tiap label dideteksi sebagai objek. Pelabelan diberikan untuk ukuran ≥ 200 . Pemilihan ukuran ini untuk meminimalkan kesalahan objek yang terdeteksi. Segmentasi juga menggunakan morfologi *opening* dan *closing*. Morfologi *opening* bertujuan untuk menebalkan garis-garis bentuk objek. Morfologi *closing* bertujuan untuk menghilangkan lubang kecil dan mengisi gap pada garis-garis bentuk (*contour*). Morfologi *opening* dan *closing* digunakan untuk menghilangkan *noise* dari segmen-segmen objek.

5.2.2.1.4 Ekstraksi ciri

Proses ekstraksi ciri berguna untuk mengekstrak nilai-nilai unik suatu objek yang membedakan dengan objek yang lain. Tiap label yang telah dideteksi sebagai objek kemudian diekstrak cirinya. Ciri diekstrak untuk mendapatkan nilai ciri yang unik. Ciri yang diekstrak ada 3 yaitu *Local Binary Pattern (LBP)*, area

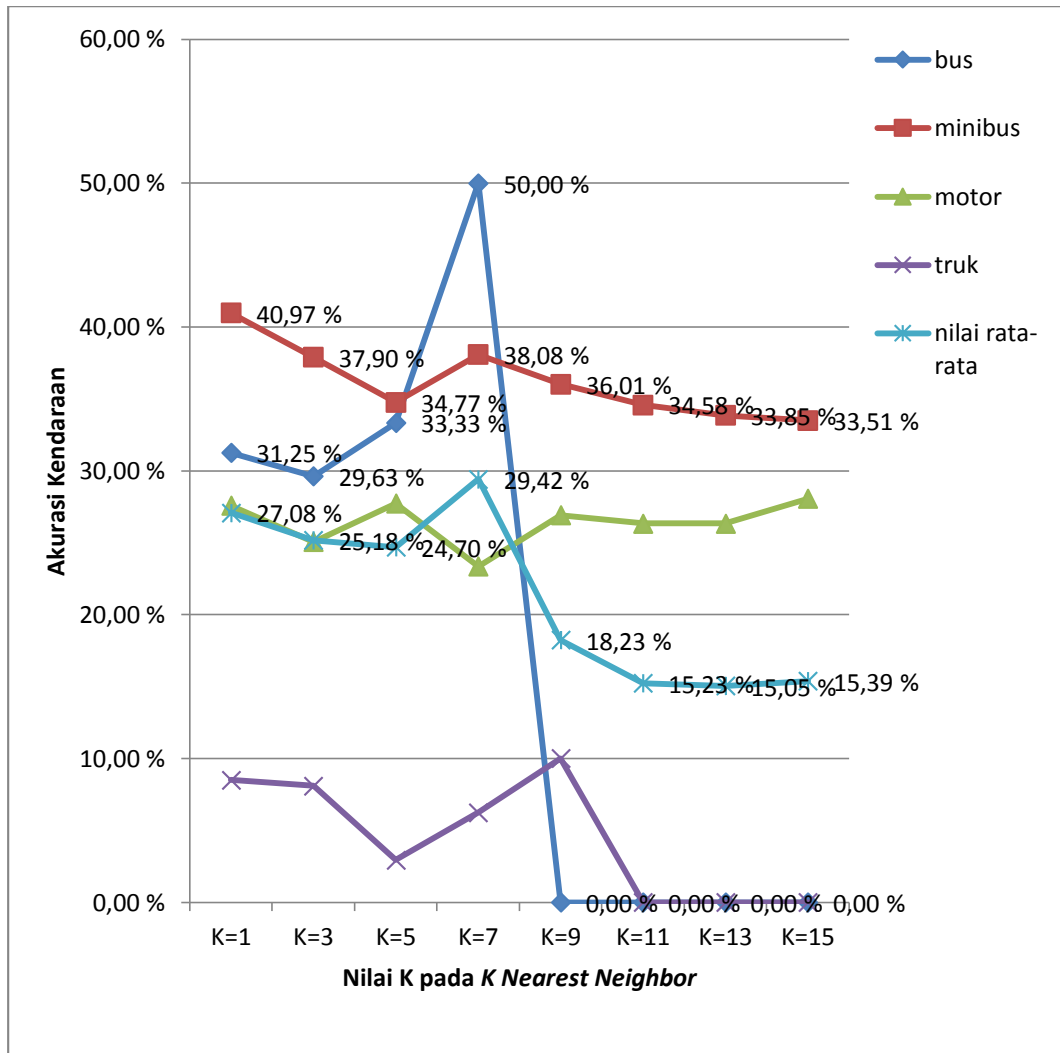
ROI, dan Aspect ratio. Ciri *Local Binary Pattern* (LBP) dipilih karena memiliki 59 dimensi sehingga memberikan klasifikasi menjadi lebih baik. Nilai ciri sebanyak 59 dimensi pada LBP ini digunakan karena dapat membedakan secara unik antara objek yang satu dengan yang lainnya. Ciri area ROI dipilih karena area dapat digunakan untuk membedakan kelas kendaraan. Ciri *aspect ratio* dipilih karena perbedaan nilai rasio dapat membedakan kelas kendaraan. Jenis kendaraan ada 4: motor, minibus, bus, truk. Alasan pemilihan 4 kelas kendaraan ini karena mengikuti peraturan pemerintah no. 55 tahun 2012 tentang kendaraan (Kemenhub, 2012) dan akurasinya bagus menurut peneliti (Matos dan Souza, 2012). Data set juga disusun berdasarkan lokasinya.

5.2.2.1.5 K-Nearest Neighbor

Metode yang digunakan untuk mendapatkan kelas kendaraan adalah *K-Nearest Neighbor* (KNN). Algoritme *K-Nearest Neighbor* (KNN) memprediksi kategori *instance* baru berdasarkan informasi dari *instance-instance* terdekat dengannya atau tetangga terdekatnya. Ciri yang digunakan untuk mencari kelas kendaraan adalah *Local Binary Pattern* (LBP), *area Region of Interest* (ROI) dan *Aspect Ratio*. Nilai LBP ada 59 dimensi, nilai area ROI ada 1 dan nilai aspect ratio ada 1. Metode KNN dipilih karena akurasinya diatas 90% dan klasifikasi kendaraannya juga menggunakan 4 kelas kendaraan (Matos dan Souza, 2012). KNN adalah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pelatihan yang jaraknya paling dekat dengan objek tersebut (Matos dan Souza, 2012). Kelas kendaraannya adalah motor, mobil, bus, truk karena mengikuti peraturan pemerintah no. 55 tahun 2012 tentang kendaraan (Kemenhub, 2012) dan akurasinya bagus menurut peneliti lain (Matos dan Souza, 2012). Untuk mencari nilai K terbaik dilakukan *KNN train*. Tujuan penentuan nilai K terbaik agar diperoleh hasil akurasi yang terbaik.

Gambar 5.16, Gambar 5.17 dan Gambar 5.18 menunjukkan percobaan dengan beberapa variasi jumlah ketetanggaan KNN. Nama-nama video yang dipilih untuk *KNN train* adalah babarsari_8, galeria_8, condongcatur_11. Alasan pemilihan video ini karena :

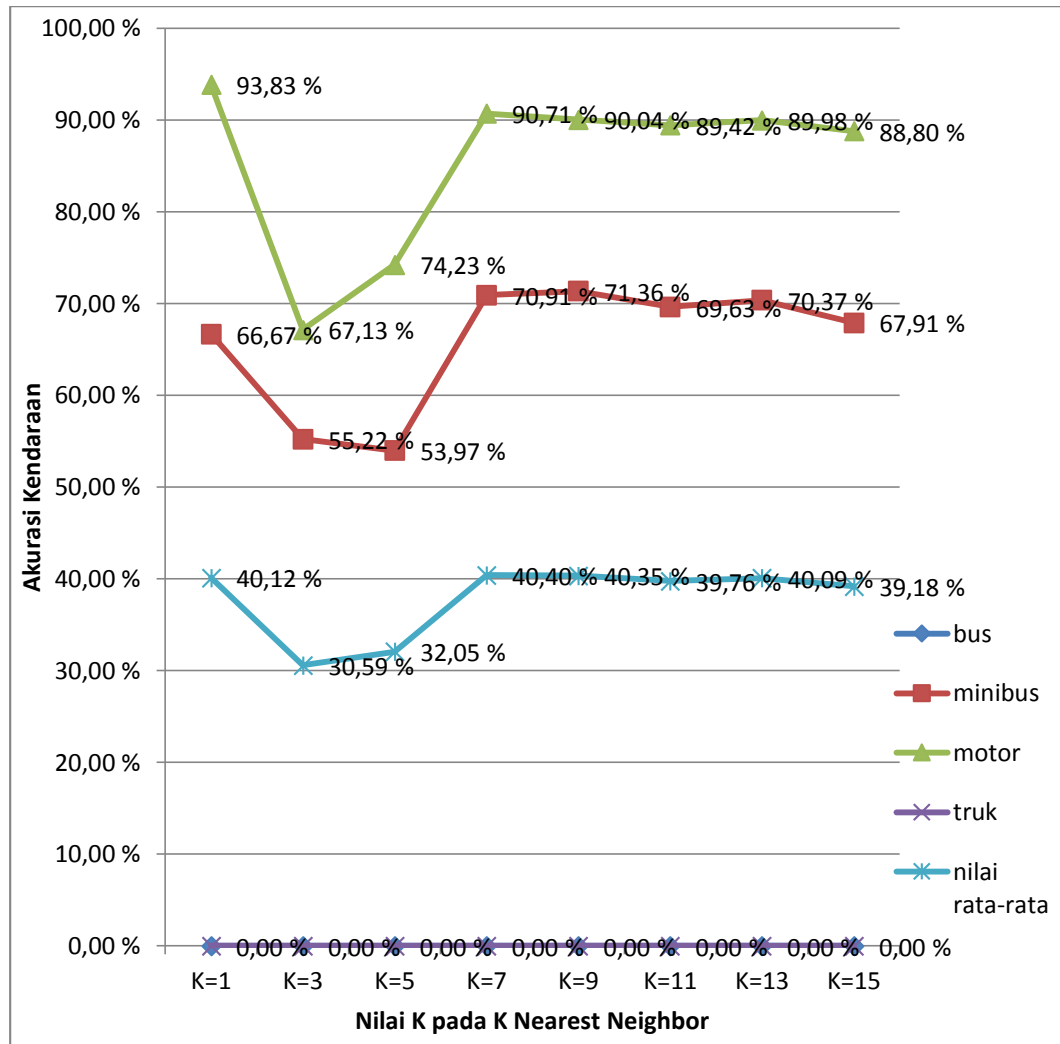
1. kendaraan-kendaraannya hanya kendaraan yang menghadap depan dan belakang di 3 lokasi persimpangan Yogyakarta.
2. jumlah variasi jenis kendaraannya sama dengan jumlah klas kendaraan yang digunakan pada penelitian ini kecuali Galeria_8 karena bus dan truk tidak melintas di Galeria.



Gambar 5.16 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Babarsari

Frame rate video ini adalah 30 fps. Jumlah frame video-video dari babarsari_8, galeria_8, condongcatur_11 berturut-turut adalah 1814 frame, 1000 frame, dan 1000 frame. Nilai rata-rata tertinggi dari video babarsari_8, galeria_8, condongcatur_11 berturut-turut adalah 29,42, 80,81, dan 24,95 dengan K=7. Dari

beberapa eksperimen itu maka dapat disimpulkan bahwa nilai K terbaik diperoleh pada $K = 7$. Setelah itu, KNN model menyimpan data training dengan $K=7$ dan alat ukur jarak Euclidean distance. Parameter KNN Model adalah nilai K dan alat ukur jarak yang digunakan. Alasan pemilihan Euclidean distance karena sederhana (Shi dkk., 2013).



Gambar 5.17 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Galeria

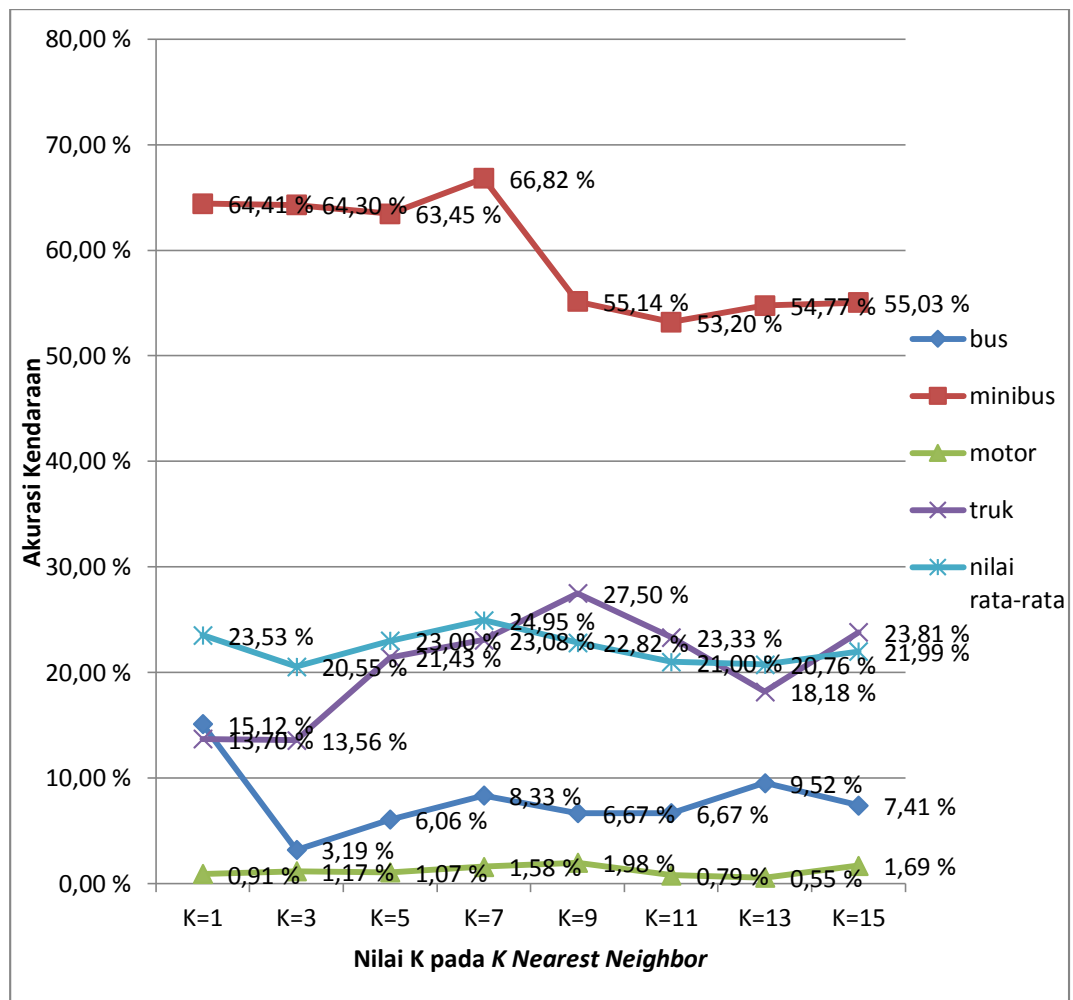
5.2.2.2 Testing (Pengujian)

Untuk data pengujian, sistem membaca ciri dalam format XML dari SVS lalu sistem mengkonversi ciri dalam format XML ke numerik sehingga berbentuk tabel data numerik. Kemudian, sistem mengambil ciri LBP, area ROI dan *aspect*

ratio. Ketiga fitur ini digunakan untuk klasifikasi menggunakan KNN untuk mendapatkan kelas kendaraan. Parameter K yang digunakan adalah 7. Contoh hasil keluaran kelas kendaraan ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Hasil keluaran kelas kendaraan

No	Nama Kelas	Nama lokasi	Nama video	Nomor frame	Waktu frame (<i>time frame</i>)	Koordinat	
						X	Y
1.	Motor	Babarsari	Babarsari_4	190	2017-11-07T12:51:32.110+0700	212	125



Gambar 5.18 Hubungan antara nilai K pada KNN dengan akurasi kendaraan di Condong Catur

Tabel 5.3 menunjukkan akurasi klasifikasi kendaraan berdasarkan Arsitektur SVS. Pengujian dilakukan pada lokasi Babarsari, Galeria dan Condong Catur. Jumlah frame yang digunakan untuk menguji 1000 frame, akurasi kendaraan dilihat per kelas kendaraan.

Tabel 5.3 Akurasi klasifikasi kendaraan berdasarkan Arsitektur SVS

Lokasi	bus	minibus	sepeda motor	truk
Babarsari_8	-	17,51	37,33	-
Galeria_8	-	70,91	90,71	-
Condong catur_11	6,25	65,73	7,76	2,44

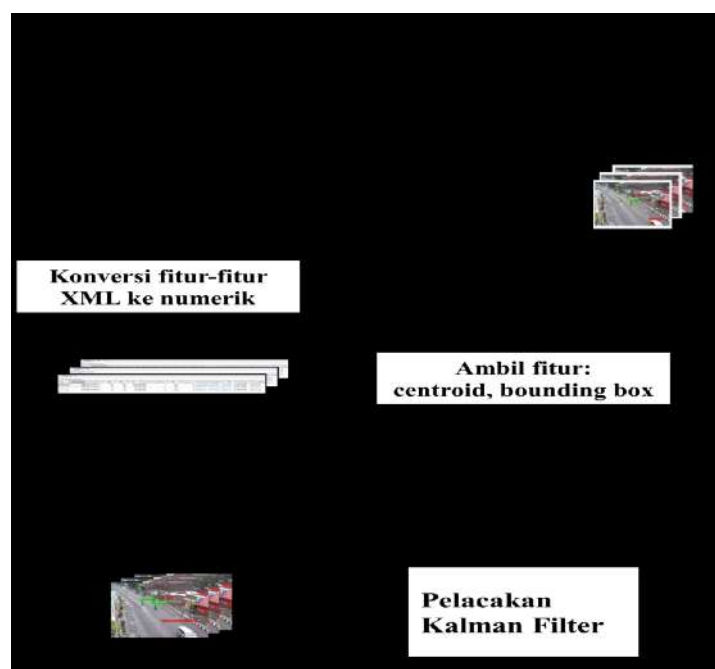
Dari hasil pengujian klasifikasi kendaraan yang ciri-ciri objeknya berasal dari SVS pada Tabel 5.3 menunjukkan keakuratan tertinggi ditemukan pada sepeda motor adalah di Galeria (90,71%), untuk mobil di Galeria (70,91%), untuk bus di Condong Catur (6,25%), dan untuk truk di Condong Catur (2,44%). Untuk penelitian ke depan, penggunaan KNN perlu dihindari karena prosesnya yang lama untuk *real-time processing*.

5.2.3 Hasil pengujian pada pelacakan kendaraan berdasarkan jenis kendaraan yang ciri-ciri objeknya berasal dari SVS

Pelacakan kendaraan berdasarkan jenisnya dibuat sebagai pembuktian penyelesaian frame yang hilang. Pengujian ini bertujuan untuk membuktikan ciri objek lalu lintas yang telah dikirim ke DCCU dapat digunakan dengan baik. Pelacakan ini juga diperlukan untuk mengetahui informasi mengenai kondisi lalu lintas terkini. Alasan pengujian SVS dengan membuat pelacakan kendaraan berdasarkan jenisnya adalah karena pihak kepolisian dan Dinas Perhubungan membutuhkan informasi mengenai apa dan siapa yang menjadi penyebab kecelakaan.

Gambar 5.19 menunjukkan model pelacakan kendaraan berdasarkan jenisnya yang mengikuti Arsitektur *Smart Video Sensor*. Pelacakan dengan metode *Kalman Filter* dilakukan setelah kelas kendaraan terpilih. Metode *Kalman Filter* dipilih karena metode ini dapat memprediksi lokasi terbaik untuk kendaraan dalam frame video berikutnya (Liu dan Yang, 2010; Purwanto dkk., 2013; Li dkk., 2014; Teoh dan Bräunl, 2012). Ciri yang digunakan untuk pelacakan adalah fitur *centroid* dan *bounding box*. Proses pelacakan ini, pertama, *state* awal yakni nilai *centroid* dan *bounding box*. Kedua, *state* matrik awal diubah menjadi *state* matrik sebelumnya. Ketiga, *state* matriks sebelumnya diprediksi *state* matrik berikutnya. Keempat, *state* diperbarui menggunakan pengukuran dan *Kalman Gain* untuk

memastikan kebenaran nilai state berikutnya. Pengukuran ini berasal dari nilai *centroid state* matriks saat ini. Hasilnya diperoleh *state* matriks terkini. Kelima, *state* matriks terkini diubah menjadi *state* matriks sebelumnya. Keenam, *state* matriks terkini dikurangi dengan *state* matriks sebelumnya untuk membentuk *path* kendaraan. Satu siklus pelacakan selesai ditandai dengan terbentuknya *path* kendaraan. Siklus ini diulangi pada proses pelacakan sampai ciri dalam format XML selesai dibaca.



Gambar 5.19 Pelacakan Kendaraan sesuai jenisnya yang ciri-ciri objeknya berasal dari SVS

Metode *Kalman Filter* dipilih karena metode ini dapat memprediksi lokasi terbaik untuk kendaraan dalam frame video berikutnya (Liu dan Yang, 2010; Purwanto dkk., 2013; Li dkk., 2014; Teoh dan Bräunl, 2012). Ciri yang digunakan untuk pelacakan adalah *centroid*. *Centroid* adalah titik tengah dari objek. Nilai *centroid* berisi koordinat x dan y yang merupakan titik tengah objek. Perhitungan *Kalman Filter* dijelaskan secara detil pada Lampiran 4 karena ini penting untuk memastikan kebenaran nilai *centroid* yang digunakan.

Akurasi kendaraan dihitung dari banyaknya kendaraan yang terlacak benar (Salarpour dkk., 2011) seperti yang ditunjukkan pada persamaan (4.1). Gambar 5.20 menunjukkan hasil pengujian pelacakan kendaraan berdasarkan Arsitektur

SVS di Yogyakarta. Pada Gambar 5.20a terdapat 2 motor yang terlacak di Babarsari. Pada Gambar 5.20b terdapat 2 motor dan 1 mobil yang terlacak di Galeria. Pada Gambar 5.20c terdapat 1 motor, 1 mobil dan 1 bus yang terlacak di Condong Catur.



Gambar 5.20 Hasil pengujian pelacakan kendaraan berdasarkan Arsitektur SVS di Yogyakarta (a) Babar Sari, (b) Galeria, (c) Condong Catur

Validasi pelacakan kendaraan berdasarkan *Kalman Filter* perlu dilakukan untuk mengetahui perbedaan nilai *centroid* yang berasal dari ciri dalam format XML dengan nilai *centroid* yang dihasilkan oleh *Kalman Filter*. Cara menghitung perbedaannya menggunakan *average difference*, dimana alat ukur jarak yang digunakan adalah *Euclidean distance*. Alasan pemilihan *Euclidean distance* adalah karena sederhana dalam mencari jarak antara dua buah titik (Shi dkk., 2013).

Sampel data yang digunakan untuk validasi pelacakan kendaraan berdasarkan *Kalman Filter* yang mengikuti Arsitektur SVS berasal dari lokasi Galeria, dengan nama video Galeria_8. Tabel 5.4 dan Tabel 5.5 menunjukkan ilustrasi objek antara *centroid* dari *Kalman Filter* dan ciri dalam format XML. Pada frame ke 53 di deteksi ada 1 objek. Awalnya nilai *centroid* x dan nilai *centroid* y dari ciri dalam format XML menjadi masukan dari nilai *centroid* KFx dan nilai *centroid* KFy dari *Kalman Filter*. Kedua tabel itu menunjukkan besar *average difference* antara titik *centroid* dari *Kalman Filter* dan ciri dalam format XML. Total *average difference* dari sampel data yang diwakili oleh dua tabel tersebut yaitu 0,054179115. Artinya rata-rata perbedaan nilai *centroid* Kalman Filter dengan nilai *centroid* dari ciri dalam format XML adalah 0,05. Satuan dari nilai 0,05 adalah piksel. Hasil ini menunjukkan terjadi

pergeseran nilai *centroid Kalman Filter* sebesar 0,05 dibanding dengan nilai *centroid* yang berasal dari ciri dalam format XML.

Tabel 5.4 Ilustrasi 1 objek antara *centroid* dari *Kalman Filter* dan ciri dalam format XML

No.	frameno	objectno	centroidx	centroidy	centroidKFx	centroidKFy	distance
1	53	1	233	167	233	167	0
2	54	1	234	167	233,9722	167	0,0278
3	55	1	234	167	234,0058	167	0,0058
4	56	2	236	171	235,9025	170,7944	0,22754694
5	57	2	236	172	236,036	172,0136	0,03848324
6	58	2	236	173	236,0299	173,02	0,03597235
7	59	2	237	173	236,9669	173,0651	0,07303164
8	60	1	237	173	237,0275	173,0476	0,05497281
9	61	2	238	175	237,9688	174,9231	0,08298825
10	62	2	239	173	238,977	173,1524	0,15412579
11	63	2	240	173	239,9857	173,0054	0,01528561
12	64	2	241	174	240,9913	173,9439	0,05677059
13	65	1	241	177	241,0467	176,857	0,15043234
14	66	2	242	177	241,9808	177,0599	0,06290191
						average =	0,07043653

Tabel 5.5. Ilustrasi objek lain antara *centroid* dari *Kalman Filter* dan ciri dalam format XML

No.	frameno	objectno	centroidx	centroidy	centroidKFx	centroidKFy	distance
1	56	1	181	187	181	187	0
2	57	1	182	187	181,9722	187	0,0278
3	58	1	182	188	182,0058	187,9475	0,0528194
4	59	1	183	189	182,9539	188,9646	0,0581237
5	61	1	184	190	183,9964	190,0038	0,0052345
6	62	1	183	191	183,0769	190,9771	0,0802373
7	63	1	184	192	183,9496	191,9841	0,0528486
8	64	1	184	192	184,0131	192,0421	0,044091
9	66	1	185	194	184,9892	193,983	0,0201405
						average =	0,0379217

BAB 6

PENUTUP

6.1 Kesimpulan

Berdasarkan tujuan beserta hasil yang telah diperoleh dari penelitian ini, dapat disimpulkan bahwa:

- a. Permasalahan frame video yang hilang telah diatasi dengan memodifikasi konsep Arsitektur Smart Video Sensor (SVS) yang fokusnya membuat Modul Ekstraksi Ciri dan Modul Representasi Ciri dengan mengekstrak banyak ciri objek lalu lintas yang ciri-cirinya diperlukan untuk pengontrolan lampu lalu lintas, pelacakan kendaraan, mengatasi kepadatan lalu lintas dan parameter lalu lintas lainnya yang langsung dapat digunakan untuk menganalisa data lalu lintas selanjutnya. SVS mengekstrak ciri tiap objek menggunakan *time stamp* yang menyimpan informasi waktu terkini. *Time stamp* tersimpan dalam setiap frame dalam format XML. *Time stamp* disimpan untuk menyelesaikan masalah data video lalu lintas yang tidak lengkap. Dengan *time stamp*, frame terkini dapat terhubung dengan frame berikutnya. Hasilnya akan mempermudah untuk melacak objek. Dengan demikian permasalahan frame video yang hilang telah teratasi.
- b. Skema representasi telah dihasilkan untuk mempresentasikan ciri pada Modul Representasi Ciri. Hasil eksperimen yang telah dilakukan menunjukkan bahwa representasi ciri yang dihasilkan telah sesuai dengan keadaan videonya. Hasil ini menunjukkan skema representasi ciri yang dihasilkan sudah betul. Hal ini ditandai dengan ciri yang dikirim sama dengan yang diterima dan ciri dapat dibaca dengan baik.
- c. Hasil eksperimen yang telah dilakukan pada representasi ciri menunjukkan bahwa ciri dalam format XML yang telah dikirim ke DCCU dapat dibaca dengan benar.
- d. Klasifikasi kendaraan dengan metode *K-Nearest Neighbor* dan menggunakan ciri-ciri objek lalu lintas yang berasal dari *Smart Video Sensor (SVS)* dapat bekerja dengan baik. Hasil akurasi kelas kendaraan tertinggi ditemukan pada sepeda motor adalah di Galeria (90,71%), untuk mobil di Galeria (70,91%),

untuk bus di Condong Catur (6,25%), dan untuk truk di Condong Catur (2,44%).

- e. Pelacakan kendaraan sesuai jenisnya dengan metode *Kalman Filter* dan menggunakan ciri-ciri objek lalu lintas yang berasal dari *Smart Video Sensor (SVS)* dapat bekerja dengan baik. Terjadi perbedaan nilai *centroid* Kalman Filter sebesar 0,055 piksel dibanding dengan nilai *centroid* yang berasal dari ciri dalam format XML.

6.2 Saran

Penelitian ini masih perlu pengembangan lebih lanjut. Saran yang dapat diberikan untuk penelitian selanjutnya adalah :

- a. Melakukan seleksi ciri dari ciri dalam format XML yang diterima oleh DCCU agar ciri-ciri yang digunakan untuk merekonstruksi kendaraan berdasarkan jenisnya dapat lebih berkualitas dan meningkatkan nilai akurasi.
- b. Menghitung kecepatan kendaraan, jumlah kendaraan dan kepadatan kendaraan menggunakan ciri-ciri dalam format XML.
- c. Ekstraksi ciri tidak lagi dilakukan pada DCCU, tetapi DCCU cukup memanfaatkannya saja, sehingga proses pelatihan pada Gambar 5.15 tidak perlu lagi mengekstrak ciri.
- d. Perlu ada koneksi antara LPU dan DCCU agar *monitoring* pada DCCU lebih maksimal.
- e. Untuk penelitian ke depan, penggunaan KNN perlu dihindari karena prosesnya yang lama untuk *real-time processing*.

DAFTAR PUSTAKA

- Abouelatta, O.B., 2013. Classification of Copper Alloys Microstructure using Image Processing and Neural Network. *Journal of American Science*, 9(6), pp.213–223. Available at: http://www.jofamericanscience.org/journals/am-sci/am0906/025_18156am0906_213_223.pdf.
- Adhikari, S.P., Yoo, H.J. dan Kim, H., 2011. Real time on-road vehicle detection with low-level visual features and boosted cascade of haar-like features. *Journal of Institute of Control, Robotics and Systems*, 17(1), pp.17–21.
- Ambardekar, A.A., 2007. *Efficient Vehicle Tracking and Classification for an Automated Traffic Surveillance System*. University of Nevada, Reno.
- Anderson, D.T., Sjahputera, O., Stone, K. dan Keller, J.M., 2012. Causal cueing system for above ground anomaly detection of explosive hazards using support vector machine localized by K-nearest neighbor. *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications, CISDA 2012*.
- Arróspide, J., Salgado, L., Marinas, J., Imágenes, G.D.T. De dan Telecomunicación, E.T.S.I., 2012. HOG-like Gradient-based Descriptor for Visual Vehicle Detection. Dalam *2012 Intelligent Vehicle Symposium*. pp. 223–228.
- Basri, B., Bayu, I. dan Achmad, A., 2015. Gaussian Mixture Models Optimization for Counting The Numbers of Vehicle by Adjusting The Region of Interest under Heavy Traffic Condition. Dalam *2015 International Seminar on Intelligent Technology and Its Applications*. pp. 245–249.
- Bayer, F. dan Hauser, J., 2012. Trajectory optimization for vehicles in a constrained environment. Dalam *Proceedings of the IEEE Conference on Decision and Control*. pp. 5625–5630.
- Belbachir, A.N., 2010. *Smart Cameras* A. N. Belbachir, ed., London, UK: Springer Science + Business Media.
- BINKOT, 1997. Indonesian Highway Capacity Manual and Software (IHCM). ,

7802112(264).

- Bishop, G. dan Welch, G., 2001. An introduction to the Kalman Filter. Dalam *SIGGRAPH 2001: Course 8*. pp. 1–16. Available at: http://old.shahed.ac.ir/references/kalman_filter_notes.pdf.
- Board, T.R., 2000. *Highway Capacity Manual*, United States of America: National Research Council.
- Bosch, A., Zisserman, A. dan Munoz, X., 2007. Representing shape with a spatial pyramid kernel. Dalam *Proceedings of the 6th ACM international conference on Image and video retrieval - CIVR '07*. pp. 401–408. Available at: <http://dl.acm.org/citation.cfm?doid=1282280.1282340>.
- Boyle, J. dan Ferryman, J., 2015. Vehicle subtype, make and model classification from side profile video. Dalam *12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2015)*. pp. 1–7.
- Bravo, C., Sánchez, N., García, N. dan Menéndez, J.M., 2013. Outdoor vacant parking space detector for improving mobility in smart cities. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8154 LNAI, pp.30–41.
- Bray, T., Paoli, J., Initiative, T.E., Maler, E. dan Microsystems, S., 2000. *Extensible Markup Language (XML) 1.0 (Second Edition)*, Keio: Sun Microsystems, Inc.
- Bray, T., Paoli, J., Sperberg-McQueen, C.M. dan Maler, E., 2013. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, Keio.
- Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. dan Cowan, F.Y.J., 2006. *Extensible Markup Language (XML) 1.1 (Second Edition)*, Keio.
- Bubenikova, E., Pirnik, R. dan Holecko, P., 2013. Optimisation of Video-Data Transmission in Telematic System Keywords. , pp.123–134.
- Burkul, S.R., Pawar, P.R. dan Jagtap, K.R., 2014. Estimation of Vehicle Parameters using Kalman Filter : Review. *International Journal of Current Engineering and Technology*, 4(4), pp.2731–2735.
- Burrus, C.S., Gopinath, R. a dan Guo, H., 1998. *Introduction to Wavelets and Wavelet Transforms: A Primer*, Available at:

<http://www.amazon.com/Introduction-Wavelets-Wavelet-Transforms-Primer/dp/0134896009>.

- C, R. dan Kumar, T.R.S., 2015. Vehicle Detection and Counting. *International Journal of Electrical and Electronics Engineers*, 07(01), pp.160–166. Available at: <https://www.behance.net/gallery/4057777/Vehicle-Detection-Tracking-and-Counting>.
- Carey, P., 2010. *New Perspectives HTML, XHTML and XML* 3rd ed., United States of America: Cengage Learning, Inc.
- Carey, P., 2007. *New Perspectives on XML* 2nd ed., United States of America: Carey Associates, Inc.
- Chadha, A. dan Carolina, N., 2012. Comparative Study and Optimization of Feature- Extraction Techniques for Content based Image Retrieval. , 52(20), pp.35–42.
- Chan, A.B. dan Vasconcelos, N., 2012. Counting people with low-level features and Bayesian regression. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 21(4), pp.2160–77. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/22020684>.
- Chan, Y.-M., Huang, S.-S., Fu, L.-C., Hsiao, P.-Y. dan Lo, M.-F., 2012. Vehicle detection and tracking under various lighting conditions using a particle filter. *IET Intelligent Transport Systems*, 6(1), p.1. Available at: <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2011.0019>.
- Chang, S., He, J., Jiang, Y., Khoury, E. El, Ngo, C., Yanagawa, A. dan Zavesky, E., 2008. Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search.
- Chen, Y. dan Park, P.S., 2013. Object Tracking Based on Online Classification Boosted by Discriminative Features. *International Journal of Energy, Information and Communication*, 4(6), pp.9–20. Available at: <http://dx.doi.org/10.14257/ijeic.2013.4.6.02>.
- Chen, Z. dan Ellis, T., 2011. Multi-shape Descriptor Vehicle Classification for Urban Traffic. *2011 International Conference on Digital Image Computing: Techniques and Applications*, pp.456–461. Available at:

<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6128760>
[Diakses Mei 17, 2013].

- Chen, Z., Ellis, T. dan Smieeee, S.A.V., 2012. Vehicle Detection, Tracking and Classification in Urban Traffic. Dalam *2012 15th International IEEE Conference on Intelligent Transportation Systems*. pp. 951–956.
- Chen, Z., Ellis, T. dan Velastin, S.A., 2011. Vehicle Type Categorization: A Comparison of Classification Schemes. Dalam *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. pp. 74–79.
- Chen, Z., Pears, N., Freeman, M. dan Austin, J., 2009. Road Vehicle Classification using Support Vector Machines. *2009. ICIS 2009. IEEE International Conference on Intelligent Computing and Intelligent Systems*, 4, pp.214–218.
- Choras, R.S., 2007a. Feature extraction for CBIR and Biometrics applications.
- Choras, R.S., 2007b. Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems. , 1(1).
- Chun-Hong, C. dan Zhi-Lei, S., 2016. Improved LBP Vehicles Texture Features Extraction Algorithm in Video Monitor Systems. *2016 International Conference on Virtual Reality and Visualization (ICVRV)*, pp.87–93. Available at: <http://ieeexplore.ieee.org/document/7938177/>.
- Daigavane, P.M., Bajaj, P.R. dan Daigavane, M.B., 2011. Vehicle Detection and Neural Network Application for Vehicle Classification. *2011 International Conference on Computational Intelligence and Communication Networks*, pp.758–762. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6112973>
[Diakses Mei 17, 2013].
- Dalal, N. dan Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. Dalam *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Daya, B., Akoum, A.H. dan Chauvet, P., 2010. Neural Network Approach for the Identification System of the Type of Vehicle. *2010 International Conference on Computational Intelligence and Communication Networks*, pp.162–166.

- Available at:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5701956>
 [Diakses Mei 17, 2013].
- Deng, T.M. dan Li, B., 2012. A Detection Method of Traffic Parameters Based on EPI. *2012 International Workshop on Information and Electronics Engineering (IWIEE)*, 29, pp.3054–3059.
- Dharani, T. dan Aroquiaraj, I.L., 2013. A Survey on Content Based Image Retrieval. *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, pp.485–490.
- Diego, S., 2014. Cross-Entropy Histogram Equalization. , pp.158–163.
- Dutta, S. dan Shen, H.W., 2016. Distribution Driven Extraction and Tracking of Features for Time-varying Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), pp.837–846.
- ETSI, 2011. *3GPP TS 26.247 v.10.0.0*, Valbonne, France.
- Evans, M., Osborne, C.J. dan Ferryman, J., 2013. Multicamera object detection and tracking with object size estimation. *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp.177–182.
 Available at:
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6636636>.
- Fawcett, J., Quin, L.R.E. dan Ayers, D., 2012. *Beginning XML Fifth.*, Indiana: John Wiley & Sons, Inc.
- Fazli, S., Mohammadi, S. dan Rahmani, M., 2012. Neural Network based Vehicle Classification for Intelligent Traffic Control. *International Journal of Software Engineering & Applications (IJSEA)*, 3(3), pp.17–22.
- Firdaus, A., Purnama, B. dan Novia Wisesty, U., 2016. Klasifikasi Kendaraan di Jalan Tol dengan Menerapkan Metode Local Binary Pattern dan Linear Discriminant Analysis. *Indosc 2016*, (August), pp.285–296. Available at:
<http://socj.telkomuniversity.ac.id/ocs/index.php/indosc/indosc2016/paper/view/158>.
- Gandhi, T. dan Trivedi, M.M., 2007. Video Based Surround Vehicle Detection, Classification and Logging from Moving Platforms: Issues and Approaches.

- 2007 *IEEE Intelligent Vehicles Symposium*, pp.1067–1071. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4290258>.
- Gao, S., Zhong, Y. dan Li, W., 2010. Robust Adaptive Filtering Method for SINS / SAR Integrated Navigation System. *Aerospace Science and Technology*, 1, pp.1–6. Available at: <http://dx.doi.org/10.1016/j.ast.2010.08.007>.
- Gibson, J.D., 2000. *Handbook of Image & Video Processing* A. Bovik, ed., Canada: Academic Press.
- Glotin, H., Zhao, Z. dan Dumont, E., 2009. LSIS TREC VIDEO 2009 High Level Feature Retrieval using Compact Profile Entropy Descriptors. *Entropy*, 2008.
- Gonzalez, R.C. dan Woods, R.E., 2012. *Digital Image Processing Third Edition*.
- Gonzalez, R.C., Woods, R.E. dan Eddins, S.L., 2009. *Digital image processing using MATLAB* Second Edi.,
- Gonzalez, R.C., Woods, R.E. dan Masters, B.R., 2008. *Digital Image Processing, Third Edition*, Prentice-Hall, Inc. Available at: <http://biomedicaloptics.spiedigitallibrary.org/article.aspx?doi=10.1117/1.3115362>.
- Goto, Y., Yamauchi, Y. dan Fujiyoshi, H., 2013. CS-HOG : Color Similarity-based HOG. Dalam *The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*. pp. 266–271.
- Guo, J., 2013. Statistical Modeling of the Gabor Filter Magnitude Using Gamma Distribution for Effectively Vehicle Verification. *ICICS 2013*, (1).
- Guo, J., Member, S., Prasetyo, H. dan Wong, K., 2014. Vehicle Verification Using Gabor Filter Magnitude with Gamma Distribution Modeling. *IEEE Signal Processing Letters*, 21(5), pp.600–604.
- Gupta, N., 2011. Comparative Study of Different Low Level Feature Extraction Techniques for Content based Image Retrieval. , 1(1), pp.39–42.
- Han, S., Han, Y. dan Hahn, H., 2009. Vehicle Detection Method using Haar-like Feature on Real Time System. *World Academy of Science, Engineering and Technology*, pp.455–459.
- Han, Z., Ye, Q. dan Jiao, J., 2011. Combined feature evaluation for adaptive visual object tracking. *Computer Vision and Image Understanding*, 115(1),

pp.69–80.

- Han, Z., Ye, Q. dan Jiao, J., 2008. Online Feature Evaluation for Object Tracking using Kalman Filter. Dalam *19th International Conference on Pattern Recognition, 2008. ICPR 2008.* pp. 1–4. Available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4761152&isnumber=4760915>.
- Hao, Y., Yin, Y. dan Lan, J., 2013. Road Vehicle Classification Based on Extreme Learning Machine. Dalam *3rd International Conference on Electric and Electronics (EEIC 2013).* pp. 430–433.
- Hasfjord, T., 2014. *Design and implementation of a Kalman Filter based estimator for temperature control.*
- Hawladar, A.K. dan Hossain, F., 2014. A New Approach of Gradient and Threshold Based Histogram Equalization for Contrast Enhancement. , pp.242–245.
- Hu, W., Xie, N., Li, L., Zeng, X. dan Maybank, S., 2011. A Survey on Visual Content-Based Video Indexing and Retrieval. *IEEE Transactions On Systems, Man, and Cybernetics-Part C: Application and Reviews*, 41(6), pp.797–819.
- Huang, D., Chen, C., Hu, W., Yi, S. dan Lin, Y., 2012. Feature-Based Vehicle Flow Analysis and Measurement for a Real-Time Traffic Surveillance System. *Journal of Information Hiding and Multimedia Signal Processing Ubiquitous International*, 3(3), pp.282–296.
- Ince, E., 2011. Measuring traffic flow and classifying vehicle types: A surveillance video based approach. *Turk J Elec Eng & Comp Sci*, 19(4), pp.607–620.
- Jialiang, F., Jiabi, D. dan Sifa, Z., 2016. Combination of Wavelet Transform and Mutual Information for Vehicle Recognition. Dalam *International Congress on Sound & Vibration.* pp. 1–8.
- Kadir, A. dan Susanto, A., 2013. *Teori dan Aplikasi Pengolahan Citra*, Yogyakarta: Penerbit Andi Yogyakarta.
- Kafai, M. dan Bhanu, B., 2012. Dynamic Bayesian Networks for Vehicle

- Classification in Video. *IEEE Transactions on Industrial Informatics*, 8(1), pp.100–109.
- Kakarwal, S. dan Deshmukh, R., 2010. Wavelet Transform based Feature Extraction for Face Recognition. *International Journal of Computer Science and Application Issue 2010*.
- Karaimer, H.C., Cinaroglu, I. dan Bastanlar, Y., 2015. Combining Shape-Based and Gradient-Based Classifiers for Vehicle Classification. Dalam *2015 IEEE 18th Conference on Intelligent Transportation Systems*. pp. 800–805.
- Kavya, R. dan Harisha, 2015. Feature Extraction Technique for Robust and Fast Visual Tracking: A Typical Review. *International Journal of Emerging Engineering Research and Technology*, 3(1), pp.98–104. Available at: <http://www.ijeert.org/pdf/v3-i1/16.pdf>.
- Kemenhub, 2009. Government Regulation of the Republic of Indonesia no. 22 year 2009 about Traffic and Road Transportation. , pp.1–203.
- Kemenhub, 2012. Government Regulation of the Republic of Indonesia no. 55 year 2012 about Vehicle. , pp.1–118.
- Kemenhub, 2014. Peraturan Pemerintah Republik Indonesia Nomor 74 Tahun 2014 tentang Angkutan Jalan. , pp.1–78. Available at: <http://hubdat.dephub.go.id/peraturan-pemerintah/1678-peraturan-pemerintah-republik-indonesia-nomor-74-tahun-2014-tentang-angkutan-jalan>.
- Kemenhub, 2013. Peraturan Pemerintah Republik Indonesia Nomor 79 Tahun 2013 tentang Jaringan Lalu Lintas dan Angkutan Jalan. , pp.1–97. Available at: <http://hubdat.dephub.go.id/peraturan-pemerintah/1596-peraturan-pemerintah-nomor-79-tahun-2013-tentang-jaringan-lalu-lintas-dan-angkutan-jalan>.
- Klein, L.A. dan Kelley, M.R., 1996. Detection Technology for IVHS Volume I: Final Report. *Technical Report No. FHWA-RD-95-100, U.S. Department of Transportation*.
- Kocadag, F. dan Demirkol, A., 2015. Real time tracking of TV satellites on moving vehicles using Kalman filter. *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pp.1–5. Available at:

- <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7095286>.
- Kodwani, L., 2013. *Automatic Vehicle Detection, Tracking and Recognition of License Plate in Real Time Videos*.
- Kota, R.K. dan Rao, C.S., 2013. Analysis Of Classification and Tracking In Vehicles Using Shape Based Features. *International Journal of Innovative Research and Development*, 2(8), pp.279–286.
- Lee, S., Bang, M. dan Jung, K., 2015. An Efficient Selection of HOG Feature for SVM Classification of Vehicle. Dalam *2015 IEEE International Symposium on Consumer Electronics (ISCE)*. pp. 1–2.
- Li, G., Zhang, J., Lin, H., Tu, D. dan Zhang, M., 2004. A moving object detection approach using integrated background template for smart video sensor. *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.04CH37510)*, pp.462–466. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1351088>.
- Li, H., Bai, P. dan Song, H., 2014. Car Tracking Algorithm Based on Kalman Filter and Compressive Tracking. , pp.27–31.
- Li, J., Wang, J. dan Shen, W., 2010. Moving object detection in framework of compressive sampling. *Journal of Systems Engineering and Electronics*, 21(5), pp.740–745.
- Li, X., Fu, H. dan Xu, J., 2008. Automatic Vehicle Classification Based on Video with BP Neural Networks. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, (2), pp.1–3. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4681249>.
- Lin, L. dan Siang, H., 2012. Vision-based activities recognition by trajectory analysis for parking lot surveillance. *2012 IEEE International Conference on Circuits and Systems (ICCS)*, pp.137–142. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6408305>.
- Liu, H., Chen, S. dan Kubota, N., 2013a. Intelligent Video Systems and Analytics : A Survey. *IEEE Transactions On Industrial Informatics*, 9(3), pp.1222–1233.
- Liu, Y., Tian, B., Chen, S., Zhu, F. dan Wang, K., 2013b. A survey of vision-

- based vehicle detection and tracking techniques in ITS. Dalam *Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety*. Ieee, pp. 72–77. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6619606>.
- Liu, Z. dan Yang, H., 2010. A New Vehicle Tracking Method with Region Matching Based on Kalman Forecasting Model. *Engineering*, pp.559–563.
- Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp.91–110.
- Mann, B.P., Gibbs, M.M. dan Sah, S.M., 2012. Dynamics of a gravity car race with application to the Pinewood Derby. *Mechanical Sciences*, 3(2), pp.73–84. Available at: <http://www.mech-sci.net/3/73/2012/>.
- Mathew, T. V. dan Rao, K.V.K., 2006. *Introduction to Transportation Engineering: Chapter 30. Fundamental parameters of traffic flow*, India: NPTEL.
- Matos, F. dan Souza, R., 2013. Vehicle Image Classification Method Using Edge Dimensions, SVM and Prototype. Dalam *International Conference on Artificial Intelligence*. pp. 1–7. Available at: <http://worldcomp-proceedings.com/proc/p2013/ICA7275.pdf>.
- Matos, F.M.D.S. dan Souza, R.M.C.R. De, 2012. An image vehicle classification method based on edge and PCA applied to blocks. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.1688–1693. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6377980>.
- Matrella, G. dan Marani, D., 2011. An Embedded Video Sensor for a Smart Traffic Light. *2011 14th Euromicro Conference on Digital System Design*, pp.769–776. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6037486> [Diakses Maret 12, 2012].
- May, A.D., 1990. *Traffic Flow Fundamentals*, USA: Prentice-Hall, Inc.
- Messelodi, S. dan Maria, C., 2007. Vision-based bicycle / motorcycle classification. *Pattern Recognition Letters*, 28(13), pp.1719–1726.

- Mingqiang, Y., Kidiyo, K. dan Joseph, R., 2008. A survey of shape feature extraction techniques. *Pattern Recognition*, 2008, pp.43–90.
- Mishra, P.K. dan Banerjee, B., 2013. Multiple Kernel Based KNN Classifiers for Vehicle Classification. *International Journal of Computer Applications*, 71(6), pp.1–7.
- Mithun, N.C., Rashid, N.U. dan Rahman, S.M.M., 2012. Detection and Classification of Vehicles From Video Using Multiple Time-Spatial Images. *IEEE Transactions on Intelligent Transportation Systems*, 13(3), pp.1215–1225. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6156444>.
- Mohamed, A., Issam, A., Mohamed, B. dan Abdellatif, B., 2015. Real-time Detection of Vehicles Using the Haar-like Features and Artificial Neuron Networks. *Procedia Computer Science*, 73(Awict), pp.24–31. Available at: <http://dx.doi.org/10.1016/j.procs.2015.12.044>.
- Morris, B.T. dan Trivedi, M.M., 2008. Learning, modeling, and classification of vehicle track patterns from live video. *IEEE Transactions on Intelligent Transportation Systems*, 9(3), pp.425–437.
- Munawar, A., 2007. Public Transport Reform in Indonesia , A Case Study in the City of Yogyakarta. *World Academy of Science, Engineering and Technology* 28 2007, 1(4), pp.77–82. Available at: <http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.193.4052>.
- Munawar, A., 2008a. Sustainable Urban Public Transport Planning in Indonesia , Case Studies in Yogyakarta and Bandung. *The 2007 International Conference and Annual Meeting of Chinese Institute of Transportation Challenges and Opportunities towards More Sustainable Transportation*, (May), pp.13–15.
- Munawar, A., 2008b. Sustainable Urban Public Transport Planning in Indonesia , Case Studies in Yogyakarta and Jakarta. *Indonesian Students' Scientific Meeting* 2008, (May), pp.13–15. Available at: <http://munawar.staff.ugm.ac.id/wp-content/munawardelftfinal.pdf>.
- Muzakki, M.E., Mubarak, M.S., Adiwijaya, P., Telekomunikasi, J., Ters, N., Batu,

- B. dan Indonesia, B., 2017. Klasifikasi Jenis Kendaraan dengan Gabor Filtering dan Naive Bayes. Dalam *e-Proceeding of Engineering*. pp. 1083–1092.
- Neumann, D., Langner, T., Ulbrich, F., Spitta, D. dan Goehring, D., 2017. Online vehicle detection using Haar-like, LBP and HOG feature based image classifiers with stereo vision preselection. *IEEE Intelligent Vehicles Symposium, Proceedings*, (Iv), pp.773–778.
- Nixon, M.S. dan Aguado, A.S., 2008. *Feature Extraction & Image Processing* Second., London, UK: Elsevier.
- Nurchaya, E.S., 2014. Prinsip Kerja Kamera Video. , pp.1–23. Available at: <http://www.slideshare.net/endrasnc/prinsip-kerja-kamera-video>.
- Nurhadiyatna, A., Latifah, A.L. dan Fryantoni, D., 2015. Gabor filtering for feature extraction in real time vehicle classification system. *9th International Symposium on Image and Signal Processing and Analysis, ISPA 2015*, (Ispa), pp.19–24.
- Obula, R., Reddy, K., Eswara Reddy, B. dan Reddy, E.K., 2014. an Effective Glcm and Binary Pattern Schemes Based Classification for Rotation Invariant Fabric Textures. *International Journal of Computer Engineering Science (IJCES)*, 4(1), pp.1–16. Available at: <https://sites.google.com/site/ijcesjournal/nhttp://www.ijces.com/>.
- Ojala, T., Pietikäinen, M. dan Mäenpää, T., 2002. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), pp.971–987.
- Parsons, J.J. dan Oja, D., 2012. *New Perspective on Computer Concepts 2012*, United States of America: Cengage Learning, Inc.
- Ping, Z., Qian, L. dan Siyang, Z., 2015. Adaptive background updating algorithm for traffic congestion detection based on Kalman filtering and inter-frame centroid distanc. Dalam *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. Chongqing, pp. 891–895.

- Purwanto, Y., Soleman, M.A. dan Pramunendar, R.A., 2013. Pelacakan Kendaraan Bermotor di Jalan Tol Semarang Menggunakan Kalman Filter dan Mixture of Gaussian dengan Video Kualitas Rendah. , 2013(November), pp.176–182.
- Pushpalatha, K.R., M, C. dan Karegowda, A.G., 2013. Color Histogram based Image Retrieval – A Survey. *International Journal of Advanced Research in Computer Science*, 4(11), pp.119–126.
- Qin, Z., 2008. Method of vehicle classification based on video. Dalam *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Ieee, pp. 162–164. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4601652>.
- Quan, Y.U., Hui, Z., Jian, R. dan Jin-guang, L.I.U., 2009. The Analysis of Traffic Flow Characteristics at Signalized Intersections. *2009 2nd International Conference on Power Electronics and Intelligent Transportation System*, pp.382–387.
- Rabiu, H., 2013. Vehicle Detection and Classification for Cluttered Urban Intersection. *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, 3(1), pp.445–478.
- Rashid, N.U., Mithun, N.C., Joy, B.R. dan Rahman, S.M.M., 2010. Detection And Classification of Vehicles from A Video Using Time-Spatial Image. *2010 International Conference on Electrical and Computer Engineering (ICECE)*, pp.502–505.
- Ren, X. dan Yang, Z., 2010. Research on the key issue in video sensor network. *2010 3rd International Conference on Computer Science and Information Technology*, pp.423–426. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5565117>.
- Ripolles, O., Simó, J.E., Benet, G. dan Vivó, R., 2012. Smart video sensors for 3D scene reconstruction of large infrastructures. *Multimedia Tools and Applications*. Available at: <http://link.springer.com/10.1007/s11042-012-1184-z> [Diakses Juni 27, 2013].
- Ripolles, O., Simó, J.E., Benet, G. dan Vivó, R., 2014. Smart video sensors for 3D

- scene reconstruction of large infrastructures. *Multimedia Tools and Applications*, 73(2), pp.977–993.
- Salarpour, A., Salarpour, A., Fathi, M. dan Dezfoulian, M., 2011. Vehicle Tracking using Kalman Filter and Features. *Signal & Image Processing: An International Journal (SIPIJ)*, 2(2), pp.1–8.
- Sánchez, J., Benet, G. dan Simó, J.E., 2012. Video Sensor Architecture for Surveillance Applications. *Sensors (Basel, Switzerland)*, 12(2), pp.1509–1528. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3304125&tool=pmcentrez&rendertype=abstract> [Diakses Juni 27, 2013].
- van de Sande, K.E.A., Gevers, T. dan Snoek, C.G.M., 2008. A comparison of color features for visual concept classification. Dalam *Proceedings of the 2008 international conference on Content-based image and video retrieval - CIVR '08*. p. 141. Available at: <http://portal.acm.org/citation.cfm?doid=1386352.1386376>.
- Santosh, D., Venkatesh, P. dan Poornesh, P., 2013. Tracking Multiple Moving Objects Using Gaussian Mixture Model. *Ijsce.Org*, (2), pp.114–119. Available at: <http://www.ijscce.org/attachments/File/v3i2/B1453053213.pdf>.
- Santra, A.K. dan Christy, C.J., 2012. Genetic Algorithm and Confusion Matrix for Document Clustering. *International Journal of Computer Science*, 9(1), pp.322–328. Available at: <http://ijcsi.org/papers/IJCSI-9-1-2-322-328.pdf>.
- Saran, K. dan Sreelekha, G., 2015. Traffic Video Surveillance: Vehicle Detection and Classification. Dalam *2015 International Conference on Control, Communication & Computing India*. pp. 516–521.
- Shahbaz, A., Kurnianggoro, L., Hariyono, J. dan Jo, K.H., 2016. Parameter analysis of probabilistic foreground detector for intelligent surveillance system. *IECON Proceedings (Industrial Electronics Conference)*, pp.5574–5578.
- Shaoqing, M., Zhengguang, L., Jun, Z. dan Chen, W., 2009. Real-time vehicle classification method for multi-lanes roads. *2009 4th IEEE Conference on Industrial Electronics and Applications*, pp.960–964. Available at:

- <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5138343>.
- Sharma, R., 2014. Vehicle Tracking in Extreme Noisy Channel Through Kalman Filter. *International Journal of Innovative Science and Modern Engineering (IJISME)*, (12), pp.3–6.
- Shi, C., Gan, W.S., Chong, Y.K., Apoorv, A. dan Song, K.S., 2013. A vehicular noise surveillance system integrated with vehicle type classification. *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2013*.
- Shi, H., Zou, Y., Wang, Y. dan Shi, G., 2009. A Robust Traffic Parameter Extraction Method using Texture and Entropy. *2009 IEEE Intelligent Vehicles Symposium*, pp.237–241.
- Shruthi, S., 2011. Vehicle Tracking using Convolutional Neural Network. Dalam *Proceeding of the World Congress on Engineering 2011 (WCE 2011)*. pp. 6–9.
- Sina, I., Wibisono, A., Nurhadiyatna, A., Hardjono, B., Jatmiko, W. dan Mursanto, P., 2013. Vehicle Counting and Speed Measurement using Headlight Detection. Dalam *2013 International Conference on Advanced Computer Science and Information Systems, ICACISIS 2013*. pp. 149–154.
- Srikaew, A., Kumsawat, P., Attakitmongcol, K., Sroisuwan, N. dan Sotthithaworn, C., 2009. An Energy-Based Vehicle Tracking System using Principal Component Analysis and Unsupervised ART Network. Dalam *8th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING & DATA BASES (AIKED '09)*. pp. 435–440.
- Srilekha, S., Swamy, G.N. dan Krishna, A.A., 2015. A Novel Approach for Detection and Tracking of Vehicles using Kalman Filter. Dalam *2015 International Conference on Computational Intelligence and Communication Networks*. pp. 234–236.
- Srivastava, S. dan Singh, P., 2014. Real-Time Object Tracking Using Colour Feature. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3(8), pp.56–59.
- Stallings, W., 2007. *Data and Computer Communications* Eighth. T.

- Dunkelberger, ed., New Jersey: Pearson Education, Inc.
- Stauffer, C. dan Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. *Proceeding of CVPR*, pp.246–252.
- Sugiharto, A. dan Harjoko, A., 2016. Traffic Sign Detection Based On HOG and PHOG Using Binary SVM And k-NN. *3rd International Conference on Information Technology Computer and Electrical Engineering (ICITACEE)*, pp.317–321.
- Suliman, C., Cruceru, C. dan Moldoveanu, F., 2010. Kalman Filter Based Tracking in an Video Surveillance System. *Advances in Electrical and Computer Engineering*, 10, pp.30–34.
- Sundoro, H.S. dan Harjoko, A., 2016. Vehicle Counting and Vehicle Speed Measurement Based on Video Processing. *Journal of Theoretical and Applied Information Technology*, 84(2), pp.233–241.
- Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O.D. dan Wijanarto, 2009. *Teori Pengolahan Citra Digital*, Penerbit Andi Yogyakarta.
- Tamrakar, A., Ali, S., Yu, Q., Liu, J., Javed, O., Divakaran, A., Cheng, H., Sawhney, H. dan Sarnoff, S.R.I.I., 2012. Evaluation of Low-Level Features and their Combinations for Complex Event Detection in Open Source Videos. , pp.3681–3688.
- Teoh, S.S., 2011. *Development of a Robust Monocular-Based Vehicle Detection and Tracking System*.
- Teoh, S.S. dan Bräunl, T., 2012. A Reliability Point and Kalman Filter-based Vehicle Tracking Technique. Dalam *International Conference on Intelligent Systems (ICIS 2012)*. Malaysia: Planetary Scientific Research, pp. 134–138. Available at: <http://therevproject.com/publications/uwa/C2012-ICIS-A Reliability Point and Kalman Filter based Vehicle Tracking Technique-Teoh Braunl.pdf>.
- Tian, Q., Zhang, L., Wei, Y., Zhao, W. dan Fei, W., 2013. Vehicle Detection and Tracking at Night in Video Surveillance. *International Journal of Online Engineering (iJOE)*, 9(S6), p.60. Available at: <http://online-journals.org/i-joe/article/view/2828>.

- Tomasi, M., Pundlik, S. dan Luo, G., 2014. FPGA-DSP co-processing for feature tracking in smart video sensors. *Journal of Real-Time Image Processing*.
- Tomasi, M., Pundlik, S. dan Luo, G., 2016. FPGA–DSP co-processing for feature tracking in smart video sensors. *Journal of Real-Time Image Processing*, 11(4), pp.751–767.
- Vinayak, V. dan Jindal, S., 2017. CBIR System using Color Moment and Color Auto-Correlogram with Block Truncation Coding. *International Journal of Computer Applications*, 161(9), pp.1–7.
- Viola, P. dan Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. Dalam *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. pp. I–511–I–518. Available at: <http://ieeexplore.ieee.org/document/990517/>.
- Visa, S., Ramsay, B., Ralescu, A. dan Van Der Knaap, E., 2011. Confusion matrix-based feature selection. *CEUR Workshop Proceedings*, 710, pp.120–127.
- Wang, H., Bo, W. dan Sun, L., 2013. Pyramid Histogram of Oriented Gradient and Particles Swarm Optimization Based SVM for Vehicle Detection. Dalam *2013 7th International Conference on Image and Graphics (ICIG 2013)*. pp. 323–327.
- Wang, Q., Jia, K. dan Liu, P., 2015. Design and Implementation of Remote Facial Expression Recognition Surveillance System Based on PCA and KNN Algorithms. *Proceedings - 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2015*, pp.314–317.
- Wang, Y.-C., Han, C.-C., Hsieh, C.-T. dan Fan, K.-C., 2014. Vehicle color classification using manifold learning methods from urban surveillance videos. *EURASIP Journal on Image and Video Processing*, 2014(1), p.48. Available at: <https://jivp-urasipjournals.springeropen.com/articles/10.1186/1687-5281-2014-48>.
- Welch, G. dan Bishop, G., 2006. An Introduction to the Kalman Filter. *In Practice*, 7(1), pp.1–16. Available at:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.6578&rep=rep1&type=pdf>.

- Wenger, B., Mandayam, S., Violante, P.J. dan Drake, K.J., 2010. Detection of anomalous events in shipboard video using moving object segmentation and tracking. *Autotestcon, 2010 Ieee*, pp.1–6.
- Wolf, M. dan Wicksteed, C., 1997. Date and time formats. *W3C NOTE NOTE-datetime-19980827*, (September 1997), pp.12–13. Available at: <http://www.w3.org/TR/NOTE-datetime>.
- Wu, X.F., Wang, X.S. dan Lu, H.Z., 2010. Motion feature extraction for stepped frequency radar based on Hough transform. *IET Radar, Sonar & Navigation*, 4(1), p.17. Available at: <http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2009.0100>.
- Xiao, H., Cai, C., Yuan, Q., Liu, X. dan Wen, Y., 2007. A Comparative Study of Feature Extraction and Classification Methods for Military Vehicle Type Recognition Using Acoustic and Seismic Signals. , pp.810–819.
- Xiong, N. dan He, J., 2009. A Neural Network Based Vehicle Classification System for Pervasive Smart Road Security. *Journal of Universal Computer Science*, 15(5), pp.1119–1142.
- Xu, H., Zhai, G., Wu, X. dan Yang, X., 2014. Generalized equalization model for image enhancement. *IEEE Transactions on Multimedia*, 16(1), pp.68–82.
- Xu, X., Zhou, L., Mo, Q. dan Guo, Q., 2015. Vehicle Detection Algorithm Based On Codebook and Local Binary Patterns Algorithms. *Journal of Central South University*, 22(2), pp.593–600. Available at: <http://link.springer.com/10.1007/s11771-015-2560-4>.
- Yadav, G., Maheshwari, S. dan Agarwal, A., 2014. Contrast limited adaptive histogram equalization based enhancement for real time video system. *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, pp.2392–2397.
- Yang, M., Kpalma, K. dan Ronsin, J., 2008. A survey of shape feature extraction techniques. *Pattern recognition*, 2008(November), pp.43–90. Available at: [http://hal.archives-](http://hal.archives-citeulike-article-id:10157537)

ouvertes.fr/docs/00/44/60/37/PDF/ARS-Journal-

SurveyPatternRecognition.pdf\nhttp://hal.archives-ouvertes.fr/hal-00446037/.

Zhang, B. dan Zhao, C., 2011. Classification of Vehicle Make by Combined Features and Random Subspace Ensemble. Dalam *2011 Sixth International Conference on Image and Graphics (ICIG)*. Ieee, pp. 920–925. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6005630> [Diakses Mei 25, 2013].

Zhang, D., Qu, S. dan Liu, Z., 2008. Robust Classification of Vehicle based on Fusion of TSRP and Wavelet Fractal Signature. *2008. ICNSC 2008. IEEE International Conference on Networking, Sensing and Control*, pp.1788–1793.

Zou, Y., Shi, H., Wang, Y., Shi, G. dan Zhao, H., 2009. A Robust Traffic Parameter Extraction Approach for Surveillance System at Urban Intersection. *2009. CISP '09. 2nd International Congress on Image and Signal Processing*, pp.1–5.

Lampiran A. Kumpulan Algoritma

1	Fungsi EkstrakVideo()
2	baca video
3	ULANGI untuk tiap frame,
4	jpeg <= baca frame saat ini
5	bentuk frame dalam format jpeg
6	End-ULANGI
7	End-Fungsi

Gambar A.1. Algoritme Ekstrak video ke frame-frame

1	Fungsi Histogram()
2	baca fungsi ekstrak video
3	baca ukuran citra f(M,N) dalam M baris dan N kolom
4	Inisialisasi jumlah aras keabuan: L = 256
5	buatlah larik hist sebanyak L dan isi dengan nol
6	For i ← 1 to M
7	For j ← 1 to N
8	hist(f(i,j) + 1) ← hist(f(i,j)+1) + 1
9	End-For
10	End-For
11	End-Fungsi

Gambar A.2. Algoritme *Histogram*

1	Fungsi HistogramEqualization()
2	baca fungsi ekstrak video
3	baca ukuran citra f(M,N) dalam M baris dan N kolom
4	Inisialisasi jumlah aras keabuan : L = 256
5	buatlah larik hist sebanyak L dan isi dengan nol
6	Hitung histogram citra menggunakan Algoritme A.2 dengan hasil berupa hist
7	$\alpha \leftarrow (L - 1) / (M \times N)$
8	C[1] ← α * hist[1]
9	For i ← 1 to L - 1
10	c[i+1] ← c[i] + round(α * hist[i+1])
11	End-For
12	For y ← 1 to M
13	For x ← 1 to N
14	g(y,x) ← c[f(y,x)]
15	End-For
16	End-For
17	End-Fungsi

Gambar A.3. Algoritme *Histogram Equalization*

```

1 Fungsi Erosi( )
2   baca ukuran citra F(M,N) dalam M baris dan N kolom
3   baca elemen penstruktur H(s,t) dalam s baris dan t kolom
4   if margin < 3
5       buat ordinat hotspot untuk nomor baris: hoty = round(M/2);
6       buat absis hotspot untuk nomor kolom: hotx = round(N/2);
7   End-if
8   Inisialisasi Xh=[ ], Yh=[ ], jum_anggota=0;
9   For baris ← 1 to s
10      For kolom ← 1 to t
11          If H(baris, kolom)==1
12              jum_anggota = jum_anggota + 1;
13              Xh(jum_anggota) = -hotx + kolom;
14              Yh(jum_anggota) = -hoty + baris;
15          End-if
16      End-for
17  End-for
18  G(M,N) ← nol
19  For baris ← 1 to M
20      For kolom ← 1 to N
21          cocok = true;
22          For indeks ← 1 to jum_anggota
23              xpos = kolom + Xh(indeks);
24              ypos = baris + Yh(indeks);
25              If (xpos >= 1) && (xpos<=N) && (ypos >= 1) && (ypos <= M)
26                  If F(ypos, xpos) ~= 1
27                      cocok = false;
28                      break;
29                  End-if
30              Else-if
31                  cocok = false;
32              End-if
33          End-for
34          If cocok
35              G(baris, kolom) = 1;
36          End-if
37      End-for
38  End-for
39 End-fungsi

```

Gambar A.4. Algoritme Erosi

```

1 Fungsi Dilasi( )
2   baca ukuran citra F(M,N) dalam M baris dan N kolom
3   baca elemen penstruktur H(s,t) dalam s baris dan t kolom
4   if margin < 3
5       buat ordinat hotspot untuk nomor baris: hoty = round(M/2);
6       buat absis hotspot untuk nomor kolom: hotx = round(N/2);
7   End-if
8   Inisialisasi Xh=[ ], Yh=[ ], jum_anggota=0;
9   For baris ← 1 to s
10      For kolom ← 1 to t
11          If H(baris, kolom)==1
12              jum_anggota = jum_anggota + 1;
13              Xh(jum_anggota) = -hotx + kolom;
14              Yh(jum_anggota) = -hoty + baris;
15          End-if
16      End-for
17  End-for
18  G(M,N) ← nol
19  For baris ← 1 to M
20      For kolom ← 1 to N
21          For indeks ← 1 to jum_anggota
22              If F(baris, kolom) == 1
23                  xpos = kolom + Xh(indeks);
24                  ypos = baris + Yh(indeks);
25                  If (xpos >= 1) && (xpos <= N) && (ypos >= 1) && (ypos <= M)
26                      G(ypos, xpos) = 1;
27                  End-if
28              End-if
29          End-for
30      End-for
31  End-for
32 End-fungsi

```

Gambar A.5. Algoritme Dilasi

```

1 Fungsi EkstraksiCiri()
2     baca ukuran frame f(M,N) dalam M baris dan N kolom
3     ubah frame dalam aras keabuan
4     buat larik dengan nama matrix dan isi dengan kosong: matrix = [ ]
5     gg = M / 4
6     kk = gg * 3
7     baca kotak bounding box yang ada di dalam frame
8     baca ukuran kotak bounding box B(X,Y) dalam X baris dan Y kolom
9     For d ← 1 to X
10        if B(d,2)<gg || B(d,2) > kk
11            matrix = [ matrix d]
12        end-if
13    End-For
14    B(matrix,:) = [ ]
15    Buat larik numbox dan isi dengan nol
16    For ff ← 1 to numbox
17        bb = B(ff,:)
18        cropimage = imcrop(frame,bb)
19        if cropimage tidak kosong
20            Baca ukuran cropimage(lk,op) dalam lk baris dan op kolom
21            if ukuran tinggi area > 20 dan lebar area >20
22                % hitung persamaan 39 ciri
23                width = bb(3)
24                height = bb(4)
25                area = width * height
26                centroid = [bb(1)+(0.5*width),bb(2)+(0.5*height)
27                ratio=double(height)/double(width)
28                perimeter=(2*height)+(2*width)
29                compactness = (4*3.14*double(area)) /
30                (double(perimeter)*double(perimeter))
31                colorHOG = extractHOGFeatures(cropimage)
32                I = double(rgb2gray(cropimage))
33                cropimage2 = histeq(imcrop(rgb2gray(frame),bb))
34                grayHOG = extractHOGFeatures(I)
35                mapping = getmapping(8,'u2')
36                H2 = lbp(cropimage2,3,8,mapping,'h')
37                H1 = H2 / norm(H2,2)
38                [C,S] = wavedec2(cropimage,2,'sym4')
39                [Ea,EDetails] = wenergy2(C,S)
40                bin = 8
41                angle = 360
42                L = 3
43                roi = [bb(2)+height;bb(2);bb(1);bb(1)+width]
44                PHOGfeature = anna_phog(frame,bin,angle,L,roi)
45                PHOGfeature1 = PHOGfeature'

```

Gambar A.6. Algoritma Ekstraksi Ciri

```

44 humomoent = hu_moments(I)
45 colormoments = colorMoments(cropimage)
46 glcm = graycomatrix(I)
47 stats = graycoprops(glcm,'all')
48 statisticaltexture=[stats.Contrast          stats.Correlation
stats.Energy stats.Homogeneity]
49 cropimagebw = im2bw(imcrop(mask,bb))
50 stats = regionprops(cropimagebw,'all')
51 convexhull=regionprops(cropimagebw, 'ConvexHull')
52 diameter = regionprops(cropimagebw, 'EquivDiameter')
53 fillarea = regionprops(cropimagebw, 'FilledArea')
54 whitearea = regionprops(cropimagebw, 'FilledImage')
55 majoraxis      =      regionprops(cropimagebw,
'MajorAxisLength')
56 minoraxis      =      regionprops(cropimagebw,
'MinorAxisLength')
57 perimeterobject = regionprops(cropimagebw, 'Perimeter')
58 solidity = regionprops(cropimagebw, 'Solidity')
59 orientation = regionprops(cropimagebw, 'Orientation')
60 extent = regionprops(cropimagebw, 'Extent')
61 eulernumber = regionprops(cropimagebw, 'EulerNumber')
62 Eccentricity = regionprops(cropimagebw, 'Eccentricity')
63 Skewness = skewness(cropimagebw)
64 Entropy = entropy(cropimagebw)
65 standard = std2(cropimagebw)
66 ggg = stats.ConvexHull
67 convexhullx = stats(1).ConvexHull(:,1)'
68 convexhully = stats(1).ConvexHull(:,2)'
69 axisratio   =   double(stats(1).MajorAxisLength)   /
double(stats(1).MinorAxisLength)
70 circularity   =   (4*3.14*stats(1).Area)   /
(stats(1).Perimeter)^2
71 dispersedness = (stats(1).Perimeter^2) / stats(1).FilledArea
72 arearatio = double(stats(1).Area) / double(area)
73         End-if
74     End-if
75 End-for
76 End-fungsi

```

Gambar A.6. Algoritma Ekstraksi Ciri (lanjutan)

1	Function RepresentationFeatureToXML()
2	buat dokumen node dalam XML dan masukkan dalam variabel docNode
3	identifikasi elemen root dalam toc
4	set list feature_XML
3	tambahkan elemen node 'HFrame' untuk menghasilkan Header Frame.
4	beri nama atribut 'Value', 'LocationNm' dalam elemen HFrame
5	isi 'location' untuk diisi lokasi video
6	tambahkan elemen dalam variabel product
7	identifikasi elemen node 'HFrame' untuk menghasilkan Header Frame dalam variabel product
8	beri atribut 'Value', 'VideoNm' dalam elemen HFrame
9	isi 'name_of_video' untuk diisi nama video
10	tambahkan elemen dalam variabel product
11	identifikasi elemen node 'HFrame' untuk menghasilkan Header Frame dalam variabel product
12	beri atribut 'Value', 'FrameNo' dalam elemen HFrame
13	isi 'num2str(count)' untuk memberi nomor frame
14	tambahkan elemen dalam variabel product
15	inisialisasi format waktu c = datetime ('now', 'TimeZone', 'local', 'Format', 'yyyy-MM-dd"T" 'HH:mm:ss.MSZ');
16	identifikasi elemen node 'HFrame' untuk menghasilkan Header Frame dalam variabel datenode
17	beri atribut 'Value', 'FTimes' dalam elemen HFrame
18	isi 'char(c)' untuk time stamp
19	tambahkan elemen dalam variabel product
20	numnode = docNode.createElement('ObjectNo');
21	numnode.setAttribute('Value','Object');
22	numnode.appendChild(docNode.createTextNode(num2str(ff)));
23	product.appendChild(numnode);
24	for idx = 1:numel(functions)
25	curr_node = docNode.createElement('Feature');
26	curr_file = [functions2{idx}];
27	curr_node.setAttribute('Value',curr_file);
28	curr_node.appendChild(docNode.createTextNode(functions{idx}));
29	product.appendChild(curr_node);
30	End-for
31	tambahkan elemen node 'ObjectNo' untuk menghasilkan node tiap objek. Setiap elemen ObjectNo di dalam file ini memiliki sebuah atribut 'Value', 'Object' dan siapkan numnode untuk memberi nomor objek
32	baca fungsi EkstraksiCiri() mulai dari objek ke-1 sampai semua objek yang tersegmen. Setiap elemen Object memiliki banyak ciri. Node ciri diberi nama elemen 'Feature'. Tiap ciri bisa berisi lebih dari satu nilai ciri. Nilai ciri memiliki atribut 'Value' dan 'curr_file' untuk nilai ciri. Tiap ciri memiliki nama ciri. Nama ciri dibuat dengan menambahkan node teks.

Gambar A.7. Ciri dibentuk dalam format XML

```

1 function theStruct = parseXML( )
2 try tree = xmlread(filename); % baca file xml masukkan dalam variabel tree
3 catch
4     error('Failed to read XML file %s.',filename);
5     % bila tidak berhasil beri komentar bahwa file xml tidak berhasil dibaca
6     % dan program berhenti
7 end
8 try theStruct = parseChildNodes(tree);
9     % coba baca node nilai ciri yang ada didalam variabel tree dan
10    % masukkan dalam variabel theStruct
11 catch
12    error('Failed to parse XML file %s.',filename);
13    % bila tidak berhasil beri komentar parse file xml tidak berhasil dibaca
14    % dan program berhenti
15 end
16 function children = parseChildNodes(theNode)
17 children = [ ]; % sediakan array variabel children yang masih kosong
18 if theNode.hasChildNodes % jika node memiliki childNodes maka
19     childNodes = theNode.getChildNodes; % masukkan node ke dalam
20     variabel childNodes
21     numChildNodes = childNodes.getLength; % masukkan panjang
22     childNodes ke dalam variabel numchildNodes
23     allocCell = cell(1, numChildNodes); % siapkan array sepanjang jumlah
24     numChildNodes dan masukkan dalam variabel allocCell
25     children = struct('Name', allocCell, 'Attributes', allocCell, ...
26                     'Data', allocCell, 'Children', allocCell);
27     % bentuk struktur array children
28     for count = 1:numChildNodes % ulangi untuk count = 1 sampai semua
29         numChildNodes selesai dibaca
30         theChild = childNodes.item(count-1); % lakukan baca item childNode
31         yang ke count-1 dan masukkan dalam variabel theChild
32         children(count) = makeStructFromNode(theChild); % masukkan
33         variabel theChild dalam array children
34     end %end-ulangi
35 end %end-jika
36 function nodeStruct = makeStructFromNode(theNode)
37 nodeStruct=struct('Name',char(theNode.getNodeName),'Attributes',
38                 parseAttributes(theNode),'Data','Children',parseChildNodes(theNode));
39     % buat struktur node nodeStruct
40 if any(strcmp(methods(theNode), 'getData')) % jika ada, bandingkan string
41     yang ada dengan data yang diambil
42     nodeStruct.Data = char(theNode.getData); % masukkan karakter yang
43     diambil dalam nodeStruct.Data
44 else nodeStruct.Data = ''; % nodeStruct.Data diisi dengan kosong
45 end % end-jika

```

Gambar A.8. konversi ciri dalam struktur array


```

28 function attributes = parseAttributes(theNode)
29 attributes = [ ]; % sediakan array variabel attributes yang masih kosong
30 if theNode.hasAttributes % jika theNode berisi atribut
31     theAttributes = theNode.getAttributes; % simpan atribut dari theNode
        didalam variabel theAttributes
32     numAttributes = theAttributes.getLength; % hitung panjang variabel
        theAttributes dan masukkan dalam variabel
        numAttributes
33     allocCell = cell(1, numAttributes); % siapkan array sepanjang jumlah
        numAttributes dan masukkan dalam variabel
        allocCell
34     attributes = struct('Name', allocCell, 'Value', allocCell); % bentuk struktur
        array attributes
35     for count = 1:numAttributes % ulangi untuk count = 1 sampai semua
        numAttributes selesai dibaca
36         attrib = theAttributes.item(count-1); % lakukan baca item theAttribute
        yang ke count-1 dan masukkan dalam variabel attrib
37         attributes(count).Name = char(attrib.getName); % ambil nama yang
        ada di dalam variabel attrib lalu ubah ke bentuk
        karakter kemudian masukkan dalam struktur Name
        attributes
38         attributes(count).Value = char(attrib.getValue); % ambil nilai yang
        ada di dalam variabel attrib lalu ubah ke bentuk
        karakter kemudian masukkan dalam struktur Value
        attributes
39     end %end-ulangi
40 end %end-jika

```

Gambar A.8. konversi ciri dalam struktur array (lanjutan)

```

1 function feature=makevariable_frame(dataxml)
2     a=dataxml.Children(6).Children;
3     [x,zxc]=size(a);
4     feature={};
5     counthead=1;
6     for dd=1:zxc
7         f=dataxml.Children(6).Children(dd).Name;
8         gg=strcmp(f,'HFrame'); %check header
9         if gg==1
10            feature{1,1}{1,1}=dataxml.Children(2).Children.Data;
11            feature{1,1}{2,1}=dataxml.Children(4).Children.Data;
12            feature{1,1}{3,1}=str2double(dataxml.Children(6).Children(1).Data);
13            feature{1,1}{4,1}=dataxml.Children(6).Children(dd).Children.Data;
14            counthead=counthead+1;
15        end
16        g=strcmp(f,'ObjectNo');
17        if g==1
18            numberobject=dataxml.Children(6).Children(dd).Children.Data;
19            count=1;
20        end
21        jj=strcmp(f,'Feature');
22        if jj==1
23            If count==4 || count==8 || count==9 || count==10 || count==11 ||
                count==12 || count==13 || count==14 || count==15 || count==17 ||
                count==23 || count==24 || count==32 || count==33
24                kk=dataxml.Children(6).Children(dd).Children.Data;
25                mm=size(kk,2);
26                jj=kk(2:mm-1);
27                nn=strsplit(jj,' ');
28                gg=size(nn,2);
29                oo=[ ];
30                for tt=1:gg
31                    oo=[oo str2double(nn{1,tt})];
32                end
33            elseif count ==40||count ==41
34                oo=dataxml.Children(6).Children(dd).Children.Data;
35            else
36                oo=str2num( dataxml.Children(6).Children(dd).Children.Data);
37            end
38            feature{str2num(numberobject)+1,1}{count,1}=oo;
39            count=count+1;
40        end
41    end
42 end

```

Gambar A.9. mengubah variabel string ke variabel number

```

1 function final = makestruct(result)
2   final=[]; % create an empty array of tracks
3   for h=2:size(result,1)
4     finaltemp=[];
5     finaltemp.LocationNm = result{1,1}{1,1};
6     finaltemp.VideoNm = result{1,1}{2,1};
7     finaltemp.FrameNo = result{1,1}{3,1};
8     finaltemp.FTime = result{1,1}{4,1};
9     finaltemp.WBB = result{h,1}{1,1};
10    finaltemp.HBB=result{h,1}{2,1};
11    finaltemp.AreaROI =result{h,1}{3,1};
12    finaltemp.Centroid= result{h,1}{4,1};
13    finaltemp.AspectRatio=result{h,1}{5,1};
14    finaltemp.PBB= result{h,1}{6,1};
15    finaltemp.Compactness= result{h,1}{7,1};
16    finaltemp.ColorHOG= result{h,1}{8,1};
17    finaltemp.GrayHOG= result{h,1}{9,1};
18    finaltemp.LBP= result{h,1}{10,1};
19    finaltemp.ColorEnergy= result{h,1}{11,1};
20    finaltemp.PyramidHOG =result{h,1}{12,1};
21    finaltemp.HuMoment= result{h,1}{13,1};
22    finaltemp.Colormoment= result{h,1}{14,1};
23    finaltemp.StatisticalTexture= result{h,1}{15,1};
24    finaltemp.ObjectArea= result{h,1}{16,1};
25    finaltemp.MassPoint= result{h,1}{17,1};
26    finaltemp.MajorAxis= result{h,1}{18,1};
27    finaltemp.MinorAxis= result{h,1}{19,1};
28    finaltemp.Perimeter= result{h,1}{20,1};
29    finaltemp.EquivDiameter= result{h,1}{21,1};
30    finaltemp.Solidity= result{h,1}{22,1};
31    finaltemp.ConvexHullX= result{h,1}{23,1};
32    finaltemp.ConvexHullY= result{h,1}{24,1};
33    finaltemp.ConvexArea= result{h,1}{25,1};
34    finaltemp.Angle= result{h,1}{26,1};
35    finaltemp.Extent= result{h,1}{27,1};
36    finaltemp.EulerNumber= result{h,1}{28,1};
37    finaltemp.Eccentricity= result{h,1}{29,1};
38    finaltemp.FilledArea= result{h,1}{30,1};
39    finaltemp.AreaRatio= result{h,1}{31,1};
40    finaltemp.BoundingBox= result{h,1}{32,1};
41    finaltemp.Skewness= result{h,1}{33,1};
42    finaltemp.Entropy= result{h,1}{34,1};
43    finaltemp.StandardDeviation= result{h,1}{35,1};
44    finaltemp.AxisRatio= result{h,1}{36,1};
45    finaltemp.Circularity= result{h,1}{37,1};

```

Gambar A.10. mengubah dalam bentuk tabel

46	finaltemp.WhiteArea= result{h,1}{38,1};
47	finaltemp.Dispersedness= result{h,1}{39,1};
48	final=[final;finaltemp];
49	end
50	end

Gambar A.10. mengubah dalam bentuk tabel (lanjutan)

1	Fungsi BuatDataset()
2	Baca fungsi EkstrakVideo pada Algoritme 1
3	Lakukan pra pemrosesan dengan <i>Histogram Equalization</i> menggunakan Algoritme 3
4	Lakukan segmentasi dengan <i>Gaussian Mixture Model</i>
5	Simpan segmen-segmen citra dengan extention .jpeg
6	Kelompokkan citra berdasarkan kelas kendaraannya
7	Baca Algoritme penyatuan dataset
8	End-Fungsi

Gambar A.11. Pembuatan dataset

1	Fungsi PenyatuanDataset()
2	Inisialisasi fitur, kelas, direktori, ukuran
3	Baca tiap daerah lokasi
4	Baca tiap video di lokasi itu
5	Baca segmen yang dituju
6	Baca data dalam tiap kelas
7	Simpan ukuran tiap citra dalam variabel hl
8	Baca tiap file citra
9	Baca fitur lbp
10	Simpan fitur berdasarkan fitur dan kelasnya
11	End-Fungsi

Gambar A.12. Penyatuan dataset

Lampiran B. Manfaat Ciri

Tabel B.1 Ciri Bentuk

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
1.	Bounding Box	(Sina dkk., 2013), (C dan Kumar, 2015)	(Sánchez dkk., 2012; Morris dan Trivedi, 2008; Kafai dan Bhanu, 2012; Rabiu, 2013)	(Sina dkk., 2013) (Sánchez dkk., 2012)	(Boyle dan Ferryman, 2015)	(Sina dkk., 2013)	(Sina dkk., 2013)(Basri dkk., 2015)	
2.	WBB	(Chan dkk., 2012) (Mithun dkk., 2012)	(Chen dkk., 2011; Mithun dkk., 2012; Mishra dan Banerjee, 2013)	(Chan dkk., 2012)				
3.	HBB	(Chan dkk., 2012)	(Chen dkk., 2011)	(Chan dkk., 2012)				
4.	Major axis		(Chen dkk., 2011)					
5.	Minor axis		(Chen dkk., 2011)					

Tabel B.1 Ciri Bentuk (lanjutan)

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
6.	Area ROI	(Sina dkk., 2013)				(Sundoro dan Harjoko, 2016)	(Sundoro dan Harjoko, 2016)	
7.	Centroid	(Sina dkk., 2013) (C dan Kumar, 2015) (Mithun dkk., 2012)				(Sundoro dan Harjoko, 2016)	(Sundoro dan Harjoko, 2016)	
8.	Filled Area		(Chen dkk., 2011)					
9.	Object Area	(Sina dkk., 2013) (Mithun dkk., 2012)	(Chen dkk., 2011; C dan Kumar, 2015; Sánchez dkk., 2012; Mithun dkk., 2012; Mishra dan Banerjee, 2013)				(C dan Kumar, 2015)	

Tabel B.1 Ciri Bentuk (lanjutan)

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
10.	Aspect Ratio		(C dan Kumar, 2015; Mishra dan Banerjee, 2013)				(C dan Kumar, 2015)	
11.	Perimeter		(Chen dkk., 2011; Sánchez dkk., 2012)					
12.	Dispersedness		(Chen dkk., 2011; Sánchez dkk., 2012; Mithun dkk., 2012; Mishra dan Banerjee, 2013)					
13.	Invers Dispersedness		(Sánchez dkk., 2012)					
14.	Edge Density				(Bravo dkk., 2013)			
15.	grayHOG	(Arróspide dkk., 2012)	(Lee dkk., 2015)	(Tian dkk., 2013)			(C dan Kumar, 2015)	

Tabel B.1 Ciri Bentuk (lanjutan)

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
16.	PHOG	(Wang dkk., 2013)	(Mishra dan Banerjee, 2013)		(Bravo dkk., 2013)			
17.	Equivdiameter		(Chen dkk., 2011)					
18.	Convex Area		(Chen dkk., 2011)					
19.	Solidity		(Chen dkk., 2011)					
20.	Convex Hull		(Karaimer dkk., 2015)					
21.	Orientation		(Chun-Hong dan Zhi-Lei, 2016)	(Shruthi, 2011)				
22.	Extent		(Chen dkk., 2011) (Sánchez dkk., 2012)					
23.	Euler Number	(Kodwani, 2013)						
24.	Eccentricity		(Chen dkk., 2011)					
25.	Area Ratio		(Mithun dkk., 2012)					

Tabel B.1 Ciri Bentuk (lanjutan)

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
26.	Axis Ratio		(Mithun dkk., 2012)					
27.	Circularity		(Chen dkk., 2011)					
28.	Elongation		(Karaimer dkk., 2015) (Morris dan Trivedi, 2008) (Yang dkk., 2008)					
29.	Haar like feature	(Chan dkk., 2012)	(Mishra dan Banerjee, 2013)					
30.	Gabor	(Teoh, 2011)	(Muzakki dkk., 2017; Nurhadiyatna dkk., 2015).					(Guo dkk., 2014; Guo, 2013).
31.	SIFT		(Mishra dan Banerjee, 2013)					

Tabel B.2 Ciri Tekstur

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
1.	LBP	(Rabiu, 2013; Xu dkk., 2015; Neumann dkk., 2017)	(Rabiu, 2013; Firdaus dkk., 2016)					
2.	<i>Invariant Moment</i>		(Hao dkk., 2013)					
3.	<i>Statistical Texture</i>		(Abouelatta, 2013)					

Tabel B.3 Ciri Warna

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
1.	<i>ColorHOG</i>	(Goto dkk., 2013)						
2.	<i>ColorMoments</i>		(van de Sande dkk., 2008)					
3.	<i>ColorHistogram</i>	(Gandhi dan Trivedi, 2007)	(van de Sande dkk., 2008; Gandhi dan Trivedi, 2007; Wang dkk., 2014)					

Tabel B.3 Ciri Warna (lanjutan)

No.	Ciri	Deteksi	Klasifikasi	Pelacakan	Posisi ban kendaraan	Kecepatan	Jumlah	Verifikasi
4.	<i>ColorCorrelogram</i>		(Wang dkk., 2014)					
5.	<i>ColorEnergy</i>		(Zhang dkk., 2008)	(Srikaew dkk., 2009)				

Lampiran C. Kesesuaian nilai centroid dalam Kalman Filter

Pada pelacakan dengan *Kalman Filter*, objek yang dilacak berasal dari frame ke 145 daerah Condong Catur. Pelacakan objek dimulai dari inisialisasi nilai \hat{x}_0 dan P_0 . \hat{x}_0 dan P_0 ditetapkan pada awal proses rekursif *Kalman Filter*. \hat{x}_0 dan P_0 hanya digunakan satu kali saat awal proses rekursif *Kalman Filter*. \hat{x}_0 adalah posisi objek yang terlacak saat inisialisasi di frame pertama. \hat{x}_0 menyatakan *state* matrik awal yang memiliki ordo 4 x 1. Nilai \hat{x}_0 berasal dari nilai ciri *centroid*. Nilai variabel awal P_0 diberikan secara acak (*random*). Pada penelitian ini, nilai P_0 dengan ordo 4 x 4 yang diberikan adalah:

$$P_0 = \begin{bmatrix} 9,4804 & 3,6042 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 9,4804 & 3,6042 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix}$$

Nilai \hat{x}_0 dan P_0 diubah menjadi nilai state sebelumnya.

$\hat{x}_0 \rightarrow \hat{x}_{k-1}$ dapat dihitung dengan persamaan (3.118).

$$\hat{x}_0 = \begin{bmatrix} 247,9994 \\ -1,0059 \\ 413,0645 \\ 1,8514 \end{bmatrix} \rightarrow \hat{x}_{k-1} = \begin{bmatrix} 247,9994 \\ -1,0059 \\ 413,0645 \\ 1,8514 \end{bmatrix}$$

State sebelumnya \hat{x}_{k-1} berisi nilai posisi di arah x = 247,9994, nilai *velocity* di arah x = -1,0059, nilai posisi di arah y = 413,0645, nilai *velocity* di arah y = 1,8514. *State* ini menjadi titik awal untuk pembuatan *path* kendaraan.

$P_0 \rightarrow P_{k-1}$ yang dapat dihitung dengan persamaan (3.119).

$$P_0 = \begin{bmatrix} 9,4804 & 3,6042 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 9,4804 & 3,6042 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix} \rightarrow P_{k-1} = \begin{bmatrix} 9,4804 & 3,6042 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 9,4804 & 3,6042 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix}$$

Pada tahap awal, state baru diprediksi dengan $\hat{x}_k^- = A\hat{x}_{k-1} + B_{u_k}$ yang dapat dihitung dengan persamaan (3.120).

$$\hat{x}_k^- = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \\ y \\ \bar{y} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta T^2 & 0 \\ \Delta T & 0 \\ 0 & \frac{1}{2}\Delta T^2 \\ 0 & \Delta T \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

$$\hat{x}_k^- = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 247,9994 \\ -1,0059 \\ 413,0645 \\ 1,8514 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}1^2 & 0 \\ 0 & \frac{1}{2}1^2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\hat{x}_k^- = \begin{bmatrix} 247,9994 - 1,0059 \\ -1,0059 \\ 413,0645 + 1,8514 \\ 1,8514 \end{bmatrix}$$

$$\hat{x}_k^- = \begin{bmatrix} 246,9935 \\ -1,0059 \\ 414,9159 \\ 1,8514 \end{bmatrix}$$

Dari persamaan (3.120) maka diperoleh hasil state prediksi \hat{x}_k^- dengan nilai posisi di arah x = 246,9935, nilai velocity di arah x = -1,0059, nilai posisi di arah y = 414,9159, nilai velocity di arah y = 1,8514. Kemudian matriks kovarian proses (*process covariance matrix*) P_k^- dihitung dengan persamaan (3.125).

$$P_k^- = AP_{k-1}A^T + Q$$

$$P_k^- = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 9,4804 & 3,6042 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 9,4804 & 3,6042 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix} A^T + Q$$

$$P_k^- = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 9,4804 & 3,6042 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 9,4804 & 3,6042 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 9,4804+3,6042+0+0 & 3,6042+65,7584+0+0 & 0+0+0+0 & 0+0+0+0 \\ 0+3,6042+0+0 & 0+65,7584+0+0 & 0+0+0+0 & 0+0+0+0 \\ 0+0+0+0 & 0+0+0+0 & 0+0+9,4804+3,6042 & 13,0846 \\ 0+0+0+0 & 0+0+0+0 & 0+0+0+3,6042 & 65,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 13,0846 & 69,3626 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 13,0846 & 13,0846 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 13,0846 & 69,3626 & 0 & 0 \\ 3,6042 & 65,7584 & 0 & 0 \\ 0 & 0 & 13,0846 & 13,0846 \\ 0 & 0 & 3,6042 & 65,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 13,0846+69,3626+0+0 & 0+69,3626+0+0 & 0+0+0+0 & 0+0+0+0 \\ 3,6042+65,7584+0+0 & 0+65,7584+0+0 & 0+0+0+0 & 0+0+0+0 \\ 0+0+0+0 & 0+0+0+0 & 26,1692 & 13,0846 \\ 0+0+0+0 & 0+0+0+0 & 69,7584 & 65,7584 \end{bmatrix} + \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix}$$

State perlu diperbarui untuk memastikan kebenaran nilai state prediksi dengan menggunakan persamaan (3.130) dan *Kalman Gain* yang ditunjukkan pada persamaan (3.126). Langkah ini diawali dengan menentukan nilai *Kalman Gain* seperti yang ditunjukkan pada persamaan (3.126).

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \right)^{-1}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix}}{\begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix}}{\begin{bmatrix} 182,4472 & 0 \\ 0 & 126,1692 \end{bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}}$$

$$K_k = \frac{\begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix}}{\begin{bmatrix} 192,4472 & 0 \\ 0 & 136,1692 \end{bmatrix}}$$

$$K_k = \begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix} \begin{bmatrix} 192,4472 & 0 \\ 0 & 136,1692 \end{bmatrix}^{-1}$$

$$K_k = \begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix} \frac{1}{(192,4472)(136,1692) - (0)(0)} \begin{bmatrix} 136,1692 & 0 \\ 0 & 192,4472 \end{bmatrix}$$

$$K_k = \begin{bmatrix} 182,4472 & 0 \\ 69,3626 & 0 \\ 0 & 126,1692 \\ 0 & 69,7584 \end{bmatrix} \begin{bmatrix} \frac{1}{192,4472} & 0 \\ 0 & \frac{1}{136,1692} \end{bmatrix}$$

$$K_k = \begin{bmatrix} 0,9480 & 0 \\ 0,3604 & 0 \\ 0 & 0,9265 \\ 0 & 0,5123 \end{bmatrix}$$

State prediksi perlu dipastikan kebenarannya dengan persamaan (3.129) berdasarkan nilai pengukuran dari persamaan (3.131) dan nilai *Kalman Gain* dari persamaan (3.126).

$$z_k = H\hat{x}_k + v_k$$

$$z_k = H\hat{x}_k + 0$$

$$z_k = H\hat{x}_k$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(H\hat{x}_k - H\hat{x}_k^-)$$

$$\hat{x}_k = \hat{x}_k^- + K_k H\hat{x}_k - K_k H\hat{x}_k^-$$

$$\hat{x}_k - K_k H\hat{x}_k = \hat{x}_k^- - K_k H\hat{x}_k^-$$

$$(1 - K_k H)\hat{x}_k = (1 - K_k H)\hat{x}_k^-$$

$$\hat{x}_k = \hat{x}_k^-$$

$$\hat{x}_k = \begin{bmatrix} 246,9935 \\ -1,0059 \\ 414,9159 \\ 1,8514 \end{bmatrix}$$

Nilai state terkini \hat{x}_k berisi nilai posisi di arah $x = 246,9935$, nilai *velocity* di arah $x = -1,0059$, nilai posisi di arah $y = 414,9159$, nilai *velocity* di arah $y = 1,8514$. Nilai *state* terkini \hat{x}_k dan nilai *state* sebelumnya \hat{x}_{k-1} digunakan untuk membuat *path* kendaraan. Langkah selanjutnya, matrik kovarian proses terkini P_k dihitung dengan persamaan (3.132). Dengan demikian diperolehlah, nilai matriks state terkini \hat{x}_k dan matriks kovarian proses terkini P_k .

$$P_k = (I - K_k H) P_k^-$$

$$P_k = \left(I - \begin{bmatrix} 0,9480 & 0 \\ 0,3604 & 0 \\ 0 & 0,9265 \\ 0 & 0,5123 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \right) \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix}$$

$$P_k = \left(I - \begin{bmatrix} 0,9480 + 0 & 0 + 0 & 0 + 0 & 0 + 0 \\ 0,3604 + 0 & 0 + 0 & 0 + 0 & 0 + 0 \\ 0 + 0 & 0 + 0 & 0 + 0,9265 & 0 + 0 \\ 0 + 0 & 0 + 0 & 0 + 0,5123 & 0 + 0 \end{bmatrix} \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix} \right)$$

$$P_k = \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0,9480 & 0 & 0 & 0 \\ 0,3604 & 0 & 0 & 0 \\ 0 & 0 & 0,9265 & 0 \\ 0 & 0 & 0,5123 & 0 \end{bmatrix} \right) \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix}$$

$$P_k = \left(\begin{bmatrix} 0,052 & 0 & 0 & 0 \\ -0,3604 & 1 & 0 & 0 \\ 0 & 0 & 0,0735 & 0 \\ 0 & 0 & -0,5123 & 1 \end{bmatrix} \right) \begin{bmatrix} 182,4472 & 69,3626 & 0 & 0 \\ 69,3626 & 65,7584 & 0 & 0 \\ 0 & 0 & 126,1692 & 13,0846 \\ 0 & 0 & 69,7584 & 90,7584 \end{bmatrix}$$

$$P_k = \begin{bmatrix} 9,4872544 + 0 + 0 + 0 & 3,6068552 + 0 + 0 + 0 & 0 + 0 + 0 + 0 & 0 + 0 + 0 + 0 \\ -65,75397088 + 69,3626 + 0 + 0 & -24,99828104 + 65,7584 + 0 + 0 & 0 + 0 + 0 + 0 & 0 + 0 + 0 + 0 \\ 0 + 0 + 0 + 0 & 0 + 0 + 0 + 0 & 0 + 0 + 9,2734362 + 0 & 0 + 0 + 13,1581 + 0 \\ 0 + 0 + 0 + 0 & 0 + 0 + 0 + 0 & 0 + 0 - 64,63648116 + 69,7584 & 0 + 0 - 6,70324058 + 90,7584 \end{bmatrix}$$

$$P_k = \begin{bmatrix} 9,4872544 & 3,6068552 & 0 & 0 \\ 3,60862912 & 40,76011896 & 0 & 0 \\ 0 & 0 & 9,2734362 & 13,1581 \\ 0 & 0 & 5,12191884 & 84,05515942 \end{bmatrix}$$

Berikutnya, nilai k terkini diubah menjadi nilai k sebelumnya dengan simbol $k-1$ untuk state sebelumnya. Nilai \hat{x}_k diberikan ke \hat{x}_{k-1} dan nilai P_k diberikan ke P_{k-1} seperti ditunjukkan pada persamaan (3.133) dan (3.134) agar siklus Kalman Filter berikutnya dapat dilanjutkan sampai frame terakhir.

$$\hat{x}_k = \begin{bmatrix} 246,9935 \\ -1,0059 \\ 414,9159 \\ 1,8514 \end{bmatrix} \rightarrow \hat{x}_{k-1} = \begin{bmatrix} 246,9935 \\ -1,0059 \\ 414,9159 \\ 1,8514 \end{bmatrix}$$

$$\begin{aligned}
 P_k &= \begin{bmatrix} 9,4872544 & 3,6068552 & & 0 & 0 \\ 3,60862912 & 40,76011896 & & 0 & 0 \\ & 0 & 0 & 9,2734362 & 13,1581 \\ & 0 & 0 & 5,12191884 & 84,05515942 \end{bmatrix} \rightarrow P_{k-1} \\
 &= \begin{bmatrix} 9,4872544 & 3,6068552 & & 0 & 0 \\ 3,60862912 & 40,76011896 & & 0 & 0 \\ & 0 & 0 & 9,2734362 & 13,1581 \\ & 0 & 0 & 5,12191884 & 84,05515942 \end{bmatrix}
 \end{aligned}$$

Lampiran D Nama dan Jumlah dimensi ciri

No.	Nama ciri	Jumlah dimensi ciri
1.	<i>Bounding Box</i>	4
2.	<i>Width of the Bounding Box (WBB)</i>	1
3.	<i>Height of the bounding box (HBB)</i>	1
4.	<i>Major axis</i>	1
5.	<i>Minor axis</i>	1
6.	<i>Area ROI</i>	1
7.	<i>Centroid</i>	2
8.	<i>Filled Area</i>	1
9.	<i>Object Area</i>	1
10.	<i>Aspect ratio</i>	1
11.	<i>Perimeter of Bounding Box (PBB)</i>	1
12.	<i>Dispersedness</i>	1
13.	<i>Invers Dispersedness</i>	1
14.	<i>Edge density</i>	1
15.	<i>gray HOG</i>	108
16.	<i>Piramid Histogram of Oriented Gradient (PHOG)</i>	661
17.	<i>Equivdiameter</i>	1
18.	<i>Convex Area</i>	1
19.	<i>Solidity</i>	1
20.	<i>Convex Hull</i>	222
21.	<i>Orientation</i>	1
22.	<i>Extent</i>	1
23.	<i>Eulernumber</i>	1
24.	<i>Eccentricity</i>	1
25.	<i>Area ratio</i>	1
26.	<i>Axis ratio</i>	1

No.	Nama ciri	Jumlah dimensi ciri
27.	<i>Circularity</i>	1
28.	<i>Elongation</i>	1
29.	<i>Haar like feature</i>	1
30.	<i>Gabor</i>	1
31.	<i>Scale Invariant Feature Transform (SIFT)</i>	1
32.	<i>Local Binary Pattern (LBP)</i>	59
33.	<i>Invariant Moment</i>	7
34.	<i>Statistical Texture</i>	22
35.	<i>Color HOG</i>	108
36.	<i>Color Moment</i>	6
37.	<i>Color Histogram</i>	1
38.	<i>Color Correlogram</i>	1
39.	<i>Color Energy</i>	2
Total dimensi ciri		1229

Lampiran E. Data set

Data Set terdiri dari Data Pelatihan dan Data Pengujian

Data Pelatihan

No.	Nama video	Lokasi	Jumlah menit	Jumlah frame
1.	Galeria_1	Galeria	0,34 menit	1000
2.	Galeria_2	Galeria	0,34 menit	1000
3.	Galeria_5	Galeria	0,34 menit	1000
4.	Galeria_10	Galeria	0,34 menit	1000
5.	CondongCatur_1	CondongCatur	0,34 menit	1000
6.	CondongCatur_2	CondongCatur	0,34 menit	1000
7.	CondongCatur_3	CondongCatur	0,34 menit	1000
8.	CondongCatur_4	CondongCatur	0,34 menit	1000
9.	Babarsari_1	Babarsari	0,34 menit	1000
10.	Babarsari_2	Babarsari	0,34 menit	1000
11.	Babarsari_10	Babarsari	0,34 menit	1000
12.	Babarsari_14	Babarsari	0,34 menit	1000

Data Pengujian

No.	Nama video	Lokasi	Jumlah menit	Jumlah frame
1.	Galeria_3	Galeria	0,34 menit	1000
2.	Galeria_4	Galeria	0,34 menit	1000
3.	Galeria_8	Galeria	0,34 menit	1000
4.	Galeria_12	Galeria	0,34 menit	1000
5.	CondongCatur_11	CondongCatur	0,34 menit	1000
6.	CondongCatur_12	CondongCatur	0,34 menit	1000
7.	CondongCatur_13	CondongCatur	0,34 menit	1000
8.	CondongCatur_14	CondongCatur	0,34 menit	1000
9.	Babarsari_3	Babarsari	0,34 menit	1000
10.	Babarsari_4	Babarsari	0,34 menit	1000
11.	Babarsari_7	Babarsari	0,34 menit	1000
12.	Babarsari_13	Babarsari	0,34 menit	1000

Lampiran F. Rekomendasi Spesifikasi *Smart Video Sensor*

Alternatif 1:

Mini PC FUJITECH H 815i /INTEL Core i5, 3.2 Ghz/RAM 4GB/HDD 500GB
MINI PC FUJITECH H 815i

Feature:

Produk MINI PC FUJITECH H 815i memiliki kelas spesifikasi sejajar atau lebih tinggi dari PC pada umumnya. Ukurannya lebih kecil di banding dengan Personal Computer, Sudah Support semua Operating System, Design yang mewah dan ringan, Konsumsi listrik max cuma 80 Watt dan hemat tempat

Specification:

- Processor: Intel Core i5 - 4460, 3.2 Ghz
- Chipset: H81 socket 1150
- Memory DDR3 4GB 1066Mhz, Dual Channel, Max Memory 16 GB
- HDD 2.5" SATA HDD 500GB
- Card Type: LAN 100/1000 Fast Internet By Realtek 8111E
- Display Output VGA + HDMI port
- Audio Chipset Integrated by Realtek ALC887
- 3X Audio Port, USB port 2.0 = 4 x 2.0, USB port 3.0 = 2 x 3.0 ,
- VGA max shared 1760MB
- Adaptor Output 12V 5A
- Dimensi 20 x 22.5 x 6 cm
- WIFI : YES

Alternatif 2:

Mini PC ASUS VIVO VCB62B Core i5
ASUS VIVO vC62B Core i5 4210

Ram 4 GB

Hdd 500 GB

wifi, keyboard mouse

dos

Alternatif 3:

LENOVO M92p |Mini PC | core i5 3470T 2.9ghz 4GB-500GB|DVDRW|Win 7 Ori

Lenovo Thinkcenter USFF M92p

Chipset Q77 / Core i5 3470T/ 4gb / 500gb

Pc SUPER mini

Tiny lenovo M92p

Core i5 3470T

Sodim DDR3 4GB 1 keping

Hdd 500GB

DVDRW

VGA 1760MB

Licensi windows 7

Adaptor 19V

**USB3 4 port*

Kelengkapan = pc mini + adaptor

Alternatif 4:

Komputer mini INTEL NUC Core I5 D54250WYKH2 fullset

ISI Komputer:

- NUC

- Motherboard

- Processor

- Ram / memory

- Harddisk

- Adaptor

- Bracket

Features and Configurations:

Processor:4th generation Intel i5 4250U processor (1.3 GHz with turbo capability to achieve 2.6 GHz, dual-core processor with 3 MB smart cache)

Memory: 2GB of 1600/1333 MHz 1.35V DDR3L memory (NOTE: 1.5V DDR3 memory is not supported)

*Graphics:**Intel HD Graphics 4400**One Mini DisplayPort 1.2 supporting ultra-high definition
4K displays and multiple monitor functionality**One Mini HDMI* 1.4a port**Audio**Intel HD Audio2 via Mini HDMI 1.4a**Mini DisplayPort 1.2 output supporting 8 channel (7.1) digital audio**Intel HD Audio via stereo analog audio jack (microphone in/headphone
out/speaker out)**Peripheral connectivity**Four super high-speed USB 3.0 ports (two back panel ports and two front ports)**Two additional high-speed USB 2.0 ports via internal header**Storage: 500GB 2.5-inch HDD**Networking**Integrated Intel PRO 10/100/1000 Network Connection**Wireless antennae (integrated into the chassis)**Enclosure:**Silver aluminum ring with diamond cut around black top**Dimensions: 116.6mm x 112mm x 51.5mm**Power adapter: 19V, 65W DC-DC power adapter**Power cord options (types B, E, G, and no cord options)**Additional features:**Intel Visual BIOS**Windows* 8.1 and 8 logo**Compatible with Linux***VESA mount bracket (75 x 75mm and 100 x 100mm compatible)*

Half-length PCIe mini-card slot*

Low-acoustics active cooling design

Kensington lock support

3-year advanced warranty replacement

Included in the box

Intel NUC D34010WYK Board integrated in enclosure

VESA mount bracket, with screws

19V, 65W DC-DC power adapt