TESIS - KM 185401

KOMBINASI METODE FILTER KORELASI DAN FILTER KALMAN UNTUK PELACAKAN KENDARAAN BERGERAK DENGAN PENANGANAN OKLUSI PADA VIDEO TRANSPORTASI

LULUK SARIFAH
NRP 0611 1850 010 012

DOSEN PEMBIMBING:
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.
Dr. Mahmud Yunus, M.Si.

DEPARTEMEN MATEMATIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

THESIS - KM 185401

COMBINATION OF CORRELATION FILTER AND KALMAN FILTER METHODS FOR MOVING VEHICLE TRACKING BY THE HANDLING OF OCCLUSION ON TRANSPORTATION VIDEOS

LULUK SARIFAH
NRP 0611 1850 010 012

SUPERVISORS:
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.
Dr. Mahmud Yunus, M.Si.

DEPARTMENT OF MATHEMATICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Matematika (M.Mat)
di
Institut Teknologi Sepuluh Nopember
Oleh:
LULUK SARIFAH
NRP: 06111850010012

Tanggal Ujian: 7 Juli 2020
Periode Wisuda: September 2020

Disetujui oleh:
Pembimbing:

1. Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T. .....
   NIP: 19690405 199403 2 003

2. Dr. Mahmud Yunus, M.Si. .....
   NIP: 19620407 198703 1 005

Penguji:

1. Dr. Dra. Mardlijah, M.T. .....
   NIP: 19670114 199102 2 001

2. Dr. Valeriana Lukitosari, S.Si., M.T. *Valeriana* .....
   NIP: 19710928 199802 2 001

Kepala Departemen Matematika
Fakultas Sains dan Analitika Data

Subchan, S.Si., M.Sc., Ph.D.
NIP: 19710513 199702 1 001

v

KOMBINASI METODE FILTER KORELASI DAN FILTER KALMAN UNTUK PELACAKAN KENDARAAN BERGERAK DENGAN PENANGANAN OKLUSI PADA VIDEO TRANSPORTASI

Nama Mahasiswa	:	Luluk Sarifah
NRP	:	0611 1850 010 012
Pembimbing	:	1. Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.
		2. Dr. Mahmud Yunus, M.Si.

ABSTRAK

Perkembangan teknologi pengolahan citra digital memiliki banyak manfaat salah satunya dalam bidang pelacakan objek (*object tracking*), seperti pelacakan kendaraan bergerak pada video transportasi. Pelacakan pada kendaraan bergerak memiliki tantangan, salah satunya ketika terjadi oklusi, yaitu gangguan yang menyebabkan keakuratan dari suatu pelacakan objek berkurang bahkan mengalami kegagalan. Penelitian ini bertujuan untuk menerapkan kombinasi metode Filter Korelasi dan Filter Kalman untuk melakukan pelacakan kendaraan bergerak pada video transportasi. Metode Filter Korelasi digunakan untuk mendeteksi gerakan atau perpindahan dari objek kendaraan, sedangkan metode Filter Kalman untuk mengantisipasi atau memprediksi gerakan dari objek kendaraan. Salah satu keistimewaan dari metode Filter Korelasi ialah kecepatan dan perhitungannya yang efisien dalam pelacakan visual, tetapi terkadang kurang *robust* (handal) dalam penanganan oklusi, sehingga sering mengalami perbaikan dari algoritmanya. Sedangkan metode Filter Kalman memiliki keistimewaan berupa hasil prediksi yang presisi. Persentase akurasi diperoleh dengan mendeteksi *frame* yang terlacak benar dan *frame* yang terlacak salah dari setiap *frame* video, baik video tanpa oklusi maupun video dengan oklusi yang bersifat *single object* dan *multiple objects*. Dari simulasi yang dilakukan pada keseluruhan video, diperoleh rata-rata persentase akurasi sebesar 89% dengan metode Filter Korelasi dan 94% dengan metode kombinasi pada video tanpa oklusi, sedangkan rata-rata persentase akurasi pada video dengan oklusi diperoleh sebesar 78% dengan metode Filter Korelasi dan 84.3% dengan metode kombinasi. Adapun dalam penanganan oklusi metode kombinasi mampu menangani oklusi sebagian akan tetapi belum sepenuhnya mampu menangani oklusi secara keseluruhan.

Kata kunci: Filter Kalman, Filter Korelasi, Oklusi, dan Pelacakan

COMBINATION OF CORRELATION FILTER AND KALMAN FILTER METHODS FOR MOVING VEHICLE TRACKING BY THE HANDLING OF OCCLUSION ON TRANSPORTATION VIDEOS

Name	:	Luluk Sarifah
NRP	:	0611 1850 010 012
Supervisors	:	1. Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.
		2. Dr. Mahmud Yunus, M.Si.

ABSTRACT

The development of digital image processing technology has many benefits, one of which is in the field of object tracking, such as moving vehicle tracking on transportation videos. Tracking on a moving vehicle has challenges, one of which is when occlusion occurs, which is interference that causes the accuracy of an object tracking to be reduced or even fail. This study aims to apply a combination of Correlation Filter and Kalman Filter methods for moving vehicle tracking on transportation videos. Correlation Filter method is used to detect movement or displacement of vehicle object, while Kalman Filter method is to anticipate or predict the movement of vehicle object. One of the features of Correlation Filter method is that its speed and calculation are efficient in visual tracking, but sometimes it is less robust in handling occlusion, so it often experiences improvements from the algorithm. While Kalman Filter method has the privilege of being precise prediction results. Percentage of accuracy is obtained by detecting frames that are tracked correctly and frames that are tracked incorrectly from each video frame, both videos without occlusion or videos with occlusions that are single object and multiple objects. From the simulations performed on the entire video generated an average percentage of accuracy 89% with Correlation Filter method and 94% with combination method on videos without occlusion, while the average percentage of accuracy on videos with occlusion 78% with Correlation Filter method and 84.3% with combination method. As for the handling of occlusion, the combination method can handle partial occlusion but is not yet fully able to handle overall occlusion.

Key words: Correlation Filter, Kalman Filter, Oclussion, and Tracking

KATA PENGANTAR

Dengan rahmat Allah SWT, syukur Alhamdulillah penulis dapat menyelesaikan Tesis yang berjudul:

Kombinasi Metode Filter Korelasi dan Filter Kalman untuk Pelacakan Kendaraan Bergerak dengan Penanganan Oklusi pada Video Transportasi

Tesis ini disusun sebagai salah satu syarat kelulusan Program Studi Strata 2 (S-2) Program Magister Departemen Matematika Fakultas Sains dan Analitika Data Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Terselesaikannya Tesis ini tidak terlepas dari bantuan dan dukungan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

(1). Rektor dan Wakil Rektor Institut Teknologi Sepuluh Nopember Surabaya

(2). Dekan, Kepala Departemen, dan Kepala Program Studi Pascasarjana Matematika Institut Teknologi Sepuluh Nopember Surabaya

(3). Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T. dan Bapak Dr. Mahmud Yunus, M.Si., selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, perhatian, arahan, nasehat, dan motivasi kepada penulis, sehingga penulis dapat menyelesaikan Tesis ini.

(4). Ibu Dr. Dra. Mardlijah, M.T., dan Ibu Dr. Valeriana Lukitosari, S.Si., M.Si. selaku dosen penguji yang telah banyak memberikan saran dan masukan sehingga Tesis ini dapat diselesaikan dengan baik.

(5). Ibu Prof. Dr. Erna Apriliani M.Si., selaku dosen wali yang telah membimbing dan memotivasi selama menempuh pendidikan magister.

(6). Seluruh Bapak dan Ibu dosen, staf, dan karyawan Jurusan Matematika ITS yang telah memberikan bekal ilmu pengetahuan kepada penulis dan juga atas segala bantuan, kemudahan, dan kelancaran selama penulis mengikuti proses perkuliahan.

(7). Terkhusus untuk kedua orang tua, H. Hosni dan Hj. Siti Fatimah (Tunnah) dengan segala kasih sayangnya yang selalu memberikan motivasi, saran, dukungan, dan doa kepada penulis sehingga dapat menyelesaikan Tesis ini.

(8). Adek saya Siti Aisyah, kakak-kakak saya Ikhlas, Halimatus sa'diyah, Ahmad Fauzi, Farida, Mansur dan Mailah, keponakan saya Sofiatul Jannah, Ahmad Kusairi, Sarifatus Zahroh, Safaatul Maulidia, Ahmad Robi'i, Alviatul Hasanah, Azkiyatul Humairah, Siti Alifatul Mansuroh, serta seluruh keluarga yang telah memberikan doa dan dukungan yang tak pernah henti sampai terselesaikannya Tesis ini.

(9). Mohammad Rifky Muhaemin, Bertus, Maskur, Maulidi, Koko, Sipul, Alvin, Yudi dan teman-teman lainnya yang telah membantu penulis dalam mengambil data video serta membantu dalam pengerjaan Tesis ini.

(10). Teman-teman S2 Matematika ITS khususnya angkatan 2018 – 2019 atas bantuan, persahabatan dan kenangan selama penulis menempuh pendidikan di Pascasarjana Matematika ITS.

(11). Seluruh pihak yang yang tak bisa penulis sebutkan satu per satu, atas segala saran, dukungan, dan motivasi kepada penulis dalam penyelesaian Tesis ini.

Penulis sangat menghargai kritik dan saran yang membangun dari berbagai pihak, sehingga penelitian selanjutnya diharapkan bisa lebih baik dan semoga laporan Tesis ini dapat bermanfaat bagi semua pihak, bagi kemajuan dan perkembangan ilmu pengetahuan, dan dapat berkontribusi terhadap kemajuan ITS, bangsa, dan Negara.

Surabaya, Agustus 2020

Penulis

DAFTAR ISI

		Hal
HALAMAN JUDUL		i
LEMBAR PENGESAHAN		v
ABSTRAK		vii
ABSTRACT		ix
KATA PENGANTAR		xi
DAFTAR ISI		xiii
DAFTAR GAMBAR		xv
DAFTAR TABEL		xix
BAB 1	PENDAHULUAN	1
1.1	Latar Belakang	1
1.2	Rumusan Masalah	3
1.3	Batasan Masalah	3
1.4	Tujuan Penelitian	3
1.5	Manfaat Penelitian	4
BAB 2	KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1	Penelitian-Penelitian Terkait	5
2.2	Dasar Teori	6
	2.2.1 Citra Digital	6
	2.2.2 Video Digital	7
	2.2.3 *Background Subtraction* (BS)	8
	2.2.4 *Gaussian Mixture Model* (GMM)	9
	2.2.5 Operasi Morfologi	12
	2.2.6 *Tracking*	14
	2.2.7 Oklusi	16
	2.2.8 Filter Korelasi (FK)	17
	2.2.9 Filter Kalman (FK)	20
BAB 3	METODE PENELITIAN	23
3.1	Objek Penelitian	23
3.2	Peralatan	23
3.3	Tahap Penelitian	23
3.4	Tempat Penelitian	26

BAB 4 PERANCANGAN DAN IMPLEMENTASI 27
4.1 Perancangan Proses 27
4.1.1 Perancangan Deteksi Objek 29
4.1.2 Perancangan Pelacakan Objek 37
4.2 Perancangan Perangkat Lunak 45
4.2.1 Lingkungan Perancangan 45
4.2.2 Perancangan Data 46
4.3 Implementasi Perangkat Lunak pada Java 47
4.3.1 Halaman Awal *User Interface* 48
4.3.2 Halaman Utama *User Interface* 48
4.4 Implementasi Algoritma 50
4.4.1 Implementasi *Input* Video 50
4.4.2 Implementasi Pemilihan ROI 51
4.4.3 Konversi Video menjadi Barisan *Frame* 53
4.4.4 Implementasi Deteksi Objek 54
4.4.5 Implementasi Pelacakan Objek 60

BAB 5 Simulasi dan Analisis Hasil 65
5.1 Data Simulasi 65
5.2 Akurasi Simulasi Pelacakan Objek 66
5.3 Simulasi Deteksi Objek 67
5.4 Simulasi Pelacakan Objek 73
5.4.1 Simulasi Pelacakan Objek Pada Video Tanpa Oklusi . . . 74
5.4.2 Simulasi Pelacakan Objek Pada Video dengan Oklusi . . 84
5.5 Pembahasan Hasil Pengujian 94
5.6 Pembahasan Penyebab Besar Kecilnya Kinerja 97

BAB 6 KESIMPULAN DAN SARAN 103
6.1 Kesimpulan 103
6.2 Saran 103

DAFTAR PUSTAKA 105

DAFTAR GAMBAR

Hal

Gambar 2.1 Representasi *Frame* Citra dari Suatu Video. 8
Gambar 2.2 Ilustrasi dari *Background Subtraction* 9
Gambar 2.3 Diagram Alir Metode GMM 10
Gambar 2.4 (a). Citra Asal, (b). Citra Hasil Erosi 13
Gambar 2.5 (a). Citra Asal, (b). Citra Hasil Dilasi 13
Gambar 2.6 (a). Citra Asal, (b). Citra Hasil *Opening* 14
Gambar 2.7 (a). Citra Asal, (b). Citra Hasil *Closing* 14
Gambar 2.8 Pelacakan Objek Bergerak 15
Gambar 2.9 Pengelompokkan Metode Pelacakan 16
Gambar 2.10 (a). Oklusi oleh Benda Diam, (b). Oklusi oleh Benda Bergerak 17
Gambar 2.11 Ilustrasi dari Matriks *Circulant* 18
Gambar 2.12 Siklus Diskrit Filter Kalman pada *Image Processing* . . . 20

Gambar 3.1 Diagram Tahapan Pelacakan Kendaraan Bergerak Pada Video Transportasi. 24

Gambar 4.1 Contoh Oklusi pada Video Transportasi 27
Gambar 4.2 Diagram Alir Perancangan Proses 28
Gambar 4.3 Diagram Alir Proses Deteksi Objek 29
Gambar 4.4 Contoh Matriks Intensitas *Frame* yang Terekstrak 30
Gambar 4.5 Contoh Inisialisasi Parameter GMM 31
Gambar 4.6 Matriks Selisih Intensitas *Frame Input* dan *Mean* 31
Gambar 4.7 Hasil *Update* Parameter GMM 32
Gambar 4.8 Hasil Normalisasi Bobot 32
Gambar 4.9 Hasil Perhitungan $\frac{\omega}{\sigma^2}$ 33
Gambar 4.10 *Smoothing* dengan Median Filter 34
Gambar 4.11 Proses *Shadow Removal* 35
Gambar 4.12 Proses Morfologi dengan Operasi *Closing* 36
Gambar 4.13 Diagram Alir Perancangan Pelacakan Objek 37
Gambar 4.14 Ilustrasi *Border Following* 38
Gambar 4.15 Contoh *Border Following* 39
Gambar 4.16 Ilustrasi Pelatihan Filter Korelasi 40
Gambar 4.17 Ilustrasi Lokasi Target dari Filter Korelasi. (a). *Model Update*, (b). Spektrum Model (Filter), (c) Objek dengan Nilai *Centroid* 40
Gambar 4.18 Ilustrasi *Bounding Box* dan *Centroid* 41
Gambar 4.19 Contoh *Labeling* Multi Objek 42
Gambar 4.20 Contoh Jarak Centroid Objek Antar *Frame* 42

Gambar 4.21 Diagram Alir Perancangan Pelacakan dengan Metode Filter Kalman 43
Gambar 4.22 Proses Akuisisi Data Video 46
Gambar 4.23 Halaman Utama *User Interface* 48
Gambar 4.24 Halaman Utama *User Interface* 48
Gambar 4.25 Panel *Input* dan Proses 1 49
Gambar 4.26 Panel Proses 2, Parameter, dan Identifikasi 49
Gambar 4.27 Panel Video 50
Gambar 4.28 *Source Code Input* Video 51
Gambar 4.29 Halaman Antar Muka *Input* Video 51
Gambar 4.30 *Source Code* Pemilihan ROI 52
Gambar 4.31 Tampilan Antar Muka Pemilihan ROI secara *Default* . . . 52
Gambar 4.32 Tampilan Antar Muka Pemilihan ROI Secara *Custom* . . 53
Gambar 4.33 Salah Satu Tampilan Pemilihan ROI 53
Gambar 4.34 *Source Code* Konversi Video menjadi Barisan *Frame* . . . 54
Gambar 4.35 *Source Code Prepocessing* 55
Gambar 4.36 Contoh *Frame* Hasil *prepocessing* 55
Gambar 4.37 *Source Code Background Subtraction* dengan GMM 56
Gambar 4.38 Contoh *Frame* Hasil *Background Subtraction* dengan GMM 56
Gambar 4.39 *Source Code Smoothing* 57
Gambar 4.40 Contoh *Frame* Hasil *Smoothing* 57
Gambar 4.41 *Source Code Shadow Removal* 58
Gambar 4.42 Contoh *Frame* Hasil *Shadow Removal* 58
Gambar 4.43 *Source Code* Operasi Morfologi 59
Gambar 4.44 Contoh *Frame* Hasil Operasi Morfologi 60
Gambar 4.45 *Source Code* Inisialisasi Filter Korelasi 60
Gambar 4.46 *Source Code Update Model* 61
Gambar 4.47 *Source Code* Filter Kalman 62
Gambar 4.48 Tampilan Pelacakan Objek dengan Metode FK dan FK 63

Gambar 5.1 (a). Citra Asal, (b). Citra Hasil BS tanpa *Preprocessing* 67
Gambar 5.2 Histogram dari Gambar 5.1 68
Gambar 5.3 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 69
Gambar 5.4 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 69
Gambar 5.5 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 70
Gambar 5.6 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 70
Gambar 5.7 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 71
Gambar 5.8 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 71
Gambar 5.9 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 72
Gambar 5.10 (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS 72
Gambar 5.11 Citra Hasil Deteksi Objek 73
Gambar 5.12 (a). Tidak Presisi, (b). Kurang Presisisi, dan (c). Presisi 74
Gambar 5.13 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 75

Gambar 5.14 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 76
Gambar 5.15 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 77
Gambar 5.16 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 78
Gambar 5.17 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 79
Gambar 5.18 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 80
Gambar 5.19 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 81
Gambar 5.20 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 82
Gambar 5.21 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 83
Gambar 5.22 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 84
Gambar 5.23 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 85
Gambar 5.24 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 86
Gambar 5.25 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 87
Gambar 5.26 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 88
Gambar 5.27 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 89
Gambar 5.28 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 90

Gambar 5.29 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 91
Gambar 5.30 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 92
Gambar 5.31 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 93
Gambar 5.32 (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK 94
Gambar 5.33 Tingkat Persentase Akurasi Pelacakan Objek pada Video Tanpa Oklusi 95
Gambar 5.34 Tingkat Persentase Akurasi Pelacakan Objek pada Video dengan Oklusi 95
Gambar 5.35 Pengaruh Pencahayaan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek 98
Gambar 5.36 Pengaruh Waktu Pengambilan Video pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek 98
Gambar 5.37 Pengaruh Bayangan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek 99
Gambar 5.38 Pengaruh Warna Kendaraan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek . . . 99
Gambar 5.39 Pengaruh Kepadatan Arus Lalu Lintas pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek 100
Gambar 5.40 Pengaruh Kecepatan Kendaraan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek 100
Gambar 5.41 Pengaruh Sudut Pengambilan Video pada Kinerja Pelacakan objek. (a). Segmentasi Objek, (b). Pelacakan Objek 101

DAFTAR TABEL

Hal

Tabel 4.1 Perangkat yang Digunakan 45
Tabel 4.2 Data Proses 47
Tabel 4.3 *Parameter Region of Interest* (ROI) 53
Tabel 4.4 Parameter Proses *Prepocessing* 55
Tabel 4.5 Parameter Proses *Background Subtraction* dengan GMM . 56
Tabel 4.6 Parameter Proses *Smoothing* dengan Median Filter 57
Tabel 4.7 Parameter Proses Morfologi dengan Operasi *Closing* 59

Tabel 5.1 Dataset Video Simulasi Pelacakan Objek Tanpa Oklusi . . . 66
Tabel 5.2 Dataset Video Simulasi Pelacakan Objek dengan Adanya Oklusi 66
Tabel 5.3 Rangkaian Proses Deteksi Objek 73
Tabel 5.4 Hasil Kinerja Pelacakan Objek 75
Tabel 5.5 Hasil Kinerja Pelacakan Objek 76
Tabel 5.6 Hasil Kinerja Pelacakan Objek 77
Tabel 5.7 Hasil Kinerja Pelacakan Objek 78
Tabel 5.8 Hasil Kinerja Pelacakan Objek 79
Tabel 5.9 Hasil Kinerja Pelacakan Objek 80
Tabel 5.10 Hasil Kinerja Pelacakan Objek 81
Tabel 5.11 Hasil Kinerja Pelacakan Objek 82
Tabel 5.12 Hasil Kinerja Pelacakan Objek 83
Tabel 5.13 Hasil Kinerja Pelacakan Objek 84
Tabel 5.14 Hasil Kinerja Pelacakan Objek 85
Tabel 5.15 Hasil Kinerja Pelacakan Objek 86
Tabel 5.16 Hasil Kinerja Pelacakan Objek 87
Tabel 5.17 Hasil Kinerja Pelacakan Objek 88
Tabel 5.18 Hasil Kinerja Pelacakan Objek 89
Tabel 5.19 Hasil Kinerja Pelacakan Objek 90
Tabel 5.20 Hasil Kinerja Pelacakan Objek 91
Tabel 5.21 Hasil Kinerja Pelacakan Objek 92
Tabel 5.22 Hasil Kinerja Pelacakan Objek 93
Tabel 5.23 Hasil Kinerja Pelacakan Objek 94
Tabel 5.24 Rata-Rata Persentase Akurasi dengan Metode Filter Korelasi 96
Tabel 5.25 Rata-Rata Persentase Akurasi dengan Metode Kombinasi . 96
Tabel 5.26 Rata-Rata Persentase Akurasi dengan Metode Filter Korelasi 97
Tabel 5.27 Rata-Rata Persentase Akurasi dengan Metode Kombinasi . 97

BAB 1
PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang yang mendasari ide dari penelitian. Selain itu, dirumuskan juga masalah-masalah yang berkaitan dengan penelitian serta batasan-batasan masalahnya. Setiap penelitian memiliki tujuan dan juga bermaksud memberi manfaat kepada masyarakat luas baik itu berupa konstribusi keilmuan maupun manfaat langsung. Sehingga, pada bab ini juga dijelaskan maksud tujuan serta kebermanfaatan penelitian.

1.1 Latar Belakang

Perkembangan teknologi pengelohan citra digital memiliki banyak manfaat salah satunya dalam bidang pelacakan objek (*object tracking*). Terlebih lagi objek yang diteliti adalah kendaraan bergerak pada video transportasi, yang juga berguna untuk pengembangan *Smart city*. Salah satu unsur pendukung dari pengembangan *Smart city* adalah *Intelligent Transportation System* (ITS) yang didalamnya juga mencakup analisis kepadatan arus lalu lintas. Berdasarkan informasi dari Badan Pusat Statistik (BPS) Indonesia yaitu jumlah kendaraan bermotor di Indonesia pada tahun 2013-2017 mengalami kenaikan sebesar $7,40\%$, hal ini sejalan dengan bertambahnya jumlah penduduk di suatu daerah sehingga kebutuhan akan transportasi pun semakin meningkat (Badan Pusat Statisitik, 2018). Adanya penambahan jumlah kendaraan bermotor ini membutuhkan analisis kepadatan lalu lintas untuk pemantauan arus lalu lintas. Salah satu teknologi untuk pemantauan arus lalu lintas ialah kamera pengawas atau CCTV (*Closed Circuit Television*), akan tetapi masih belum sempurna tanpa dilengkapi dengan deteksi dan pelacakan objek secara otomatis untuk mengetahui pergerakan objek.

Pelacakan (*tracking*) objek merupakan salah satu topik dalam *computer vision*, dimana kata *tracking* menurut kamus Inggris-Indonesia memiliki arti mengikuti jalan sehingga bisa diartikan bahwa *tracking* adalah proses mengikuti posisi dari suatu benda atau objek yang akan diamati. Pelacakan objek dapat bersifat *single object* maupun *multiple objects*, tergantung objek yang akan diteliti (Rizky, 2017). Terdapat tiga langkah mendasar dalam pelacakan objek yaitu pemilihan objek bergerak pada suatu video, pelacakan objek yang telah dipilih dari langkah pertama dari beberapa bagian *frame*, dan menganalisis pergerakan objek untuk mengenali perilaku objek tersebut (Yilmaz, dkk, 2006). Namun dalam perakteknya pelacakan objek terutama *multiple objects* sangat terhambat oleh gangguan dan mungkin mengalami kegagalan dikarenakan adanya oklusi.

Oklusi diartikan antara satu objek dengan objek lainnya berhimpitan sehingga *bounding box* mengecil atau objek menjadi *intermittent*. Oklusi terjadi ketika objek yang dilacak atau artribut penting yang digunakan untuk mengenali identitas dari objek tersebut tidak tersedia pada kamera untuk melacak ketika objek tersebut masih berada pada jangkauan kamera (Lee, dkk, 2014). Adanya oklusi ini mengakibatkan berkurangnya akurasi dalam pelacakan, sehingga dalam pengolahan citra video, oklusi dianggap sebagai salah satu rintangan yang menantang dalam pelacakan objek. Jenis oklusi yang ditemukan dalam video transportasi yaitu kendaraan yang tertutup pohon, semak-semak, rambu lalu lintas, lampu lalu lintas, tiang lampu dan lain sebagainya, serta kendaraan yang saling berhimpitan atau terhalang oleh kendaraan lainnya. Oleh karena itu pada penelitian ini akan menangani masalah oklusi tersebut yang mengacu pada tiga fitur utama dalam pelacakan objek yakni: bentuk, warna, dan gerak.

Beberapa penelitian terkait dengan pelacakan objek yaitu penelitian Zheng, dkk (2012) yang dapat melacak objek yang ditargetkan meskipun objek tersebut terhalang oleh benda lainnya, tetapi memerlukan waktu yang lama akibat dari *update template* pada saat hilangnya fitur, serta bersifat *single object*. Dalam Tesis di Institut Teknologi Sepuluh Nopember oleh Galandaru Swalaganata (2017), dimana dalam penelitian tersebut berhasil mengkombinasikan metode Camshift dan Filter Kalman untuk pelacakan objek, sehingga membuat hasil pelacakan objek semakin presisi, akan tetapi penelitian tersebut bersifat *single object* dan tanpa adanya oklusi. Fengfa Yue dan Xingfei Li (2017) dalam penelitiannya Filter Korelasi mengalami perbaikan untuk meningkatkan hasil akurasi menggunakan Filter Kalman dan pendekatan *threshold*. Penelitian tersebut hanya fokus pada *single object tracking*. Ying Wang (2018) dalam penelitiannya pelacakan objek menggunakan kombinasi metode GMM (*Gaussian Mixture Model*) dan Filter Kalman serta objek penelitian adalah kendaraan bergerak yang bersifat *multiple objects*, tetapi belum melakukan penanganan oklusi. Bin Zhou dan Tuo Wang (2019) dalam penelitiannya Filter Korelasi ditingkatkan dengan adaptive context-Aware dan structural Correlation filter, menyatakan bahwa hasil lebih baik dari pada Filter Korelasi konvensional terutama saat terjadi oklusi, akan tetapi penelitian tersebut adalah pelacakan satu objek.

Berdasarkan uraian dan analisis yang diperoleh dari hasil penelitian terdahulu yang disebut di atas, dalam kaitannya dengan penelitian guna menyelesaikan tantangan, maka dalam Tesis ini dibahas mengenai kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak. Metode Filter Korelasi digunakan untuk mendeteksi gerak atau perpindahan objek sedangkan untuk metode Filter Kalman digunakan untuk mengantisipasi atau memprediksi gerak dari objek. Penelitian ini bekerja dengan data masukan *non-real time* berupa video dari objek kendaraan bergerak yang telah direkam sebelumnya. Adapun untuk deteksi objek dilakukan dengan cara *preprocessing*, segmentasi dengan

Background Subtraction, dimana untuk menginisialisasi *background* modelnya menggunakan metode GMM, *smoothing* dengan Median Filter, *shadow removal* dan dilanjutkan dengan operasi morfologi. Keseluruhan proses dan perancangan perangkat lunak diimplementasikan menggunakan bahasa pemrograman JAVA, Open CV sebagai *library* dengan *Software tool* Netbeans 8.2.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, dapat dirumuskan suatu permasalahan yang menjadi fokus utama dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan dari kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dengan penanganan oklusi?.

2. Bagaimana kinerja dari kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dengan penanganan oklusi, melalui simulasi pada data video transportasi?.

1.3 Batasan Masalah

Untuk menghasilkan suatu hasil yang efektif, maka batasan masalah yang digunakan adalah sebagai berikut:

1. Data yang digunakan adalah video yang telah tersimpan dalam penyimpanan komputer.

2. Objek yang diamati adalah kendaraan yang bergerak di jalan raya dengan jangkauan pandangan seluruh jalan.

3. Pelacakan (*tracking*) dan deteksi objek dilakukan pada objek yang berada pada ROI (*region of interest*).

4. Simulasi dilakukan secara *offline* dan *non-real time*

5. Simulasi yang dilakukan hanya pada masalah deteksi objek dan pelacakan objek, tidak sampai pada otomasi *counting* dan klasifikasi kendaraan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang diberikan, tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan kombinasi dari metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dengan penanganan oklusi.

2. Menganalisis kinerja dari kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dengan penanganan oklusi, melalui simulasi pada data video transportasi.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian yang diusulkan, maka manfaat yang ingin diperoleh yaitu diharapkan dapat membantu penelitian dalam bidang ilmu komputer seperti *object tracking* serta dihasilkan suatu kajian yang bisa digunakan untuk penelitian-penelitian yang sejenis dan dapat dijadikan bahan penelitian lebih lanjut seperti integrasi ke sistem klasifikasi jenis kendaraan dan sistem penghitung jumlah kendaraan untuk otomasi sistem pemantauan cerdas pada lalu lintas. Selain itu, berkontribusi dalam mengetahui kinerja dari kombinasi metode Filter Korelasi dan Filter Kalman.

BAB 2
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini dijelaskan mengenai kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam penelitian, meliputi penelitian sebelumnya dan beberapa teori pendukung yang akan dijelaskan pada sub-bab selanjutnya.

2.1 Penelitian-Penelitian Terkait

Penelitian-penelitian terkait yang pernah dilakukan sebelumnya adalah sebagai berikut:

(1). Zheng, B., Xu, X., Dai, Y., dan Lu, Y. (2012), dalam penelitian tersebut mengkombinasikan *Dynamic Template Matching* dan Filter Kalman. Pendeteksian objek dilakukan dengan cara mendapatkan area dari daerah pelacakan objek dengan menggunakan metode *Inter Frame Difference* dan mengambil fitur SIFT (*Scale Invariant Feature*) pada objek tersebut. Hasil dari sistem dapat digunakan untuk melacak objek yang ditargetkan dan dapat tetap melacak objek meskipun objek tersebut terhalang oleh benda lainnya. Tetapi pada sistem tersebut memerlukan waktu yang lama akibat dari *update template* pada saat hilangnya fitur serta penelitian tersebut bersifat *single object*.

(2). Swalaganata, Galandaru (2017), dalam penelitian tersebut berhasil mengkombinasikan metode Camshift dan Filter Kalman untuk pelacakan objek. Pelacakan dengan kombinasi Camshift dan Filter Kalman menghasilkan rata-rata pelacakan presisi keseluruhan video sebesar 73% tanpa proses super-resolusi dan 93% dengan super-resolusi, sehingga penambahan proses super-resolusi dan Filter Kalman membuat hasil pelacakan objek semakin presisi, akan tetapi penelitian tersebut bersifat *single object* dan tanpa adanya oklusi.

(3). Yue, F., dan Li, X. (2017), dalam penelitian tersebut ketika target bergerak cepat, maka terjadi perubahan skala dan terjadi oklusi, sehingga untuk meningkatkan akurasi pelacakan, sebuah peningkatan pendekatan Kernel Filter Korelasi dikembangkan untuk mendapatkan pelacakan yang tepat, dan untuk *bounding box* akan menggunakan pendekatan *adaptive threshold*, sehingga tujuan dari penelitian tersebut adalah mengintegrasikan ciri dari pendekatan *threshold*, Kernel Filter Korelasi, dan Filter Kalman. Penelitian tersebut hanya fokus pada objek tunggal.

(4). Wang, Ying (2018), dalam penelitian tersebut pelacakan objek dilakukan pada kendaraan bergerak yang bersifat *multiple objects* dan dalam

menyesuaikan posisi objek menggunakan *Gaussian Mixture Model* (GMM) kemudian Filter Kalman untuk mengestimasi posisi objek pada *frame* selanjutnya, akan tetapi penelitian tersebut belum melakukan pelacakan pada kendaraan yang mengalami oklusi.

(5). Zhou, B. dan Wang, T. (2019), dalam penelitian tersebut Filter Korelasi ditingkatkan dengan adaptive context-aware dan structural Correlation Filter, diperoleh hasil bahwa, hasil dari adaptive context-Aware dan structural Correlation Filter lebih baik dari pada Filter Korelasi konvensional terutama saat terjadi oklusi akan tetapi penelitian tersebut adalah pelacakan visual (satu objek)

Berdasarkan beberapa penelitian yang telah dilakukan tersebut, maka pada penelitian ini dibahas mengenai kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dengan penanganan oklusi pada video transportasi. Metode Filter Korelasi digunakan untuk mendeteksi gerak atau perpindahan dari objek kendaraan, sedangkan metode Filter Kalman untuk mengantisipasi atau memprediksi gerakan pada objek kendaraan.

2.2 Dasar Teori

Berikut dijelaskan mengenai teori-teori pendukung dalam penelitian ini, yaitu sebagai berikut:

2.2.1 Citra Digital

Secara matematis citra digital merupakan fungsi dua variabel yang dapat dinyatakan dengan $f(x, y)$, dimana x dan y merupakan titik koordinat spasial dan amplitudo dari fungsi f pada sembarang koordinat (x, y) disebut *intensity* (intensitas) atau *gray level* (level keabuan), yang merupakan representasi dari warna cahaya yang ada pada citra analog. Untuk citra digital, nilai x dan y merupakan bilangan bulat tak negatif, dan nilai $f(x, y)$ selalu terbatas.

Citra terbagi menjadi dua yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra diam adalah citra tunggal yang tidak bergerak, sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun (*sequential*) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Pada awalnya citra yang dikenal manusia berbentuk citra kontinu, yaitu suatu representasi objek yang dihasilkan dari sistem optik yang menerima sinyal analog dan dinyatakan pada bidang dua dimensi. Nilai cahaya yang ditransmisikan pada citra kontinu memiliki rentang nilai yang tak berhingga. Contoh dari citra kontinu adalah citra yang dihasilkan oleh mata manusia dan kamera analog. Suatu citra kontinu tidak dapat direpresentasikan secara langsung oleh komputer. Oleh karena itu dilakukan proses untuk mengubah nilai-nilai yang ada pada citra kontinu agar komputer dapat membaca dan menerjemahkan informasi yang terdapat pada citra kontinu yang disebut sebagai digitalisasi. Hasil dari pemrosesan tersebut dinamakan sebagai citra digital.

Adapun proses digitalisasi terdiri dari dua tahap yaitu sebagai berikut (Gonzalez dan Woods, 2008):

(a). Digitalisasi Spasial (Sampling)

Sampling merupakan proses pengambilan informasi dari citra kontinu yang memiliki panjang dan lebar tertentu untuk membaginya ke beberapa blok kecil. Blok-blok tersebut disebut sebagai piksel, sehingga citra digital dapat dinyatakan dalam bentuk matriks yang memiliki ukuran $M \times N$, dengan kata lain citra digital yang memiliki $M \times N$ buah piksel. Pada persamaan 2.1 merupakan representasi citra digital dalam bentuk matriks.

$$\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & \cdots & \cdots & f(1,N-1) \\ \cdots & \cdots & \cdots & \cdots \\ f(M-1,0) & f(M-1,1) & \ldots & f(M-1,N-1) \end{bmatrix} \tag{2.1}$$

atau

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & \cdots & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{M-1,0} & a_{M-1,1} & \ldots & a_{M-1,N-1} \end{bmatrix}$$

Secara matematis fungsi $f(x, y)$ dapat ditulis sebagai berikut:

$$\begin{gathered} 0 \leq x \leq M-1 \\ 0 \leq y \leq N-1 \\ 0 \leq f(x,y) \leq G-1 \end{gathered}$$

dimana:
M = banyaknya baris pada array citra
N = banyaknya kolom pada array citra
G = banyaknya skala keabuan (*gray level*)

(b). Digitalisasi Intensitas (Kuantisasi)

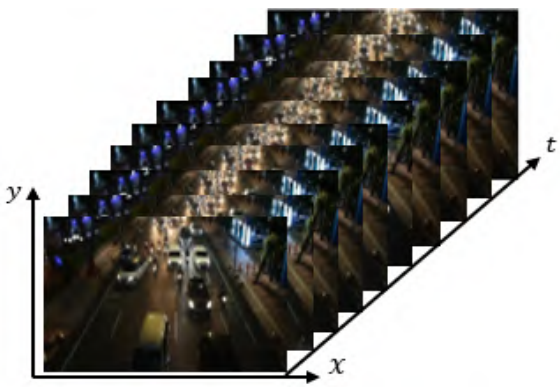
Kuantisasi merupakan proses pemberian nilai derajat keabuan di setiap titik piksel yang merupakan representasi dari warna asli dari citra kontinu. Rentang nilai keabuan adalah $0 - 255$.

2.2.2 Video Digital

Video digital adalah teknologi merekam, menangkap, memproses, menyimpan dan merekonstruksi urutan beberapa citra. Menurut Alan. C bovic dalam bukunya yang berjudul *Handbook of Image and Video Processing* menjelaskan bahwa video digital merupakan hasil sampling dan kuantisasi dari video analog (Bovik, 2000). Secara mendasar tidak ada perbedaan proses sampling dan kuantisasi antara citra digital dan video digital, dimana

secara visual, video adalah sederetan *frame* citra (koordinat spasial 2D) yang terekam oleh sensor dalam satu satuan waktu tertentu (koordinat temporal). Normalnya, kecepatan perekaman video adalah antara 25 hingga 30 *frame* per detik (FPS). Kecepatan ini disesuaikan dengan kecepatan normal visual mata manusia dalam menganalisis dan memahami pergerakan informasi yang terkandung dalam sederetan citra tersebut.

Secara matematis, video digital merupakan fungsi $f(x, y, t)$, dengan (x, y) adalah nilai diskrit atau titik koordinat spasial dari fungsi f, t menyatakan *frame* citra pada waktu t, sedangkan f menyatakan nilai intensitas warna piksel pada koordinat (x, y) dari *frame* citra pada waktu t. Video digital merupakan representasi informasi berdasarkan sekumpulan citra yang diakuisisi dan ditampilkan sesuai dengan standar *scanning system* (pemisahan video ke beberapa unit *frame* citra), *frame rate*, dan *frame size* yang digunakan oleh teknologi video tersebut (Sarifuddin, 2016). Berikut tampilan dari *frame* citra yang dapat membentuk suatu video yang dapat dilihat pada Gambar 2.1.



Gambar 2.1: Representasi *Frame* Citra dari Suatu Video.

2.2.3 *Background Subtraction* (BS)

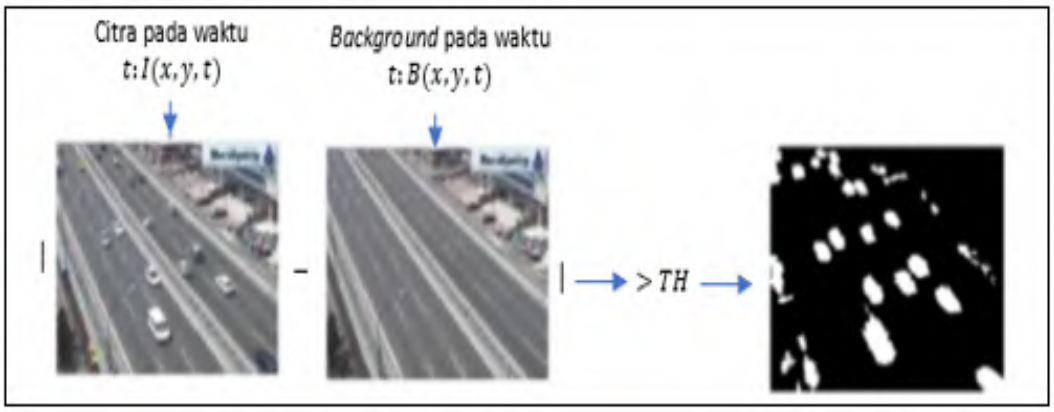
Background Subtraction (BS) digunakan untuk mendapatkan objek yang bergerak (*foreground*) pada serangkaian citra. Objek yang bergerak dapat diidentifikasi dengan melakukan pengurangan antara *frame* pada waktu t dengan *background* suatu video. Nilai citra hasil pengurangan tersebut kemudian dibandingkan dengan nilai ambang batas (*threshold* tertentu) sesuai dengan metode yang dipakai untuk membangun *foreground image* yang disebut dengan proses *differencing*. Secara matematis proses *Background Subtraction* dapat ditulis sebagai berikut:

$$|I(x, y, t) - B(x, y, t)| = d(x, y, t) \tag{2.2}$$

$$f(x, y, t) = \begin{cases} 1, & jika \quad d(x, y, t) \geq \tau \\ 0, & jika \quad d(x, y, t) < \tau \end{cases} \tag{2.3}$$

dengan $I(x, y, t)$ adalah citra pada waktu t, $B(x, y, t)$ adalah *background* model pada waktu t, $d(x, y, t)$ adalah citra hasil pengurangan I dan B pada waktu

t, $f(x, y, t)$ adalah *foreground image* (gerak yang dideteksi) dan τ adalah *threshold*. Berikut ilustrasi dari *Background Subtraction* yang dapat dilihat pada Gambar 2.2.



Gambar 2.2: Ilustrasi dari *Background Subtraction* (Amaluddin, dkk, 2011).

Terdapat berbagai macam metode untuk menginisialisasi *background* model, salah satunya adalah metode *Gaussian Mixture Model* (GMM).

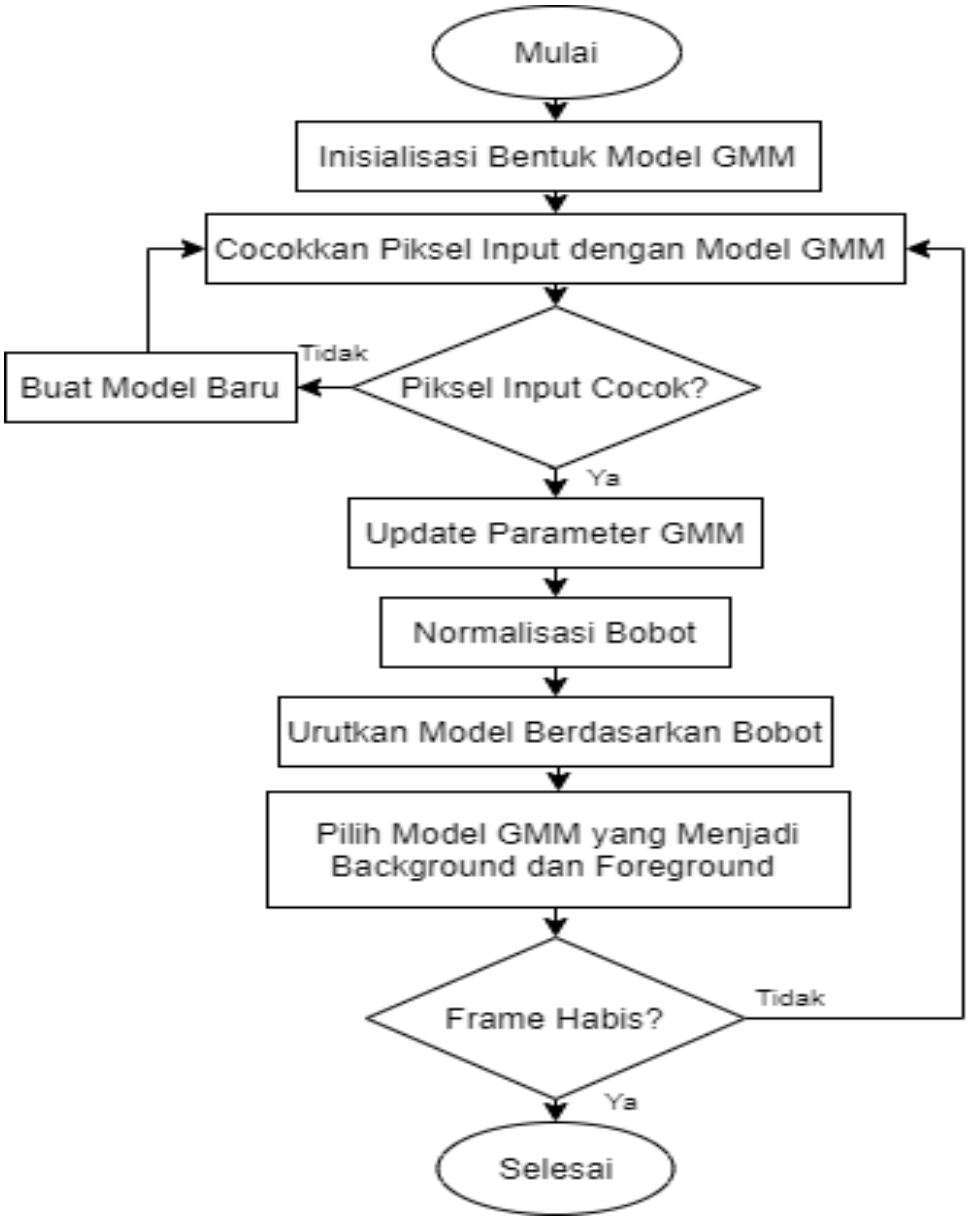
2.2.4 *Gaussian Mixture Model* (GMM)

Gaussian Mixture Model (GMM) adalah salah satu metode dari *Background Subtraction* untuk menginisialisasi *background* model. GMM merupakan tipe *density model* yang terdiri dari komponen fungsi-fungsi Gaussian. Komponen fungsi tersebut terdiri dari *weight* yang berbeda untuk menghasilkan *multi-model density*. Model-model GMM terbentuk dari data warna piksel berdasarkan waktu. Untuk sebarang waktu t, citra tersebut dimodelkan sebagai berikut (Stauffer dan Grimson, 1999):

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \tag{2.4}$$

dengan X_t adalah citra skalar maupun vektor, dan I adalah rangkaian citra. Hasil model tersebut akan dibagi 2 bagian, yaitu model yang mencerminkan *background* dan model *non-background*, dengan nilai *threshold* dari *background* yang sudah di tentukan. Jumlah model GMM yang digunakan mempengaruhi jumlah model *background*. Semakin besar jumlah model GMM yang dipakai semakin banyak model *background* yang dimiliki suatu piksel. GMM memproses tiap piksel pada citra, baik citra berupa skalar (citra *grayscale*) maupun vektor (citra berwarna). Berikut prosedur dari metode *Gaussian Mixture Model* (GMM) yang dapat dilihat pada Gambar 2.3

.



Gambar 2.3: Diagram Alir Metode GMM

Berikut tahapan dalam pemilihan distribusi yang mencerminkan *background* (Amaluddin, dkk, 2015):

(a) Pencocokan *input* terhadap distribusi
Pada tahap ini *input* dicocokkan dengan semua distribusi sampai ditemukan distribusi yang paling cocok. Suatu piksel dikatakan masuk dalam suatu distribusi jika nilai piksel tersebut masuk dalam jarak 2.5 standar deviasi dari sebuah distribusi. Untuk pencocokan *input* digunakan pertidaksamaan 2.5.

$$\mu_{i,j,k} - 2.5 * \sigma_{i,j,k} < X_{i,j,t} < \mu_{i,j,k} + 2.5 * \sigma_{i,j,k} \tag{2.5}$$

dimana X_t adalah nilai intensitas dari suatu piksel (i, j) pada *frame* ke-t, μ_k adalah nilai *mean* pada piksel (i, j) dari Gaussian ke-k, dan σ_k sebagai standar deviasi pada piksel (i, j) dari Gaussian ke-k. Apabila piksel tidak cocok dengan semua distribusi yang ada maka piksel tersebut dianggap sebagai *foreground* dan dibuat suatu distribusi baru dengan menggantikan distribusi yang paling tidak mencerminkan *background*. Distribusi baru memiliki nilai *mean* sesuai dengan nilai piksel, nilai varians yang tinggi, dan nilai *weight* yang kecil, dimana nilai awal yang diberikan pada variabel tersebut sangat mempengaruhi performa dari algoritma GMM.

(b) *Update* parameter
Pada tahap ini dilakukan *update* terhadap nilai dari parameter-parameter GMM yang nantinya digunakan untuk mengolah *input* selanjutnya. Nilai yang di-*update* terdiri dari *weight, means*, dan *standard deviation*. Nilai *weight* di-*update* menggunakan persamaan 2.6 dan di-*update* tiap waktu. Nilai *means* di-*update* menggunakan persamaan 2.8 dengan ρ yang diberikan pada persamaan 2.7. Nilai *standard deviation* di-*update* menggunakan persamaan 2.9. Setiap persamaan tersebut berlaku pada setiap piksel (i, j) (Stauffer dan Grimson, 1999).

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2.6}$$

$$\rho = \frac{\alpha}{\omega_{k,t}} \tag{2.7}$$

$$\mu_{k,t} = (1 - \rho)\mu_{k,t-1} + \rho X_t \tag{2.8}$$

$$\sigma^2_{k,t} = (1 - \rho)\sigma^2_{k,t-1} + \rho(X_t - \mu_{k,t})(X_t - \mu_{k,t}) \tag{2.9}$$

dengan $\omega_{k,t}$ adalah bobot dari Gaussian ke-k pada *frame* t, $\mu_{k,t}$ adalah *mean* dari Gaussian ke-k pada *frame* t, $\sigma_{k,t}$ adalah standar deviasi dari Gaussian ke-k pada *frame* ke-t, α adalah *learning rate* dan nilai $M_{k,t}$

adalah 1 untuk model yang cocok dan 0 untuk model yang tidak cocok. Setelah bobot di-*update* dilakukan normalisasi sehingga total bobot dari semua distribusi tepat 1. Sementara *mean* dan standar deviasi di-*update* hanya jika ada nilai piksel yang cocok dengan distribusi tersebut.

(c) Pemilihan Distribusi *Background*
Pada tahap ini dipilih model-model yang mencerminkan *background*. Pertama model-model diurutkan berdasarkan $\frac{\omega}{\sigma^2}$ sehingga distribusi yang paling mencerminkan *background* tetap di atas dan yang tidak mencerminkan *background* ada di bawah yang nantinya digantikan oleh distribusi yang lain. Untuk memilih B distribusi pertama yang dijadikan distribusi *background* digunakan persamaan 2.10, dengan T adalah nilai ambang batas yang telah ditentukan sebelumnya.

$$B = \arg\min_{b}(\sum_{k=1}^{b} \omega_k > T) \tag{2.10}$$

2.2.5 Operasi Morfologi

Operasi morfologi dalam citra digital adalah suatu *tool* untuk ekstraksi komponen *image* yang berguna dalam representasi dan deskripsi dari bentuk daerah (*region shape*) dengan *structuring element* (SE) untuk menentukan *properties of interest* dari *image*. Berikut operasi fundamental dalam operasi morfologi yaitu:

(a). Erosi
Secara matematis, jika A dan B adalah himpunan bagian Z^2, maka erosi dari A oleh B, dinotasikan dengan $A \ominus B$ dan didefinisikan sebagai (Gonzalez dan Woods, 2008):

$$A \ominus B = \{z|(B)_z \subseteq A\} \tag{2.11}$$

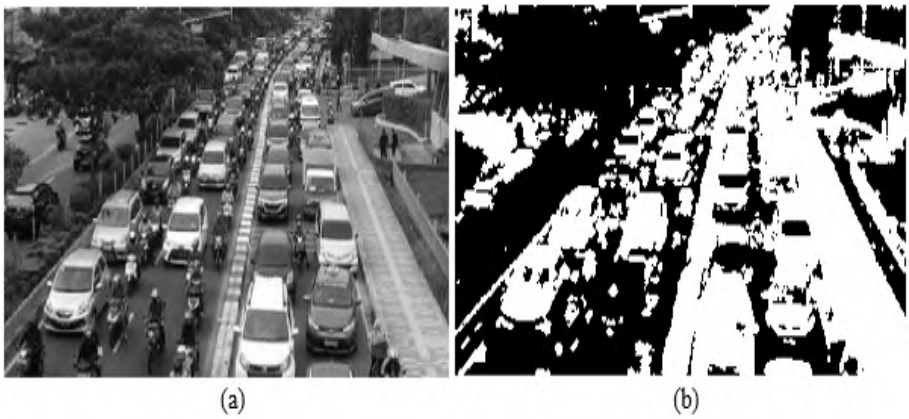
Erosi dari A oleh B adalah himpunan semua titik z sedemikian hingga semua titik z pada B termuat didalam A. Dengan kata lain, A diasumsikan sebagai citra yang akan diproses dan B adalah satu himpunan piksel yang juga disebut *structuring element*. Erosi berguna untuk menyempitkan atau menipiskan objek pada *image* biner. Berikut contoh dari Erosi yang dapat dilihat pada Gambar 2.4.

Gambar 2.4: (a). Citra Asal, (b). Citra Hasil Erosi

(b). Dilasi
Dilasi berguna untuk memperluas atau menebalkan objek pada *image* biner. Jika A dan B adalah himpunan bagian dari Z^2, dilasi antara A dan B dinyatakan $A \oplus B$ dan didefinisikan dengan (Gonzalez dan Woods, 2008):

$$A \oplus B = \{z|(\hat{B})_z \cap A \neq \varnothing\} \tag{2.12}$$

Berikut contoh dari Dilasi yang dapat dilihat pada Gambar 2.5.

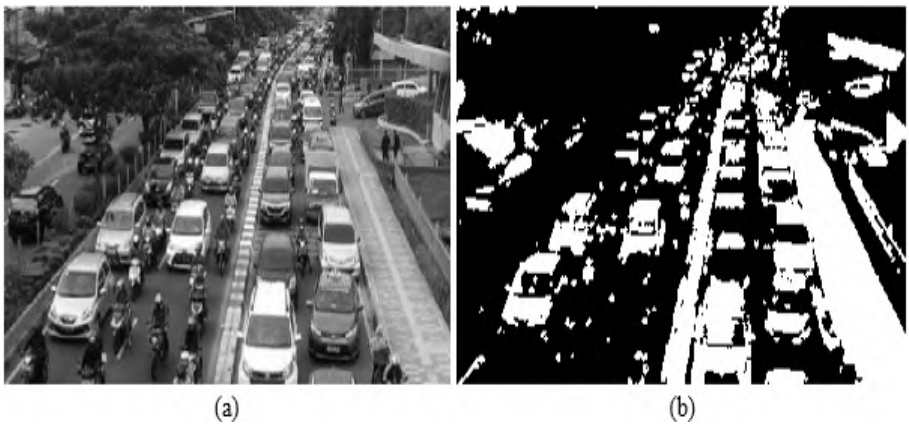


Gambar 2.5: (a). Citra Asal, (b). Citra Hasil Dilasi

Selain dua operasi fundamental Erosi dan Dilasi, terdapat dua operasi lain yang merupakan kombinasi dari dua operasi fundamental tersebut, yaitu *Opening* dan *Closing*. Operasi *Opening* adalah operasi yang melakukan proses Erosi terlebih dahulu kemudian hasilnya akan dilakukan Dilasi yang dinotasikan oleh $A \circ B$ dan didefinisikan sebagai berikut:

$$A \circ B = (A \ominus B) \oplus B \tag{2.13}$$

Operasi *Opening* secara umum digunakan untuk memperhalus kontur objek, menghilangkan gap tipis dan tonjolan pada objek. Berikut contoh dari operasi *Opening* yang dapat dilihat pada Gambar 2.6.



Gambar 2.6: (a). Citra Asal, (b). Citra Hasil *Opening*

Adapun operasi *Closing* adalah operasi yang melakukan proses Dilasi terlebih dahulu kemudian hasilnya akan dilakukan operasi Erosi yang dinotasikan oleh $A \bullet B$ dan didefinisikan sebagai berikut:

$$A \bullet B = (A \oplus B) \ominus B \tag{2.14}$$

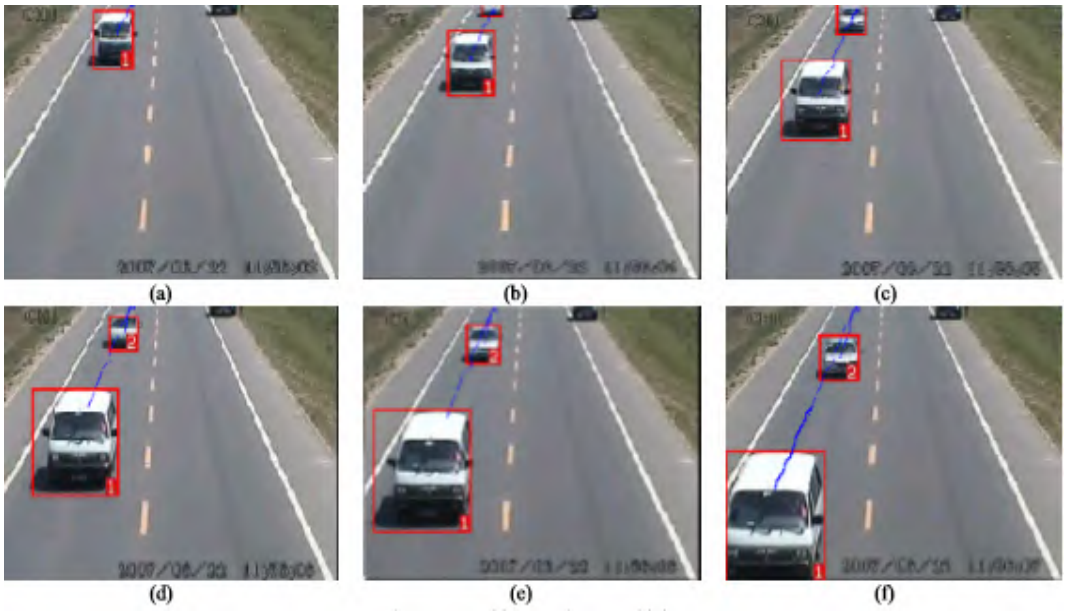
Operasi *Closing* digunakan untuk menghubungkan gap tipis, menghilangkan lubang kecil (*small holes*), dan mengisi gap pada kontur objek. Berikut contoh operasi *Closing* yang dapat dilihat pada Gambar 2.7.

Gambar 2.7: (a). Citra Asal, (b). Citra Hasil *Closing*

2.2.6 *Tracking*

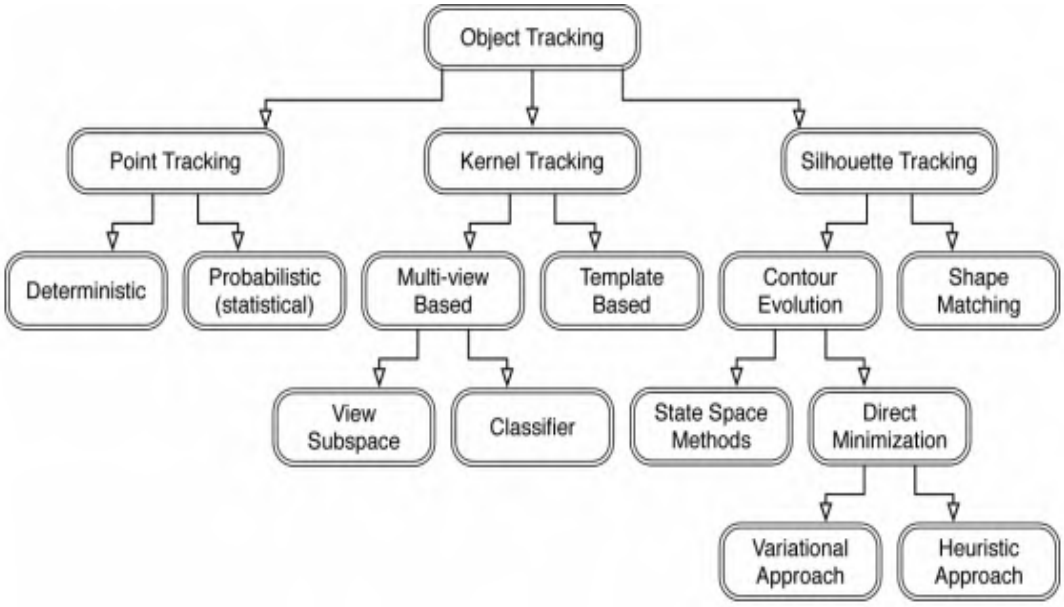
Tracking atau pelacakan secara harfiah adalah mengikuti jalan atau dalam arti bebasnya merupakan proses mencari objek bergerak dalam urutan *frame*. Pelacakan dapat dilakukan dengan menggunakan ekstraksi

ciri benda dan mendeteksi objek atau benda bergerak di urutan *frame*. Dengan menggunakan nilai posisi objek di setiap *frame*, bisa digunakan untuk menghitung posisi dan kecepatan objek bergerak (Cabido, dkk, 2012). Terdapat tiga langkah mendasar dalam pelacakan objek bergerak, yaitu pemilihan objek bergerak yang diinginkan, pelacakan objek yang telah dipilih, dan analisis pergerakan objek (Yilmaz, dkk, 2006). Pelacakan objek bergerak dibutuhkan dalam banyak aplikasi visual, diantaranya adalah pengaturan lalu lintas, *robot vision*, pengembangan militer, *sport analysis* dan sistem pengawasan atau pengamanan pada wilayah tertentu, seperti tempat parkir, mall, dan lain-lain (Rizky, 2017). Berikut contoh dari pelacakan Objek bergerak yang dapat dilihat pada Gambar 2.8



Gambar 2.8: Pelacakan Objek Bergerak
(Bukey, dkk, 2017).

Pada pelacakan objek bergerak terdapat dua pendekatan. Pendekatan pertama *tracking-by-detection* yaitu melakukan pelacakan objek dengan cara mendeteksi objek pada setiap *frame* video yang diamati, guna menentukan posisi objek di setiap *frame*nya. Pendekatan kedua *detection-by-tracking* yaitu melakukan pelacakan objek dengan cara mendeteksi objek hanya pada satu atau beberapa *frame* pertama saja, lalu untuk mengetahui gerak atau posisi objek pada *frame-frame* selanjutnya cukup melakukan pelacakan perubahan gerak antara *frame* saat ini terhadap *frame-frame* sebelumnya. Pendekatan pertama tidak dianjurkan selain dikarenakan mahalnya biaya komputasi untuk mendeteksi keberadaan suatu objek pada setiap *frame*-nya, pendekatan ini juga tidak memanfaatkan informasi yang sudah didapat pada *frame-frame* sebelumnya yang dapat digunakan untuk menghitung probabilitas kemunculan atau posisi suatu objek saat ini. Adapun pengelompokkan tentang pembagian pelacakan objek dapat dilihat pada Gambar 2.9.



Gambar 2.9: Pengelompokkan Metode Pelacakan
(Yilmaz dan Javed, 2006).

dari Gambar 2.9 menjelaskan bahwa pelacakan *point tracking* adalah objek terdeteksi di *frame* secara berurutan yang diwakili oleh titik. Titik yang digunakan sebagai tanda objek saling berhubungan dari *frame* pertama sampai *frame* terakhir yang membentuk suatu lintasan pelacakan. Titik-titik tersebut mewakili posisi objek dan pergerakan objek. Adapun MGE *tracker* dan GOA *tracker* merupakan contoh metode deterministik, sedangkan Filter Kalman, JPDAF dan PMHT merupakan metode statistikal. Kernel *tracking* mengacu pada bentuk objek dan tampilan visual objek. Misalnya, kernel dapat berupa persegi panjang atau bentuk elips yang diletakkan pada objek dengan histogram tertentu. Mean-shift, Camshift, dan KLT merupakan contoh metode berdasarkan *template*, sedangkan SVM *tracker* dan *Eigentracking* merupakan contoh metode berdasarkan *multi-view*. *Silhoutte tracking* dilakukan dengan memperkirakan wilayah objek di setiap *frame*. Metode pelacakan siluet menggunakan informasi yang ditandai dalam bentuk atau berupa wilayah objek. *State space models* dan *Variational methods* merupakan contoh *countour evalution*, sedangkan Hausdorff dan *Hough transform* merupakan contoh *matching shape*

2.2.7 Oklusi

Oklusi adalah keadaan dimana antara satu objek dengan objek lainnya berhimpitan sehingga *bounding box* mengecil atau objek menjadi *intermittent*. Oklusi terjadi ketika objek yang dilacak atau artribut penting yang digunakan untuk mengenali identitas dari objek tersebut tidak tersedia pada kamera untuk melacak ketika objek tersebut masih berada pada jangkauan kamera (Lee, dkk, 2014). Oklusi dianggap sebagai salah satu rintangan yang menantang dalam pelacakan objek karena mengurangi akurasi pelacakan.

Dalam citra video, beberapa bagian dari objek mungkin tidak terlihat karena oklusi (Rizky, 2017). Terdapat dua jenis oklusi pada video transportasi yaitu oklusi total, dimana objek tertutup secara keseluruhan dan oklusi sebagian (parsial) dimana objek tertutup hanya sebagian oleh benda yang menghalanginya. Adapun berdasarkan penyebabnya, oklusi dalam video transportasi dibagi dua. Pertama oklusi yang disebabkan oleh benda bergerak, seperti oklusi yang terjadi ketika antara satu kendaraan dengan kendaraan lainnya saling berhimpitan. Kedua oklusi yang disebabkan oleh benda diam, seperti oklusi yang terjadi ketika kendaraan terhalang oleh pohon, semak-semak, tiang lampu, lampu lalu lintas dan lain-lain. Berikut contoh dari oklusi pada video transportasi yang dapat dilihat pada Gambar 2.10.

Gambar 2.10: (a). Oklusi oleh Benda Diam, (b). Oklusi oleh Benda Bergerak

2.2.8 Filter Korelasi (FK)

Filter Korelasi telah digunakan dalam pelacakan visual selama beberapa tahun, karena kecepatan dan efisiensi komputasinya (Hamdi dan Ghanem, 2018). Filter Korelasi pada intinya menggunakan pembelajaran yang membedakan antara target dengan sekitarnya dengan tujuan dapat diterapkan pada *region* yang diinginkan dalam *frame* tertentu untuk menentukan lokasi target (Mueller, dkk, 2017). Filter Korelasi memungkinkan untuk pengambilan *dense sampling* (translasi) di sekitar target dengan biaya komputasi yang rendah (*low computational cost*). Hal ini dikarenakan memodelkan semua kemungkinan penerjemahan target dalam *search window* sebagai *circulant shifts* dan menggabungkannya untuk membentuk matriks X. Struktur matriks *circulant* memberikan solusi sangat efisien untuk *ridge regression* dalam masalah *domain fourier* (Mueller, dkk, 2017). Tujuan *tracking* menggunakan Filter Korelasi adalah menemukan fungsi yang meminimalisasi *squared eror* dari data matriks x_i dan target regresi y_i (Henriques, dkk, 2014),

$$\min_{w} \sum_{i} (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{2.15}$$

Meminimalisasi memiliki bentuk tertutup yaitu (Zhou dan Wang, 2019):

$$w = (X^T X + \lambda I)^{-1} X^T y \tag{2.16}$$

dengan w adalah vektor Filter Korelasi, X adalah matriks *circulant* yang terdiri dari semua *circulant shifts* berdasarkan *sample* x_i, y adalah vektor dari Gaussian 2D dan setiap elemen dari y adalah target regresi y_i, λ adalah parameter reguler untuk mengontrol *overvitting* dan I adalah matriks identitas. Dalam domain Fourier, kuantitas bernilai kompleks sehingga persamaan 2.16 dapat ditulis sebagai berikut (Henriques, dkk, 2014):

$$w = (X^H X + \lambda I)^{-1} X^H y \tag{2.17}$$

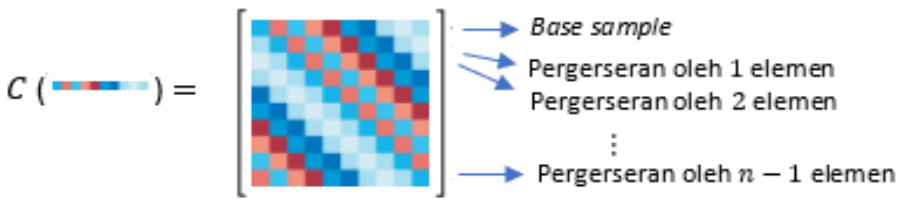
dengan X^H adalah *Hermitian transpose* atau *transpose* konjugat. Matriks *circulant* dapat didiagonalisasi menggunakan Transformasi Fourier Diskrit (TFD) seperti berikut:

$$X = F diag(\hat{x}) F^H \tag{2.18}$$

dengan F adalah matriks Transformasi Fourier Diskrit (TFD), F^H adalah *Hermitian transpose* atau *transpose* konjugat, $\hat{x}$ adalah notasi Transformasi Fourier Diskrit (TFD) dari vektor x dan $\hat{x} = F(x) = \sqrt{n}Fx$. Berikut bentuk dari matriks *circulant*, yaitu:

$$X = C(X) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \tag{2.19}$$

adapun ilustrasi dari matriks *circulant* dapat di lihat pada Gambar 2.11.



Gambar 2.11: Ilustrasi dari Matriks *Circulant*
(Yue dan Li, 2017).

Menerapkan persamaan 2.18 pada persamaan 2.17, menghasilkan *update* model yaitu:

$$\hat{w} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \tag{2.20}$$

dengan $\hat{x}^*$ adalah *complex-conjugate* dari $\hat{x}$. Untuk mendeteksi lokasi target baru di *frame* berikutnya (*target localization*) dapat menghitung respon $f(z)$. Untuk semua *candidate patches* z, kemudian mendiagonalisasi $f(z)$ agar diperoleh lokasi target tersebut (Zhou dan Wang, 2019):

$$\hat{f}(z) = \hat{r}_p = \hat{w} \odot \hat{z} \tag{2.21}$$

Persamaan 2.20 dan 2.21 merupakan solusi dalam *primal domain*, adapun solusi dalam *dual domain* yaitu diperoleh *update* model:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{x}^* \odot \hat{y} + \lambda} \tag{2.22}$$

Sedangkan untuk *target localization*-nya yaitu:

$$\hat{r}_d = \hat{z} \odot \hat{x}^* \odot \hat{\alpha} \tag{2.23}$$

Jika ditambakan kernel menggunakan *Kernel trick*, maka diperoleh (Hamdi dan Ghanem, 2018):

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{2.24}$$

dengan $\hat{k}^{xx}$ adalah baris pertama dari kernel matriks $K = C(K^{xx})$, adapun notasi $(\hat{k})$ adalah vektor dari TFD.

Berikut notasi yang relevan pada *Kernel trick*, dimana pemetaan *input* dari masalah linear ke non linear *feature-space* $\varphi(x)$ dengan *kernel trick* terdiri dari (Henriques, dkk, 2014):

(a). Menyatakan solusi w sebagai kombinasi linear dari sampel:

$$w = \sum_i \alpha_i \varphi(x_i) \tag{2.25}$$

dengan α adalah variabel optimasi dalam *dual domain*, sedangkan dalam *primal domain* adalah w, sehingga persamaan 2.15 menjadi

$$\min_{\alpha} \|K\alpha - y)\|^2 + \lambda \alpha^T K \alpha \tag{2.26}$$

(b). Penulisan algoritma dalam bentuk *dot-product*:

$$\varphi^T(x)\varphi(x') = \kappa(x, x')$$

yang dapat dihitung menggunakan fungsi kernel Gaussian atau Polinomial dan lain-lain.

Dot-produck antara semua pasangan sampel biasanya disimpan dalam matriks kernel K berdimensi $n \times n$, dengan elemen-elemen:

$$K_{i,j} = \kappa(x_i, x_j) \tag{2.27}$$

Adapun solusi untuk versi kernel dari *Ridge Regression*, yaitu:

$$\alpha = (K + \lambda I)^{-1} y \tag{2.28}$$

dengan K juga merupakan matriks *circulant*, karena K matriks *circulant* maka persamaan 2.24 dapat didiagonalisasi yang dapat dilihat pada persamaan 2.29, sedangkan untuk deteksi lokasi target yang mana dalam Kernel Filter Korelasi, kernel matriks didefinisikan sebagai $K^z = k(P_z^i, P^j x)$, sehingga untuk *target localization*-nya yaitu:

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{2.29}$$

2.2.9 Filter Kalman (FK)

Metode Filter Kalman diperkenalkan pertama kali oleh R.E Kalman pada tahun 1960 (Kleeman, 2007). Filter Kalman merupakan metode statistikal yang berbasis pada *point tracking* yang memiliki kelebihan selalu memberikan solusi optimal berupa hasil prediksi yang presisi. Filter Kalman digunakan untuk memperkirakan suatu keadaan dari suatu sistem yang linear dan berdistribusi Gaussian (Patel dan Thakore, 2013). Selain itu, waktu komputasi metode Filter Kalman relatif cepat karena proses penghitungan berupa titik dan *output* berupa titik serta metode Filter Kalman dapat melakukan pelacakan yang bersifat *single object* maupun *multiple objects*.

Pada kasus pelacakan *single object*, Filter Kalman tidak dapat melakukan pelacakan apabila objek yang bergerak di dalam suatu *frame* lebih dari satu objek. Filter Kalman menentukan letak objek berdasarkan adanya pergerakan objek dari *frame* $i+1$ dan *frame* i. Akibatnya jika terdapat dua objek yang bergerak pada saat yang sama, maka dua objek tersebut akan terlacak menjadi satu objek.

Metode Filter Kalman memperkirakan lokasi objek yang paling mungkin pada *frame* saat ini berdasarkan hasil dari pelacakan target pada *frame* sebelumnya yang dilanjutkan dengan mencari lokasi target di sekitaran lokasi sebelumnya (Huang dan Hong, 2011). Berikut dua langkah utama metode Filter Kalman yaitu prediksi dan koreksi (Huang dan Hong, 2011). Adapun siklus diskrit Filter Kalman pada citra digital ditunjukkan oleh Gambar 2.12.

Gambar 2.12: Siklus Diskrit Filter Kalman pada *Image Processing*

dari Gambar 2.12 menunjukkan bahwa tahap prediksi (*time update*) merupakan tahap estimasi dari sistem model dinamik, sedangkan tahap koreksi *(measurement update)* merupakan tahap estimasi dari model pengukuran.

Didefinisikan vektor $X_k = [x, y, v_x, v_y]$, vektor *measurement* $Z_k = [x, y]^T$, dimana x dan v_x adalah gambar target pada arah posisi horizontal dan perpindahannya, sedangkan y dan v_y adalah gambar target pada posisi vertikal dan perpindahannya. Berikut adalah persamaan prediksi dan koreksi dari metode Filter Kalman.

a. Tahap Prediksi
   Persamaan Prediksi 1 :

$$X_{pred_k} = AX_{k-1} \tag{2.30}$$

Persamaan Prediksi 2 :

$$P_{pred_k} = AP_{k-1}A^T + Q \tag{2.31}$$

b. Tahap Koreksi
Persamaan Kalman-Gain:

$$K_k = P_{pred_k}H^T(HP_{pred_k}H^T + R)^{-1} \tag{2.32}$$

Persamaan *Update* 1:

$$X_k = X_{pred_k} + K_k(Z_k - HX_{pred_k}) \tag{2.33}$$

Persamaan *Update* 2:

$$P_k = P_{pred_k} - K_kHP_{pred_k} \tag{2.34}$$

dengan:

A = Matriks transisi

H = Matriks pengukuran (*measurement*)

Q = Matriks kovarian *noise* proses

R = Matriks kovarian *noise* pengukuran (*measurement*)

Nilai dari matriks transisi (A), matriks pengukuran (H), matriks kovarian *noise* proses (Q), dan matriks kovarian *noise* pengukuran (R) adalah sebagai berikut:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

BAB 3
METODE PENELITIAN

Pada bab ini dijelaskan mengenai beberapa metode penelitian yang digunakan untuk mencapai tujuan penelitian.

3.1 Objek Penelitian

Pada penelitian ini, objek penelitian bersifat objek tunggal dan multiobjek dan untuk kebutuhan simulasi digunakan video rekaman kendaraan bergerak di jalan raya.

3.2 Peralatan

Peralatan yang digunakan adalah kamera digital dan *tripod* untuk proses merekam video lalu lintas dan *software* dengan bahasa pemrograman JAVA sebagai alat bantu untuk melakukan simulasi. Selain itu diperlukan juga OpenCV sebagai *library* pada JAVA untuk membantu proses pengolahan citra. *Software Tool* yang digunakan untuk implementasi bahasa JAVA dan OpenCV adalah Netbeans 8.2.

3.3 Tahap Penelitian

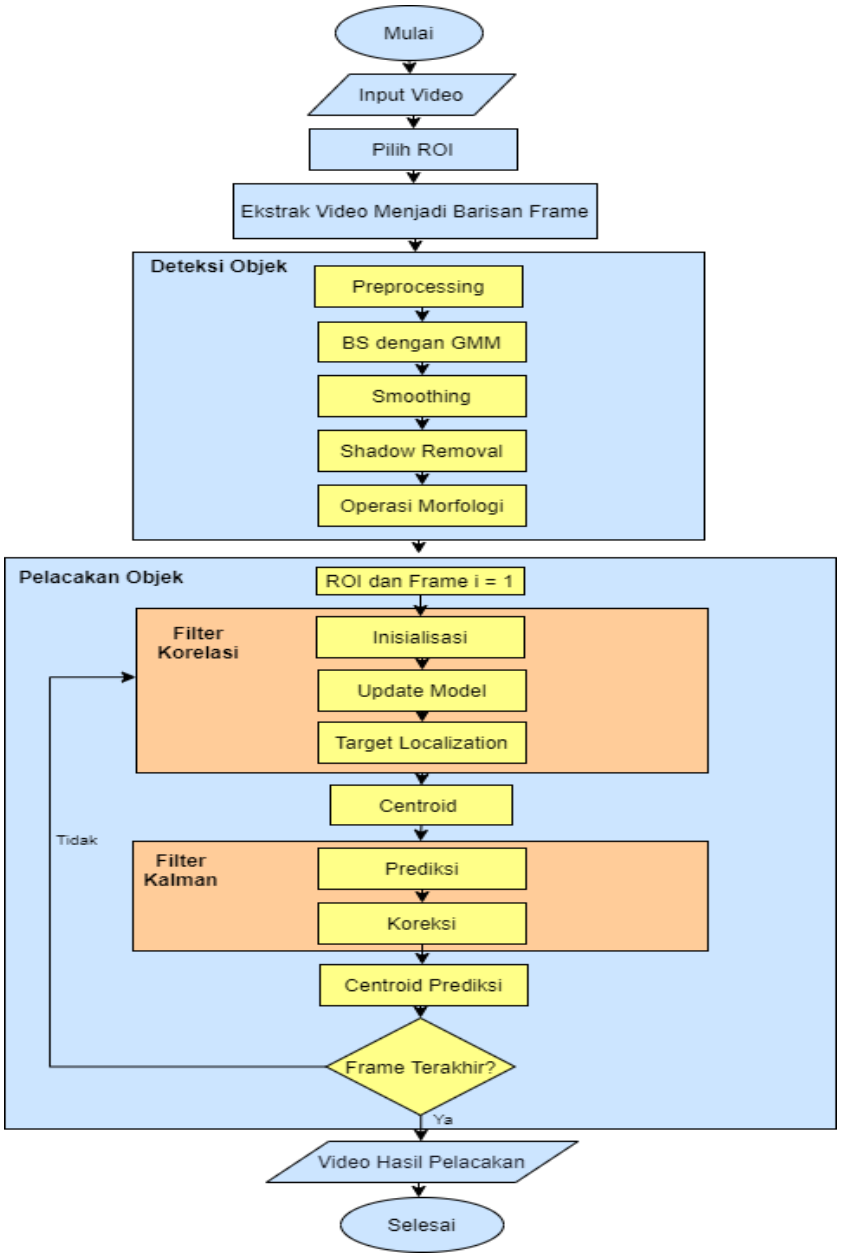
Penelitian ini meliputi beberapa tahapan-tahapan proses. Setiap proses dari tahapan-tahapan tersebut mempengaruhi dalam pengerjaan Tesis ini. Adapun langkah-langkahnya sebagai berikut:

1. Studi Literatur
   Pada tahap ini, dilakukan pendalaman kajian, analisis pada penelitian-penelitian sebelumnya dan pengumpulan berbagai informasi tentang pengolahan citra digital, video digital, deteksi objek, pelacakan objek dan hal lain yang berkaitan dengan penelitian ini. Berbagai informasi tersebut didapatkan dari berbagai sumber pustaka yaitu buku, jurnal, dan internet.

2. Akuisisi Data
   Pada tahap ini, dilakukan pengumpulan data video. Pengumpulan data video ada dua cara. Pertama, video diperoleh dengan pengambilan data mandiri, berupa rekaman video kendaraan yang melintas di jalan raya baik pada saat pagi atau siang hari dengan cuaca cerah serta ada satu video yang diambil saat sore hari. Kedua, video diperoleh dari dataset video transportasi beberapa penelitian sebelumnya dan dari Youtube.

3. Analisis dan Perancangan Proses
   Pada tahap ini, diawali dengan analisis data berupa rekaman video, deteksi objek, kemudian dilakukan perancangan proses seperti kombinasi

metode Filter Korelasi dan Filter Kalman. Filter Korelasi digunakan untuk mendeteksi gerak atau perpindahan objek sedangkan metode Filter Kalman digunakan untuk antisipasi atau prediksi gerak dari objek. Berikut blok diagram dari pelacakan objek kendaraan bergerak dapat dilihat pada Gambar 3.1.



Gambar 3.1: Diagram Tahapan Pelacakan Kendaraan Bergerak Pada Video Transportasi.

Adapun tahapan dari Gambar 3.1 adalah sebagai berikut:

a. *Input* video yang telah tersimpan pada direktori. Video diperoleh

dari akuisisi di jalan raya, penelitian-penelitian sebelumnya dan Youtube.

b. Menentukan ROI (*region of interest*), dimana ROI digunakan sebagai area pemilihan objek yang akan tersimpan sebagai *template* acuan untuk pelacakan objek. Berdasarkan ROI pengguna dapat memilih objek sesuai kebutuhan, sehingga piksel lainnya selain ROI tidak akan terpilih.

c. Ekstrak video menjadi barisan *frame* yaitu video yang digunakan dilakukan proses pemecahan menjadi beberapa rangkaian citra.

d. Pada tahap selanjutnya dilakukan deteksi objek dengan cara: *preprocessing* dengan perbaikan kontras dan kecerahan untuk meminimalisir kemunculan bayangan, segmentasi dengan metode *Background Subtraction* dimana dalam menginisialisasi model *background*-nya dengan menggunakan metode GMM, *smoothing* dengan Median Filter, lalu setelah *image* menjadi lebih halus karena *noise* berkurang, diperlukan proses *shadow removal* untuk menghilangkan bayangan dan *filling holes* dengan operasi morfologi.

e. Pelacakan objek dengan metode Filter Korelasi untuk mendeteksi pergerakan atau perpindahan objek kendaraan pada video transportasi. Target yang teridentifikasi ditandai dengan *bounding box* dan dihitung *centroid*-nya. *Centroid* yang diperoleh akan dijadikan *input* pada Filter Kalman untuk dilakukan antisipasi atau memprediksi gerak di *frame* berikutnya.

f. Pelacakan objek dengan metode Filter Kalman untuk antisipasi atau memprediksi pergerakan objek kendaraan. Filter Kalman menghasilkan prediksi letak objek di *frame* selanjutnya, kemudian *output* metode Filter Kalman tersebut digunakan sebagai *input* untuk perhitungan pada metode Filter Korelasi di *frame* selanjutnya.

g. Hasil pelacakan berupa video yang merupakan rangkaian dari *frame-frame* yang terdapat objek bergerak yang dibatasi *bounding box* yang berada dalam ROI.

4. Implementasi Program
   Pada tahap ini dilanjutkan dengan implementasi program dalam bentuk perangkat lunak untuk mempermudah dalam melakukan simulasi dan analisis hasil, sehingga proses pengerjaan dapat dilakukan lebih cepat.

5. Simulasi dan Analisis Hasil
   Pada tahap ini dilakukan simulasi dan analisis kinerja terhadap perangkat lunak yang telah selesai dibuat menggunakan *input* video rekaman kendaraan yang berlalu-lalang di jalan raya, yang mana akan membandingkan persentase akurasi dan tingkat ketepatan pelacakan objek kendaraan dari *input* video yang berbeda baik tanpa oklusi

maupun dengan oklusi dengan metode Filter Korelasi dan metode kombinasi Filter Korelasi dan Filter Kalman. Kemudian dilakukan analisis pembahasan sehingga dapat dicatat beberapa hal yang dijadikan pertimbangan dalam menarik kesimpulan. Pada tahap ini juga dilakukan evaluasi terhadap program yang telah dibangun. Hasil evaluasi dicatat untuk membenahi hal-hal yang masih kurang.

6. Penarikan Kesimpulan dan Saran
   Setelah mendapatkan hasil pembahasan, kemudian dilakukan penarikan kesimpulan dari hasil pengerjaan dan memberikan saran untuk pengembangan berikutnya.

7. Diseminasi
   Pada tahap ini, penelitian yang telah dilakukan selanjutnya diseminasikan pada suatu seminar internasioal, yaitu *The 4th International Conference On Science* (ICOS) di Universitas Hasanuddin, Makassar pada tanggal 22-23 Agustus 2020.

8. Penyusunan Tesis
   Setelah semua tahap terlewati, kemudian dilakukan penyusunan Tesis dan kelengkapan dalam penelitian.

3.4 Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Komputasi, Departemen Matematika, Fakultas Sains dan Analitika Data. Institut Teknologi Sepuluh Nopember Surabaya. Adapun pengambilan video dilakukan di jalan raya di Surabaya dan diambil dari atas jembatan penyeberangan orang (JPO), serta adapula yang diambil di jalan raya di Pamekasan.

BAB 4
PERANCANGAN DAN IMPLEMENTASI

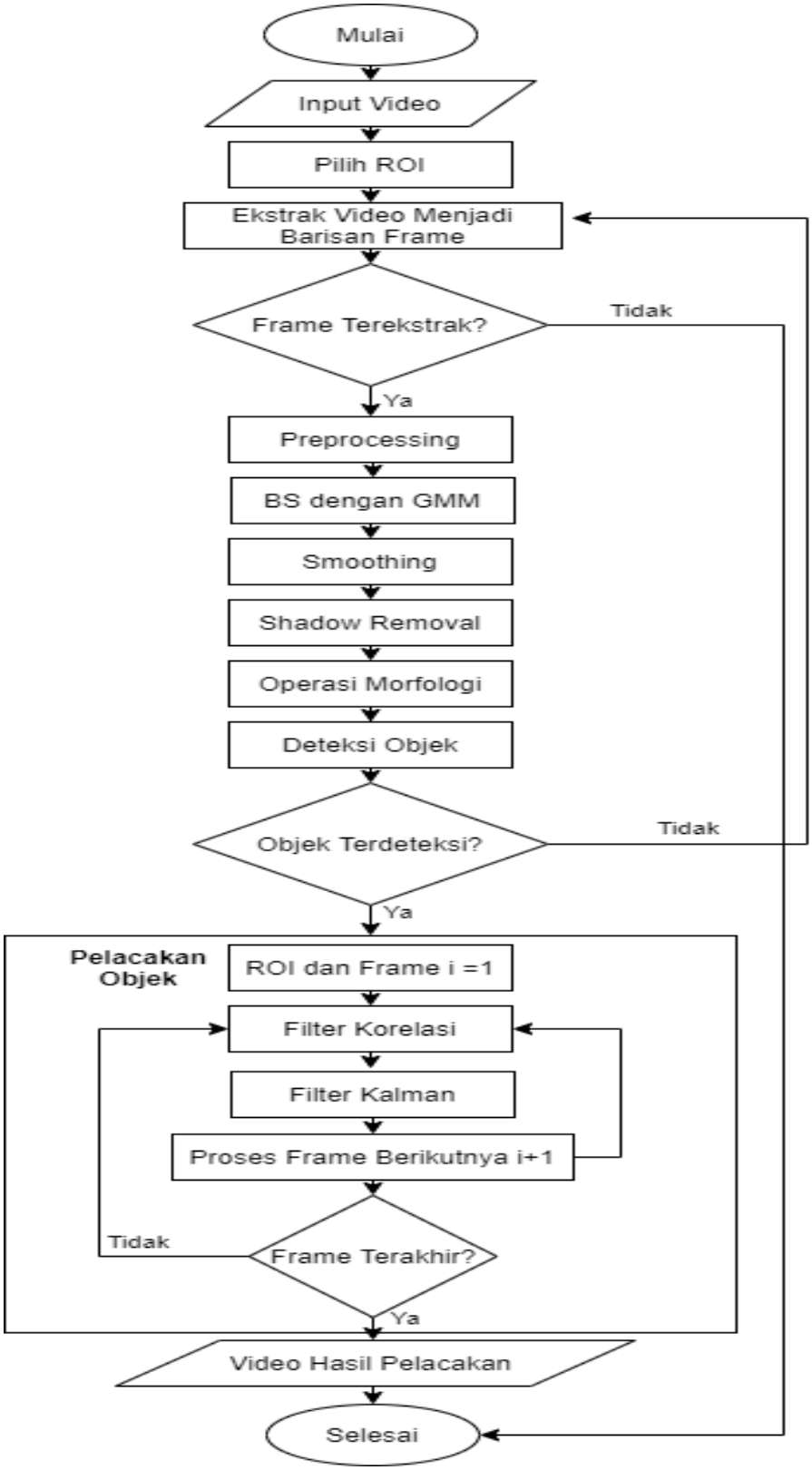
Pada bab ini dijelaskan mengenai analisis dan perancangan proses penelitian yang meliputi perancangan proses deteksi objek, dan perancangan proses pelacakan objek dengan kombinasi metode Filter Korelasi (FK) dan Filter Kalman (FK). Selain proses-proses tersebut, pada bab ini juga dibahas mengenai perancangan perangkat lunak, implementasi perangkat lunak pada Java dan implementasi algoritma dengan tujuan untuk mempermudah simulasi dan analisis hasil dari perancangan proses yang telah dibuat.

4.1 Perancangan Proses

Sesuai dengan tujuan dari penelitian ini, penelitian yang dirancang harus mampu menangani oklusi kendaraan yang bergerak secara objek tunggal dan multi objek dengan data masukan berupa video digital kondisi lalu lintas, dikarenakan sistem untuk pelacakan objek akan sulit mendeteksi kendaraan akibat adanya oklusi pada video, yang dapat dilihat pada Gambar 4.1.

Gambar 4.1: Contoh Oklusi pada Video Transportasi

Perancangan proses bertujuan untuk menjaga agar proses pelacakan objek kendaraan bergerak dengan kombinasi metode Filter Korelasi dan Filter Kalman berjalan dengan teratur. Gambar 4.2 menjelaskan diagram alir perancangan proses penelitian.

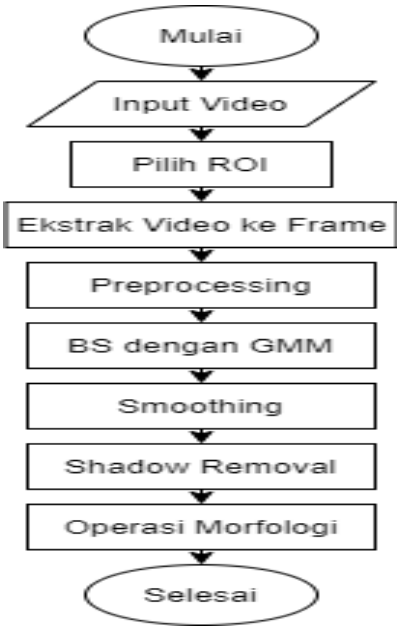


Gambar 4.2: Diagram Alir Perancangan Proses

dari Gambar 4.2, proses diawali dengan *input* video yang diambil dari penyimpanan komputer, pilih ROI (*region of intereset*), video dikonversi menjadi barisan *frame* (*sequence of frames*). Selanjutnya setiap *frame* yang berhasil diekstrak akan dilakukan deteksi objek dengan proses yaitu: *preprocessing*, *Background Substraction* menggunakan metode *Gaussian Mixture Model* (GMM), proses *smoothing* menggunakan metode Median Filter, *shadow removal*, *object detection enhancement* menggunakan operasi morfologi. Pada tahap berikutnya dilakukan proses pelacakan objek menggunakan metode Filter Korelasi kemudian diperoleh *centroid* dari masing-masing objek dan untuk memprediksi pergerakan objek di *frame* selajutnya menggunakan metode Filter Kalman. Proses pelacakan objek dilakukan dari *frame* pertama sampai *frame* terakhir dari video dan bisa mengubah ROI objek, tidak harus sama pada saat deteksi objek. Berikut perancangan proses dari deteksi objek dan pelacakan objek.

4.1.1 Perancangan Deteksi Objek

Proses deteksi objek dianggap juga sebagai inisialisasi sebelum melakukan pelacakan objek untuk membedakan *background* dan *foreground* dari setiap *frame* atau menentukan objek yang akan dilacak, proses deteksi objek pada penelitian ini diawali dengan *preprocessing*, segmentasi menggunakan metode *Background Subtraction*, dimana untuk menginisialisasi *background* model dalam memperoleh *foreground image* menggunakan metode GMM (*Gaussian Mixture Model*). Proses selanjutnya adalah *smoothing* dengan Median Filter, untuk menghilangkan bayangan dengan *shadow removal*, dan *filling holes* dengan operasi morfologi. Berikut Diagram alir dari proses deteksi objek yang dapat dilihat pada Gambar 4.3.



Gambar 4.3: Diagram Alir Proses Deteksi Objek

Adapun langkah-langkah dalam proses deteksi objek, adalah sebagai berikut:

1. *Preprocessing.*
   Setelah *input* video, pilih ROI (*region of interest*), kemudian mengekstrak video menjadi barisan *frame*, maka pada setiap *frame* akan dilakukan *preprocessing* terlebih dahulu agar diperoleh citra atau *frame* dengan kualitas yang baik dan layak untuk tahap berikutnya, dimana akan memperbaiki ketajaman dan kecerahan pada citra yang dapat berpengaruh pada proses deteksi objek dan pelacakan objek. Proses *preprocessing* pada penelitian ini menggunakan *contrast and brightness enhancement.* Misalkan piksel $f(i, j)$ adalah intensitas piksel asal pada koordinat (i, j) dan $g(i, j)$ adalah intensitas piksel hasil, dengan $\alpha > 0$ adalah parameter gain (*contrast*) dan β adalah parameter bias (*brightness*), maka proses *contrast and brightness enhancement* didefinisikan oleh persamaan 4.1.

$$g(i, f) = \alpha f(i, j) + \beta \tag{4.1}$$

2. *Background Subtraction.*
   Pada proses *Background Subtraction* akan menggunakan metode GMM (*Gaussian Mixture Model*) untuk menginisialisasi *background* model dalam memperoleh *foreground image.* GMM memiliki kelebihan yaitu tahan terhadap perubahan atau mampu beradaptasi karena setiap piksel pada setiap perubahan *frame* akan dievaluasi melalui proses *update* parameter bobot, standard deviasi, dan rata-rata (Wicaksono, 2017). Proses *Background Subtraction* dengan GMM menggunakan parameter α (*learning rate*) dengan nilai 0.01, jumlah komponen Gaussian yaitu 3, T (*threshold*) dengan nilai 0.4. Inisialisasi awal dari beberapa parameter GMM antara lain: ω_k yaitu bobot dari setiap piksel pada Gaussian ke-k dengan nilai $\frac{1}{3}$, yang mana nilai 3 merupakan jumlah dari distribusi Gaussian, μ_k adalah *mean* dari setiap piksel pada Gaussian ke-k, dimana masing-masing piksel dari setiap Gaussian mempunyai nilai random antara 0 sampai 255. σ_k yaitu standar deviasi dari setiap piksel pada Gaussian ke-k dengan nilai 6. Berikut diberikan contoh cara kerja metode GMM. Misalkan *frame* yang terekstrak seperti pada Gambar 4.4.

Frame Input

149	20	23
192	187	52
201	76	241

Gambar 4.4: Contoh Matriks Intensitas *Frame* yang Terekstrak

Kemudian berdasarkan pertidaksamaan 2.4, diperoleh pertidaksamaan 4.2.

$$|X_{i,j,t} - \mu_{i,j,k}| < 2.5 * \sigma_{i,j,k} \tag{4.2}$$

Adapun contoh inisialisasi awal parameter GMM diberikan pada Gambar 4.5.

Inisialisasi Awal Bobot (ω)

Gaussian 1

1/3	1/3	1/3
1/3	1/3	1/3
1/3	1/3	1/3

Gaussian 2

1/3	1/3	1/3
1/3	1/3	1/3
1/3	1/3	1/3

Gaussian 3

1/3	1/3	1/3
1/3	1/3	1/3
1/3	1/3	1/3

Inisialisasi Awal *Mean* (μ)

Gaussian 1

156	2	23
198	15	41
202	211	112

Gaussian 2

99	12	12
32	200	44
142	42	222

Gaussian 3

146	23	33
212	152	142
51	75	163

Inisialisasi Awal Standar Deviasi (σ)

Gaussian 1

6	6	6
6	6	6
6	6	6

Gaussian 2

6	6	6
6	6	6
6	6	6

Gaussian 3

6	6	6
6	6	6
6	6	6

Gambar 4.5: Contoh Inisialisasi Parameter GMM

dimana nilai intensitas X pada piksel (i, j) dari *frame input* yang memenuhi pertidaksamaan 4.2 masuk dalam distribusi dan berdasarkan inisialisasi awal σ sebesar 6, maka diperoleh nilai dari ruas kanan pada pertidaksamaan 4.1 sebesar 15, sedangkan ruas kiri merupakan selisih antara intensitas *frame input* dan intensitas *mean* pada *frame* sebelumnya yang masing-masing diberikan pada Gambar 4.4 dan Gambar 4.5, dimana hasil dari ruas kiri dapat dilihat pada Gambar 4.6.

Gaussian 1

7	18	5
6	172	11
1	135	129

Gaussian 2

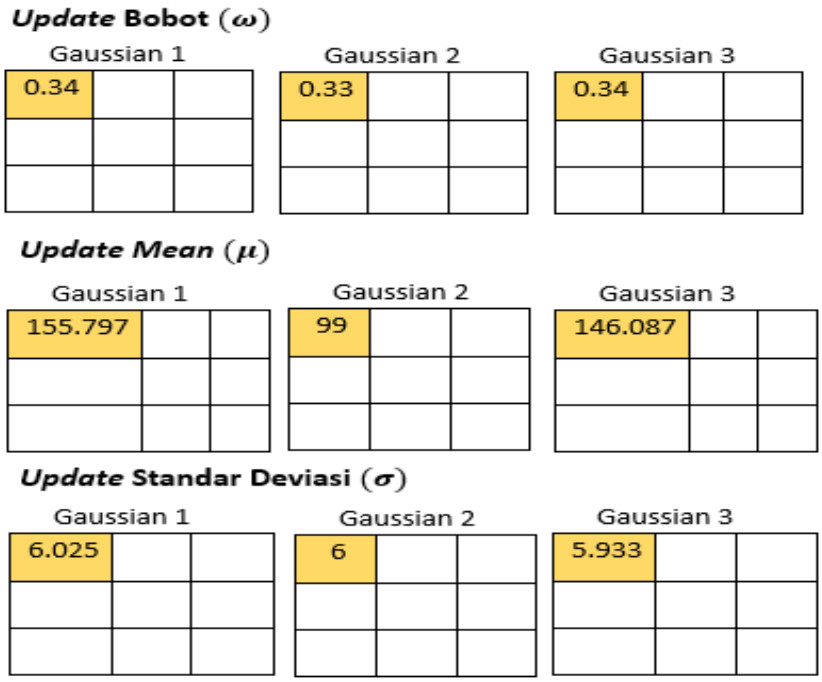
50	8	16
160	13	8
59	34	19

Gaussian 3

3	3	5
20	35	90
150	1	78

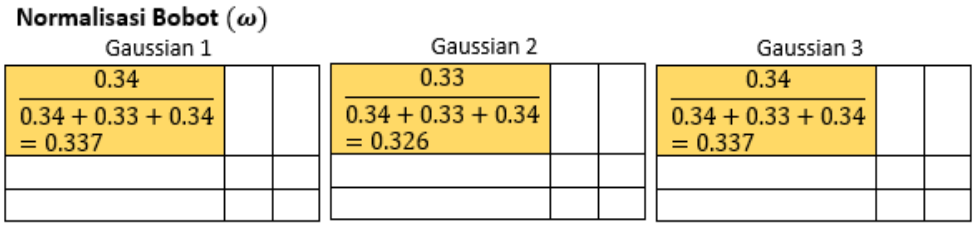
Gambar 4.6: Matriks Selisih Intensitas *Frame Input* dan *Mean*

Untuk proses selanjutnya akan diberikan contoh hanya pada piksel $(1, 1)$. Kemudian *update* bobot dari masing-masing Gaussian dengan menggunakan persamaan 2.6. Dari Gambar 4.6 dapat dilihat nilai intensitas $X_{1,1}$ yang cocok terhadap distribusinya adalah yang kurang dari 15 yaitu Gaussian 1 dan Gaussian 3, sehingga $\mu_{1,1}$ dan $\sigma_{1,1}$ dengan $k = 1$ dan $k = 3$ akan di-*update* dengan menggunakan persamaan 2.8 dan persamaan 2.9. Hasil *update* parameter GMM tersebut dapat dilihat pada Gambar 4.7



Gambar 4.7: Hasil *Update* Parameter GMM

Bobot yang di-*update* kemudian akan dinormalisasi agar jumlah dari bobot tiap komponen Gaussian tepat nilainya sama dengan 1. Hasil normalisasi bobot dapat dilihat pada Gambar 4.8



Gambar 4.8: Hasil Normalisasi Bobot

Untuk memilih model yang mencerminkan *background*, terlebih dahulu bobot akan diurutkan berdasarkan $\frac{\omega}{\sigma^2}$ yang mana hasil perhitungannya dapat dilihat pada Gambar 4.9

Perhitungan $\frac{\omega}{\sigma^2}$

Gaussian 1			Gaussian 2			Gaussian 3		
$\frac{0.337}{6.025^2} = 0.00928$			$\frac{0.326}{6^2} = 0.00906$			$\frac{0.337}{5.933^2} = 0.00957$		

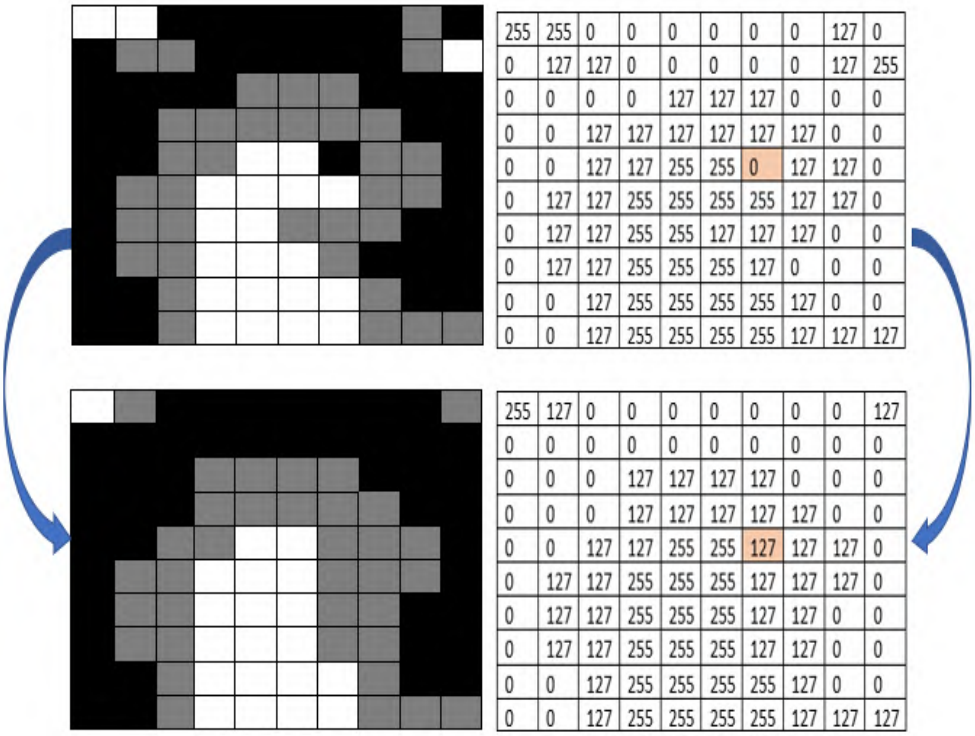
Gambar 4.9: Hasil Perhitungan $\frac{\omega}{\sigma^2}$

dari Gambar 4.9 urutan bobot dari besar ke kecil secara berturut-turut adalah Gaussian 3, Gaussian 1, Gaussian 2. Selanjutnya akan dicari beberapa distribusi pertama yang sudah terurut menggunakan persamaan 2.10 dengan T (*threshold)* adalah 0.4, sehingga pada contoh distribusi yang dipilih adalah Gaussian 3 dan Gaussian 1, kemudian semua Gaussian yang terpilih dicocokkan dengan intensitas dari *frame input* dengan menggunakan pertidaksamaan 4.2, jika ada distribusi Gaussian yang cocok maka intensitas merupakan *background*, dan jika tidak maka intensitas merupakan *foreground*.

2. *Smoothing*
   Proses *smoothing* pada penelitian menggunakan metode Median Filter. Median Filter mengurutkan nilai piksel dan ketetanggaannya dari yang terkecil, kemudian diperoleh nilai tengahnya. Proses Median Filter didenisikan pada persamaan 4.3.

$$f(x, y) = median(g(s, t)) \tag{4.3}$$

dengan $f(x, y)$ adalah nilai piksel dari hasil Median Filter, $g(s, t)$ adalah nilai piksel awal yang dicari nilai mediannya. Pada penelitian ini Median Filter membutuhkan parameter masukan berupa *size*, yaitu parameter untuk menentukan seberapa jauh jangkauan tetangga yang akan diproses untuk menentukan nilai mediannya. *Size* yang digunakan adalah 5 dan 3, dimana *size* ini selalu berukuran ganjil. Semua piksel pada *image* akan dicari nilai median dari piksel tersebut dan semua piksel tetanggannya, dimana piksel tetangga diperoleh berdasarkan parameter *scale*. Agar tidak mempengaruhi *smoothing* pada piksel berikutnya, maka nilai median dari hasil *smoothing* tidak menggantikan nilai lama, melainkan akan diletakkan pada *image* yang baru. *Output* dari proses ini berupa *image* atau *frame grayscale* dengan 3 tingkat keabuan yang berisi nilai median dari setiap piksel. Hitam bernilai 0, abu-abu bernilai 127, dan putih bernilai 255. Masing-masing warna atau nilai tersebut secara berturut-turut merepresentasikan *background*, bayangan dari *foreground*, dan *foreground*. Berikut contoh dari proses *smoothing* dengan Median Filter dapat dilihat pada Gambar 4.10.



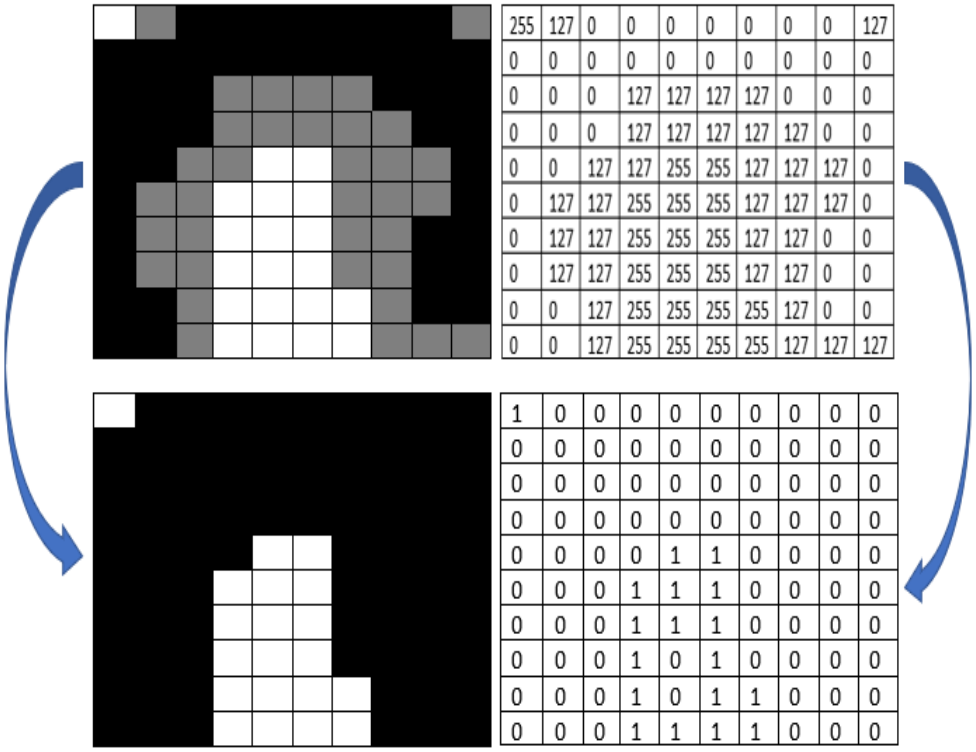
Gambar 4.10: *Smoothing* dengan Median Filter

dari Gambar 4.10, proses *smoothing* dengan Median Filter menggunakan *size* 3, nilai masing-masing piksel yang bertetanggaan setelah diurutkan adalah sebagai berikut: $0, 127, 127, 127, 127, 127, 255, 255, 255$. Hasil pengurutan tersebut mendapatkan nilai median 127, nilai median ini akan digunakan untuk menggatikan nilai pusat *mask*, sehingga nilai 0 diganti dengan 127.

3. *Shadow Removal*
   Shadow removal dilakukan untuk menghapus bayangan setelah *smoothing* dengan Median Filter, karena hasil dari *Background Subtraction* dengan metode GMM mendeteksi bayangan yang dianggap sebagai *noise*. *Image/frame* hasil dari proses *smoothing* mempunyai tiga tingkat keabuan yaitu $0, 127$, dan 255, sehingga data masukan pada proses *shadow removal* berupa *image grayscale* hasil dari proses *smoothing*. *Shadow removal* adalah proses dimana nilai-nilai tersebut dipetakan ke angka biner yaitu 0 dan 1 seperti yang dijelaskan pada persamaan 4.4. (Wicaksono, 2017).

$$I_{biner}(x,y) = \begin{cases} 0, & I_{gray}(x,y) = 0 \quad atau \quad I_{gray}(x,y) = 127 \\ 1, & I_{gray}(x,y) = 255 \end{cases} \tag{4.4}$$

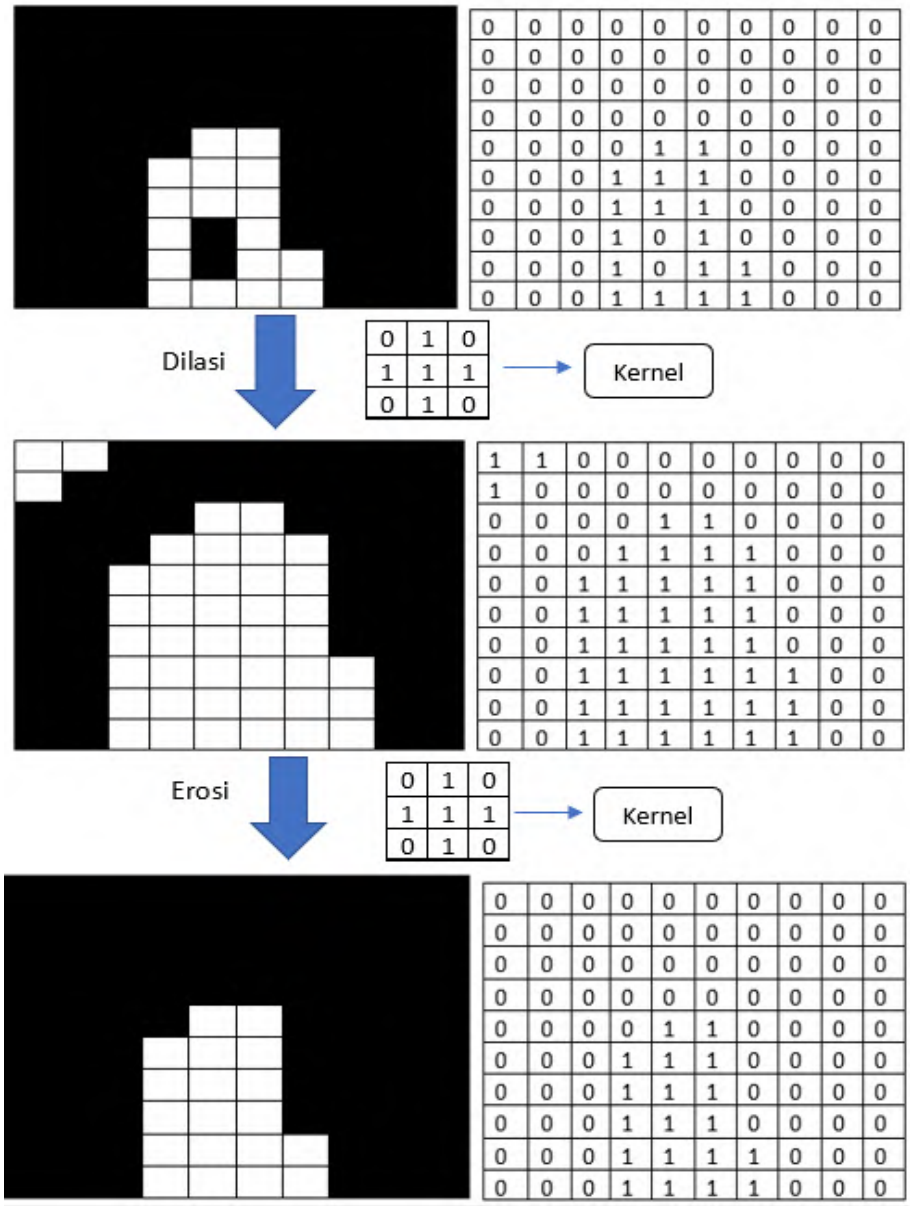
   Berikut Contoh dari proses *Shadow removal* yang dapat di lihat pada Gambar 4.11.



Gambar 4.11: Proses *Shadow Removal*

dari Gambar 4.11, nilai piksel yag awalnya berupa *image grayscale* akan diubah ke *image* biner, sehingga *image* dengan nilai piksel 0 dan 127 akan diubah menjadi 0 dan menghasilkan warna hitam, sedangkan *image* dengan nilai piksel 255 akan diubah ke nilai 1 dengan warna putih. Proses *shadow removal* berlaku pada seluruh piksel dari citra, sehingga akan mempermudah deteksi objek dikarenakan menghasilkan *image* biner yang hanya mempunyai dua nilai, yaitu 0 atau hitam yang merepresentasikan *background* dan 1 atau putih yang merepresentasikan *foreground* atau objek bergerak

4. Operasi Morfologi
   Operasi morfologi antara lain berguna untuk menutup lubang-lubang pada hasil *shadow removal* yang terkadang muncul di tengah-tengah objek, pada penelitian ini operasi morfologi menggunakan operasi *Closing*. Berikut adalah contoh proses *Closing* pada piksel yang dapat dilihat pada Gambar 4.12.

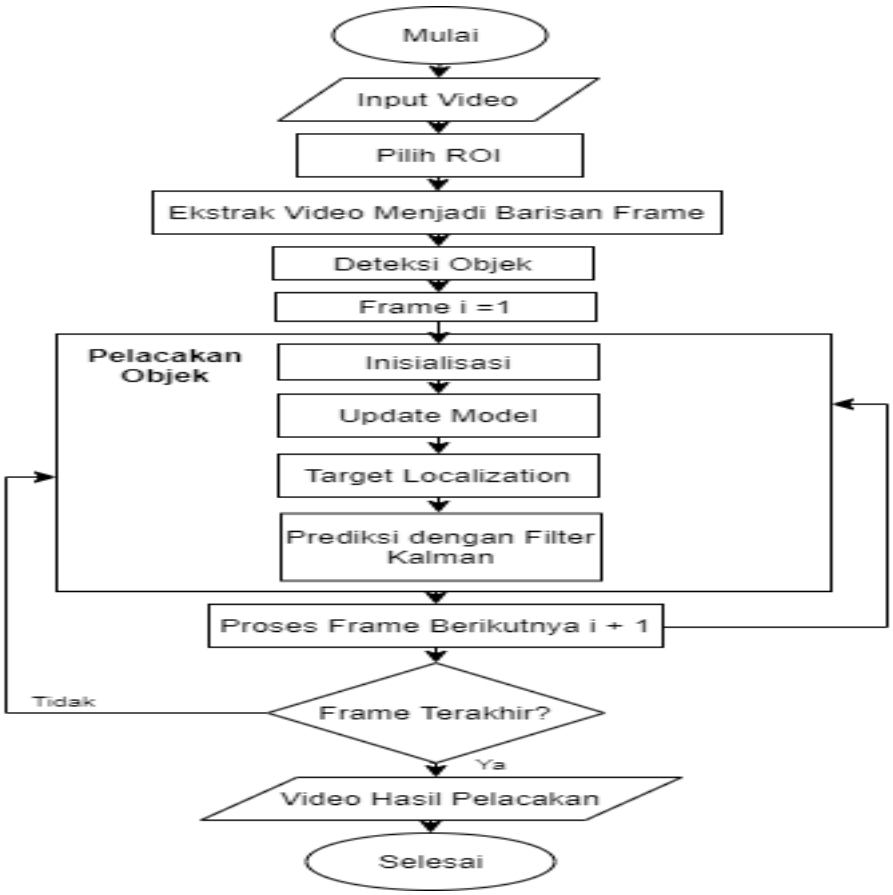


Gambar 4.12: Proses Morfologi dengan Operasi *Closing*

dari Gambar 4.12, operasi morfologi menggunakan operasi *Closing*, dengan ukuran kernel 3×3. Operasi *Closing* diawali dengan Dilasi untuk menebalkan objek kemudian dilakukan proses Erosi untuk menipiskan objek, dimana proses Erosi mengembalikan kurang lebih hampir sama dengan ukuran objek sebelum dilakukan proses Dilasi dan penutupan lubang objek, sehingga diperoleh objek yang lebih baik yang dapat megurangi lubang pada objek.

4.1.2 Perancangan Pelacakan Objek

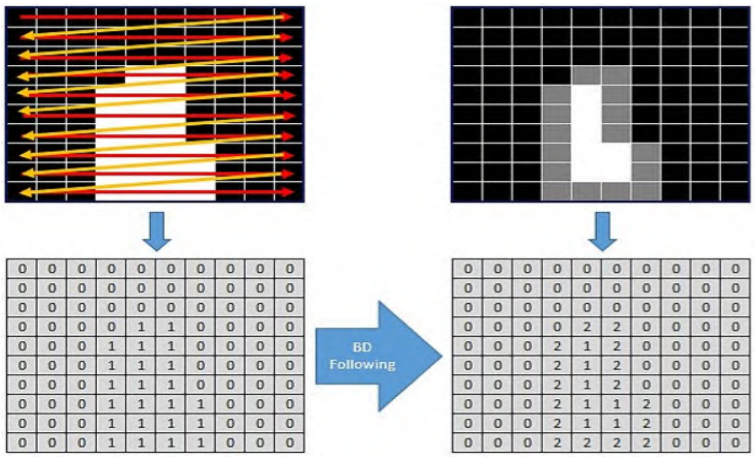
Proses pelacakan objek dilakukan dengan menggunakan metode Filter Korelasi, sedangkan prediksi pergerakan objek menggunakan metode Filter Kalman. Kedua metode tersebut saling melengkapi dalam melakukan pelacakan objek, sehingga *input* metode Filter Kalman bergantung pada *output* metode Filter Korelasi dan begitu juga sebaliknya. Objek yang akan dilacak telah didefinisikan melalui pemilihan ROI objek. Meskipun letak objek pada *frame* pertama telah diketahui, proses pelacakan objek akan tetap dilakukan dari *frame* pertama, karena informasi yang didapatkan untuk pelacakan pada *frame* selanjutnya tidak hanya dari pemilihan ROI. Diagram alir pelacakan objek dengan metode kombinasi Filter Korelasi dan Filter Kalman dapat dilihat pada Gambar 4.13.



Gambar 4.13: Diagram Alir Perancangan Pelacakan Objek

Pelacakan objek diawali dengan *input* video, pemilihan ROI, ekstrak video menjadi barisan *frame*, deteksi objek dengan membuat *bounding box* dan menentukan area minimum dari objek. Hasil dari deteksi objek tersebut dijadikan *input* pada proses pelacakan objek dengan metode Filter Korelasi sebagai inisialisasi, dilanjutkan dengan *update* model dan deteksi objek di *frame* selanjutnya (*target localization*), untuk membuat hasil (*centroid*) dari *target localization* lebih presisi, maka diprediksi dengan Filter Kalman. *Output* metode Filter Kalman juga berupa *centroid* dan koordinat *bounding box*. Kedua *output* tersebut menjadi *input* Filter Korelasi untuk *frame* selanjutnya. Proses ini dilakukan pada semua *frame* dari video. Untuk Filter Korelasi bekerja dalam *primal domain*, dimana Filter Korelasi digunakan untuk mendeteksi gerak atau perpindahan objek sedangkan metode Filter Kalman digunakan untuk antisipasi atau prediksi gerak dari objek. Berikut prancangan pelacakan objek dengan kombinasi metode Filter Korelasi dan Filter Kalman.

1. Inisialisasi
   Setelah proses *input* video, pemilihan ROI, ekstrak video menjadi barisan *frame*, maka masuk pada proses Filter Korelasi yang diawali dengan inisialisasi dari objek, karena pelacakan objek bersifat *multiple objects*, maka perlu dilakukan deteksi objek terlebih dahulu, dimana pada penelitian ini menggunakan *Background Subtraction* dengan metode GMM untuk menginisialisasi *background* model dalam memperoleh *foreground image*. Proses inisialisasi dalam Filter Korelasi dilakukan untuk menentukan objek yang akan dilacak (daerah kalkulasi) dengan menentukan nilai x, y, v_x, v_y atau istilah lain dari v_x adalah *width* dan v_y adalah *heigth* dari objek yang terdeteksi. Karena menggunakan GMM, maka perlu menggunakan konsep *connected component* untuk mendeteksi kontur pada *image biner* berdasarkan pada *outer border* atau batas paling luar dengan metode *border following* (Suzuki, dkk, 2019). Berikut ilustrasi *border following* pada *image* biner dapat dilihat pada Gambar 4.14, dengan border diberikan Label 2.



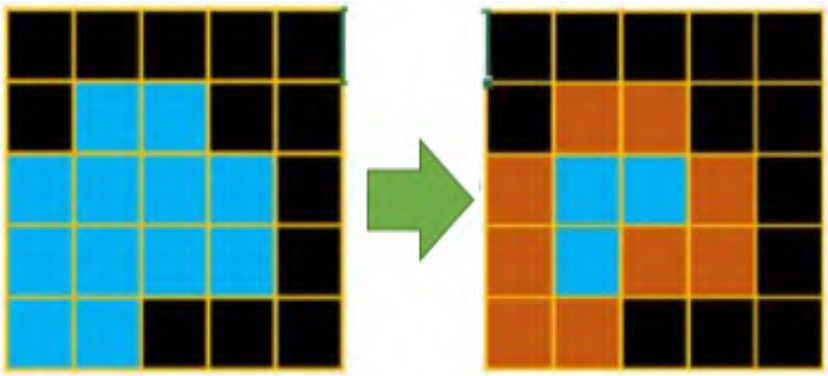
Gambar 4.14: Ilustrasi *Border Following*

Berikut algoritma *border following*:

a. Proses pencarian objek dimulai dari atas ke bawah (*row*), kiri ke kanan (*column*). Cari piksel $f(i, j)$ sebagai kandidat piksel *outer border* yang memenuhi kondisi $f(i, j-1) = 0, f(i, j) = 1$ dan jika piksel merupakan *outer most border*, yaitu memenuhi kondisi berikut:
   - Piksel $f(i, 1), f(i, 2), ..., f(i, j-1)$ adalah piksel 0, atau
   - Piksel $f(i, h)$ sejajar dengan titik border, dan piksel $f(i, h+1)$ adalah *background*.

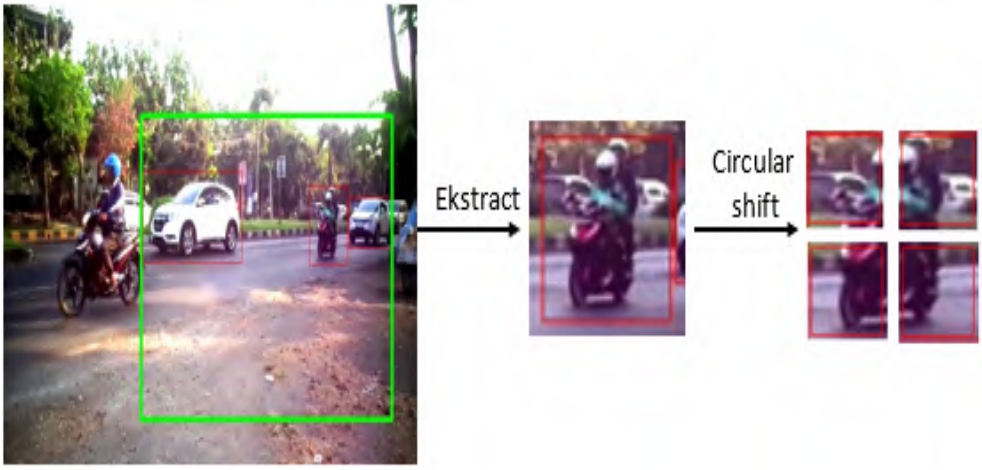
b. Jika batas akhir border telah ditemukan yaitu $f(i, j-1) = 1$ dan $f(i, j) = 0$ maka lakukan *labeling* sebagai tanda border.

c. Jalankan *raster scanning* hingga seluruh piksel terlewati. Contoh *border following* dapat dilihat pada Gambar 4.15.



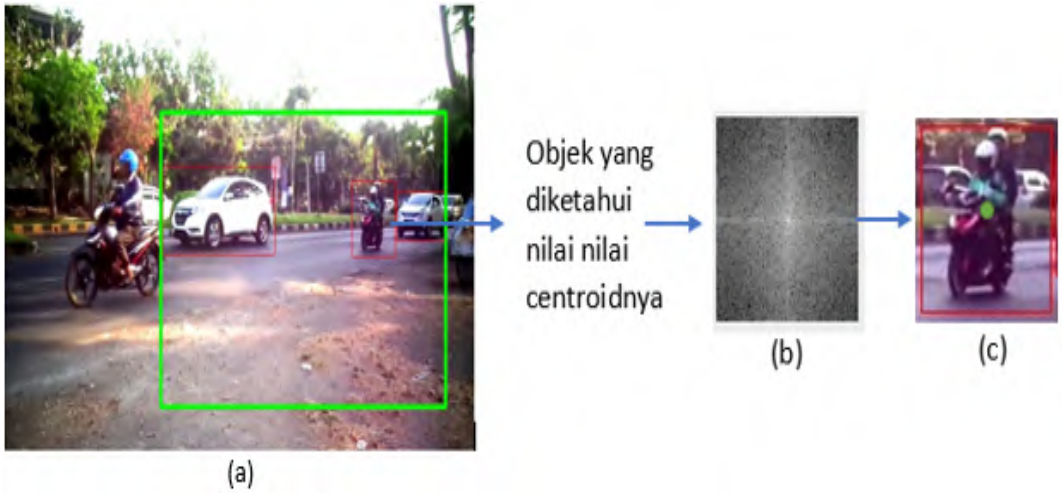
Gambar 4.15: Contoh *Border Following*

2. *Update* Model
   Update model dilakukan dengan menerapkan Filter Korelasi, proses ini juga disebut fase pelatihan (*training*) dimana akan membedakan *sample* positif yang merupakan area target objek dan *sample* negatif yang merupakan sekitaran objek (*background*). adapun untuk *training* dalam mencari *sample* menggunakan persamaan 2.20. Berikut ilustrasi dari training FK dengan *negative samples*.



Gambar 4.16: Ilustrasi Pelatihan Filter Korelasi

3. *Target Localization*
   Target localization disebut juga sebagai cara mendeteksi objek di *frame* selanjutnya dengan cara menghitung respons korelasi mirip dengan auto korelasi dan dapat ditingkatkan dengan menggunakan 2D-FFT dan 2D-IFFT, kemudian melabeli objek berupa *bounding box* berdasarkan persamaan 2.21. Berikut ilustrasi dari lokasi target yang dapat dilihat pada Gambar 4.17.



Gambar 4.17: Ilustrasi Lokasi Target dari Filter Korelasi. (a). *Model Update*, (b). Spektrum Model (Filter), (c) Objek dengan Nilai *Centroid*

dari Gambar 4.17 dapat dilihat label dari objek yang terdeteksi sehingga nantinya diperoleh *centroid* dari objek dengan koordinat (x, y) sebagai $B(x, y)$ dengan lebar (*width*) dan tinggi (*height*), kemudian dari *bounding box* akan ditentukan koordinat titik *centroid* sebagai $C(x, y)$ dengan

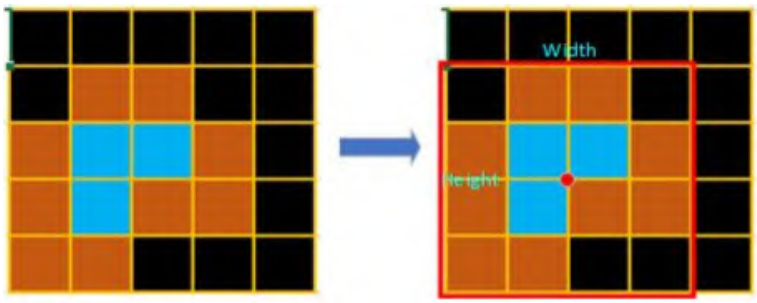
persamaan 4.5.

$$C(x, y) = (C_x, C_y) \tag{4.5}$$

dengan:
$C_x = B_x + \frac{B_{width}}{2}$
$C_y = B_y + \frac{B_{heigth}}{2}$
didenisikan,
C_x = koordinat x pada *centroid*
C_y = koordinat y pada *centroid*
B_x = koordinat x pada *bounding box*
B_y = koordinat y pada *bounding box*
B_{width} = lebar *bounding box*
B_{heigth} = tinggi *bounding box*

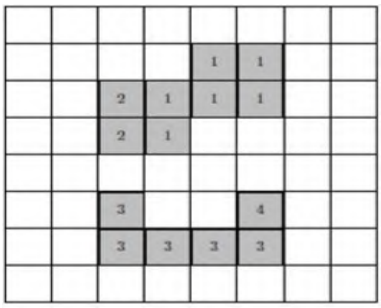
Tampilan *bounding box* dapat dilihat pada Gambar 4.18. Adapun untuk menentukan ukuran objek atau identifikasi objek, yang mana dapat mengidentifikasi objek kendaraan sepeda motor, mobil, dan kendaraan lainnya berdasarkan persamaan 2.21 dapat dihitung respon tertinggi dari objek, model ini menggunakan rumus:

$$W = B_{width} * B_{height} \tag{4.6}$$



Gambar 4.18: Ilustrasi *Bounding Box* dan *Centroid*

Untuk pelacakan objek yang bersifat *multiple objects* maka untuk menghubungkan antara satu objek dengan objek lainnya yang berada dalam ROI, objek atau kendaraan akan diberi label berupa ID berdasarkan urutan kemunculan kendaraan tersebut pada ROI. Berikut contoh dari *labeling multiple objects* yang dapat dilihat pada Gambar 4.19.



Gambar 4.19: Contoh *Labeling* Multi Objek

Ketika objek berada dalam ROI, posisi kendaraan antar *frame* akan selalu dicatat dan ketika objek meninggalkan ROI akan dihitung. Data yang tersimpan tersebut ditampilkan dalam data keluaran berupa banyaknya jumlah kendaraan yang ada di ROI. Untuk memastikan objek tidak terhitung dan tersimpan lebih dari satu kali, maka proses pelacakan objek antar *frame* perlu dilakukan. Dua objek dikatakan sama jika memenuhi pertidaksamaan 4.7 dan pertidaksamaan 4.8.

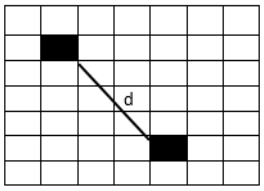
$$L_{objek1} \leq \frac{L_{objek2} + R_{objek2}}{2} \leq R_{objek1} \tag{4.7}$$

$$T_{objek2} \leq T_{objek1} \leq B_{objek2} \tag{4.8}$$

dimana objek1 adalah objek yang sedang diproses objek2 adalah sebuah objek pada *frame* sebelumnya, L adalah batas kiri objek, R adalah batas kanan objek, T adalah batas atas objek, dan B adalah batas bawah objek. Jika kedua objek teridentikasi sama maka ID pada objek sebelumnya akan berpindah ke objek yang sedang diproses dan melakukan identifikasi kendaraan. Untuk mengukur jarak *centroid* objek pada seiap *frame* digunakan pengukuran jarak *Euclidean*, yaitu:

$$d = \sqrt{(x_k - x_0)^2 + (y_k - y_0)^2} \tag{4.9}$$

dimana (x_0, x_k) dan (y_0, y_k) adalah *center* dari masing-masing objek. Berikut contoh dari ilustrasi jarak *Euclidean* pada pelacakan objek yang dapat dilihat pada Gambar 4.20.

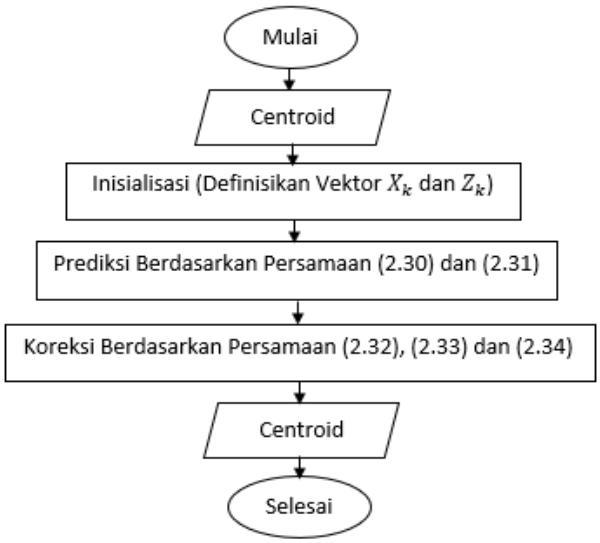


Gambar 4.20: Contoh Jarak Centroid Objek Antar *Frame*

Adapun dalam menyesuaikan *centroid* objek, sebagai ilustrasi misalkan

objek yang sudah ditandai dengan *bounding box* pada Gambar 4.16 memiliki *centroid* dengan: $x = 422.9878$, $y = 401$, perpanjangan $x = 16$, dan perpanjangan $y = 28$ diperoleh *centroid* dengan koordinat $x = 430.9878, y = 415$. Kordinat tersebut akan digunakan untuk memperkirakan posisi dan skala baru dari objek berdasarkan hasil respon korelasi. Pada penelitian ini akan menggunakan metode prediksi dengan Filter Kalman.

3. Filter Kalman
   Algoritma Filter Kalman pada dasarnya menghitung besarnya gain Kalman, mengestimasi variabel *state* dan mengestimasi *error* kovarians setiap indeks waktu. Filter Kalman pada hakikatnya digunakan untuk melacak objek tunggal, akan tetapi jika diaplikasikan untuk melacak objek banyak perlu adanya algoritma lain untuk mendeteksi objek sebelum memprediksi dengan Filter Kalman. Metode Filter Kalman digunakan untuk memprediksi pergerakan objek saat *frame* $i + 1$ berdasarkan *frame* ke-i. Berikut diagram alir proses Filter Kalman yang dapat dilihat pada Gambar 4.21.



Gambar 4.21: Diagram Alir Perancangan Pelacakan dengan Metode Filter Kalman

Merujuk pada Bab 2, langkah pertama adalah tahapan prediksi dan tahap kedua adalah koreksi. Berikut adalah contoh penyelesaiannya:

a. Tahap prediksi
   Persamaan prediksi 1: $X_{pred_k} = AX_{k-1}$

$$X_{pred_k} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 430.9878 \\ 415 \\ 0.9371 \\ 0 \end{bmatrix} = \begin{bmatrix} 431.9249 \\ 415 \\ 0.9371 \\ 0 \end{bmatrix}$$

Persamaan prediksi 2 : $P_{pred_k} = AP_{k-1}A^T + Q$

$$
\begin{aligned}
P_{pred_k} &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.982 & 0 & 0.9371 & 0 \\ 0 & 0.982 & 0 & 0.9371 \\ 0.9371 & 0 & 3.7755 & 0 \\ 0 & 0.9371 & 0 & 3.7755 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \\
&+ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 7.6317 & 0 & 4.7126 & 0 \\ 0 & 7.6317 & 0 & 4.7126 \\ 4.7126 & 0 & 4.7755 & 0 \\ 0 & 4.7126 & 0 & 4.7755 \end{bmatrix}
\end{aligned}
$$

b. Tahap koreksi
Persamaan Kalman-gain : $K_k = P_{pred_k} H^T (HP_{pred_k} H^T + R)^{-1}$

$$
\begin{aligned}
K_k &= \frac{\left(\begin{bmatrix} 7.6317 & 0 & 4.7126 & 0 \\ 0 & 7.6317 & 0 & 4.7126 \\ 4.7126 & 0 & 4.7755 & 0 \\ 0 & 4.7126 & 0 & 4.7755 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right)}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 7.6317 & 0 & 4.7126 & 0 \\ 0 & 7.6317 & 0 & 4.7126 \\ 4.7126 & 0 & 4.7755 & 0 \\ 0 & 4.7126 & 0 & 4.7755 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} \\
&= \begin{bmatrix} 0.8841 & 0 \\ 0 & 0.8841 \\ 0.546 & 0 \\ 0 & 0.546 \end{bmatrix}
\end{aligned}
$$

Persamaan *update* 1 : $X_k = X_{pred_k} + K_k(Z_k - HX_{pred_k})$

$$
\begin{aligned}
X_k &= \begin{bmatrix} 431.9249 \\ 415 \\ 0.9371 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.8841 & 0 \\ 0 & 0.8841 \\ 0.546 & 0 \\ 0 & 0.546 \end{bmatrix} \cdot \left(\begin{bmatrix} 431.9249 \\ 415 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 431.9249 \\ 415 \\ 0.9371 \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} 431.9249 \\ 415 \\ 0.9371 \\ 0 \end{bmatrix}
\end{aligned}
$$

Persamaan *update* 2 : $P_k = P_{pred_k} - K_k H P_{pred_k}$

$$
\begin{aligned}
P_k &= \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.8841 & 0 \\ 0 & 0.8841 \\ 0.546 & 0 \\ 0 & 0.546 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right) \cdot \\
&\begin{bmatrix} 7.6317 & 0 & 4.7126 & 0 \\ 0 & 7.6317 & 0 & 4.7126 \\ 4.7126 & 0 & 4.7755 & 0 \\ 0 & 4.7126 & 0 & 4.7755 \end{bmatrix} \\
&= \begin{bmatrix} 0.8841 & 0 & 0.546 & 0 \\ 0 & 0.8841 & 0 & 0.546 \\ 0.546 & 0 & 2.2026 & 0 \\ 0 & 0.546 & 0 & 2.2026 \end{bmatrix}
\end{aligned}
$$

Hasil penghitungan diperoleh nilai *centroid* dengan koordinat $x = 431.9249$, $y = 415$ dari nilai X_k baris pertama dan kedua. *Input frame* selanjutnya adalah koordinat *rectangle* (Pt) dari penghitungan *centroid* berdasarkan persamaan 4.5 dengan nilai perpanjangan koordinat x dan y yang sama seperti *input* awal yaitu, perpanjangan $x = 16$, dan perpanjangan $y = 28$. Langkah-langkah pelacakan objek dengan metode Filter Korelasi dan Filter Kalman seperti diatas dilakukan secara berurutan dan dikenakan pada semua video. Hasil akhir pelacakan berupa urutan *frame* dan video pelacakan yang telah ditandai dengan *rectangle* dan titik *centroid* objek.

4.2 Perancangan Perangkat Lunak

Tahap selanjutnya adalah perancangan perangkat lunak yang meliputi lingkungan perancangan dan perancangan data.

4.2.1 Lingkungan Perancangan

Lingkungan perancangan dalam Tesis ini meliputi perangkat keras dan perangkat lunak yang digunakan, dengan spesifikasi yang dapat dilihat pada Tabel 4.1.

Tabel 4.1: Perangkat yang Digunakan

No.	Perangkat Keras	Perangkat Lunak
1	Intel Core i5 processor 2400S	Windows 10 Pro with Media Center 64 bit
2	Memory 4096 MB RAM	Tools menggunakan Java, Netbeans IDE 8.2, OpenCV-4.1.0

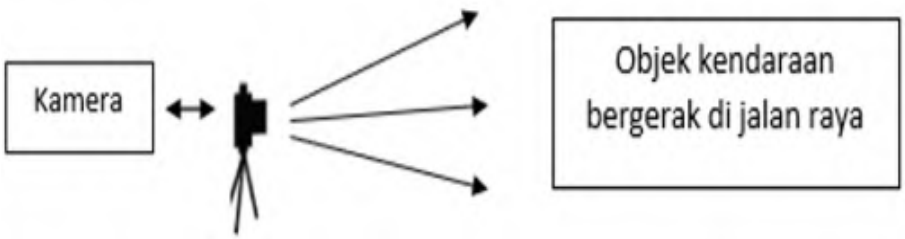
dengan perangkat keras dan perangkat lunak seperti dijelaskan pada Tabel 4.1 diharapkan memperoleh hasil yang efisien dari segi efisiensi waktu komputasi dan tentunya ketepatan pelacakan objek.

4.2.2 Perancangan Data

Data yang digunakan untuk pelacakan kendaraan bergerak terdiri dari tiga macam yaitu, data masukan, data proses, dan data keluaran. Data masukan berupa video rekaman *offline* yang dilakukan secara mandiri oleh peneliti maupun dari penelitian-penelitian sebelumnya atau dari Youtube. Data proses adalah data masukan yang dibutuhkan oleh metode-metode dari serangkaian proses yang telah disebutkan sebelumnya, sedangkan data keluaran adalah informasi mengenai hasil pelacakan objek yang telah dilakukan oleh perangkat lunak.

Adapun penjelasan dari data yang digunakan untuk pelacakan kendaraan bergerak yaitu sebagai berikut:

1. Data Masukan
   Akuisisi data video dilakukan di jalan raya pada saat siang dan sore hari. Data lainnya didapatkan dari dataset yang telah disediakan pada beberapa penelitian sebelumnya dan situs Youtube. Untuk pengambilan video secara mandiri sudut pandang kamera saat pengambilan video tampak dari atas jalan dan dari samping bawah. Video harus mencakup semua bagian ruas jalan, sehingga sistem mampu memproses seluruh objek yang terdeteksi pada ruas jalan tersebut. Adapun jarak kamera dengan jalan disesuaikan dengan tempat pengambilan video. Proses akuisisi data video yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 4.22.



Gambar 4.22: Proses Akuisisi Data Video

   Hasil akuisisi data video tersebut akan dijadikan data masukan pada perangkat lunak. Akan tetapi perlu dipertimbangkan resolusi piksel dari rekaman tersebut, semakin kecil resolusi piksel semakin menghemat waktu komputasi namun objek akan berukuran semakin kecil dan semakin besar resolusi piksel semakin jelas objek dapat terlihat. FPS (*frame per second*) dari kamera juga perlu dipertimbangkan dikarenakan akan mempengaruhi banyaknya *frame* saat konversi dari video menjadi barisan *frame*. Kamera digital (web camera) yang digunakan untuk proses akuisisi data memiliki $\geq$ 25 FPS.

2. Data Proses
   Data proses adalah data yang digunakan di dalam proses pengolahan

data masukan. Data proses diperoleh dari hasil pengolahan data masukan sesuai dengan tahapan algoritma dan metode yang telah disusun. Tabel 4.2 menjelaskan tahapan dari data proses.

Tabel 4.2: Data Proses

No.	**Tahapan**	***Input***	***Output***
1	*Input* data	Video	*Frame* citra
2	Menentukan ROI	Video	Citra ROI
3	Konversi video menjadi barisan *frame*	Video	*Frame* citra
4	*Preprocessing*	*Frame* citra	*Preprocessed frame* citra
5	*Background Subtraction*	*Preprocessed frame* citra	*Foreground image*
6	*Smoothing*	*Foreground image*	*Image grayscale smoothing*
7	*Shadow Removal*	*Image grayscale smoothing*	*Image* biner
8	Operasi morfologi	*Image* biner	*Image* biner morfologi
9	Deteksi Objek	*Image* biner morfologi	Kumpulan objek dengan *labeling*
10	Pelacakan objek dengan metode Filter Korelasi	*Frame i* dan kumpulan objek	*Centroid* dan kumpulan objek
11	Prediksi dengan metode Filter Kalman	*Centroid* dan kumpulan objek	*Centroid* dan video hasil pelacakan

3. Data Keluaran
   Data keluaran yang dihasilkan dari rancangan perangkat lunak berupa semua video hasil pelacakan objek dengan penandaan posisi target (objek) berupa *bounding box* di setiap *frame*, serta menghasilkan keluaran berupa *image* dari masing-masing proses, yaitu proses *preprocessing*, GMM, *smoothing*, *shadow removal*, operasi morfologi, dan deteksi objek.

4.3 Implementasi Perangkat Lunak pada Java

User interface atau antar muka sistem dibuat semenarik dan semudah mungkin, dengan tujuan mempermudah dalam mengoperasikan perangkat lunak. Berikut ditampilan desain antar muka dan implementasinya yang telah dibuat.

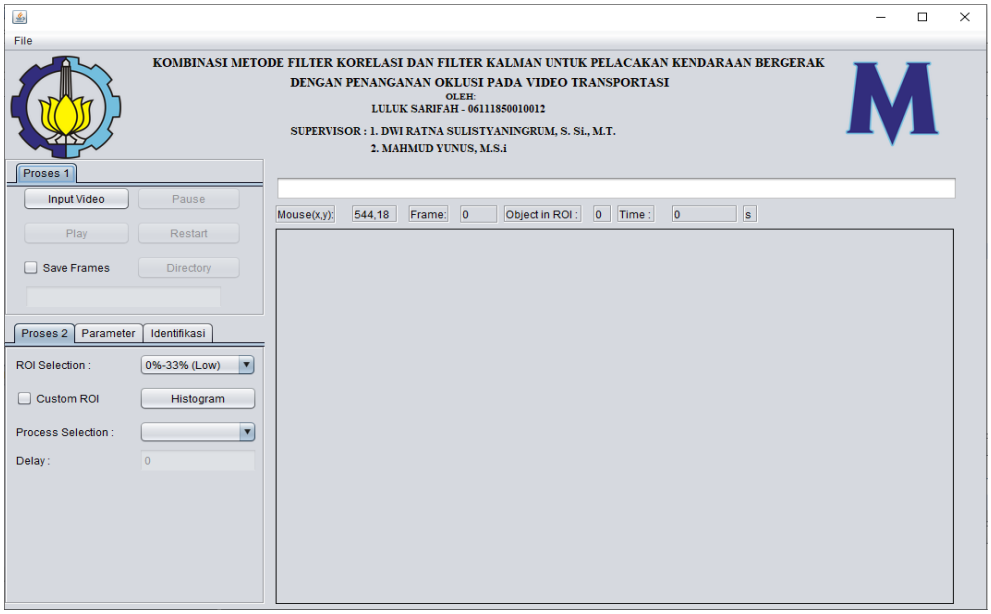
4.3.1 Halaman Awal *User Interface*

Sebelum melihat halaman utama *user interface*. Berikut ditampilkan halaman awal dari *user interface* yang dapat dilihat pada Gambar 4.23.

Gambar 4.23: Halaman Utama *User Interface*

4.3.2 Halaman Utama *User Interface*

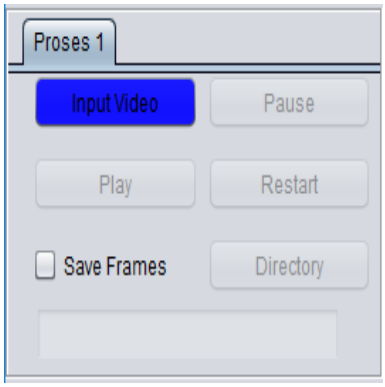
Berikut tampilan halaman utama dari *user interface* yang dapat dilihat pada Gambar 4.24.



Gambar 4.24: Halaman Utama *User Interface*

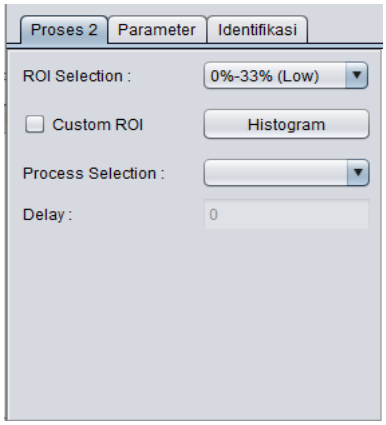
dari Gambar 4.24 dapat dilihat bahwa tampilan utama memuat 3 panel yaitu panel *input* dan proses 1, panel proses 2, parameter, dan Identifikasi serta panel video, yang ditampilkan dalam satu tampilan sehingga memudahkan *user* dalam navigasi menu-menu pada perangkat lunak. Berikut penjelasan dari masing-masing panel pada perangkat lunak.

1. Implementasi Panel *Input* dan Proses 1
   Panel *input* dan proses 1 adalah bagian halaman yang digunakan untuk memilih dan meng*input*kan video, memutar video, dan menghentikan pemutaran video, serta melakukan *restart* pada video tersebut. Kemudian juga berisi tempat (direktori) untuk menyimpan hasil video yang diolah. Panel *input* dan proses 1 dapat dilihat pada Gambar 4.25.



Gambar 4.25: Panel *Input* dan Proses 1

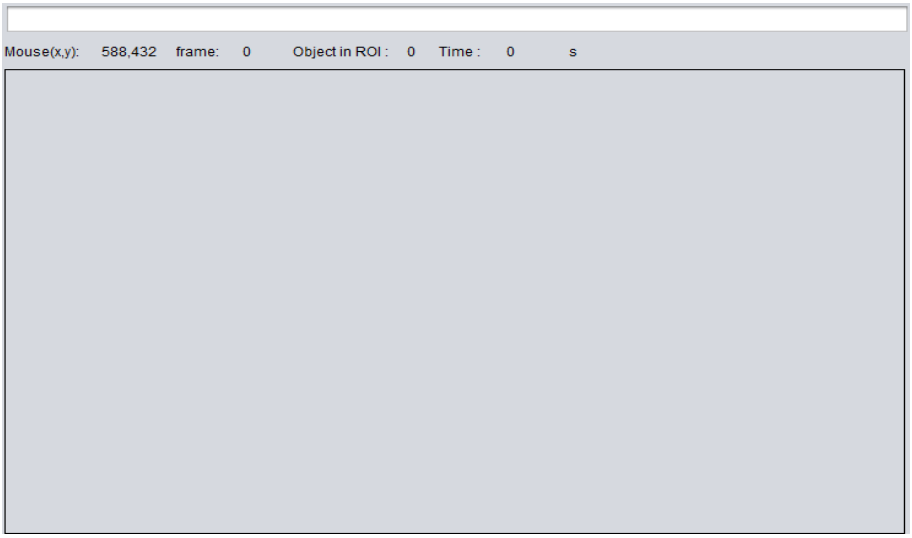
2. Implementasi Panel Proses 2, Parameter, dan Identifikasi
   Panel proses 2, parameter, dan identifikasi adalah bagian halaman yang digunakan untuk mengontrol parameter-parameter dan memproses video setelah video di*input*kan. Panel proses 2, parameter, dan identifikasi dapat dilihat pada Gambar 4.26.



Gambar 4.26: Panel Proses 2, Parameter, dan Identifikasi

berdasarkan Gambar 4.26, pada panel proses 2 berisi halaman yang digunakan untuk menentukan ROI, melihat histogram dari video, memilih proses, dan mengatur *delay* dari suatu video. Panel parameter digunakan untuk mengontrol parameter yang dibutuhkan dalam proses deteksi objek seperti parameter alpha, beta, *size* dari Median Filter, *size* dari morfologi, area minimum, learning rate, dan SE (*structure element*) sedangkan panel identifikasi digunakan untuk melihat area objek, dan respon korelasi piksel dari objek serta berisi minimum dan maksimum dari area objek

3. Implementasi Panel Video
   Panel video adalah bagian halaman yang berguna untuk menampilkan hasil dari proses-proses yang berjalan dalam perangkat lunak. Panel video ini juga sebagai tempat *user* memasukkan daerah ROI dengan melakukan *drag and drop* pada lokasi yang diinginkan. Pada panel ini terdapat informasi mengenai urutan *frame* yang sedang berjalan dan jumlah objek yang terdeteksi di dalam ROI dan waktu komputasi dari proses pelacakan objek, selain itu pada panel ini, juga terdapat direktori dari video. Panel video dapat dilihat pada Gambar 4.27.



Gambar 4.27: Panel Video

4.4 Implementasi Algoritma

Implementasi pelacakan objek terdiri dari beberapa tahapan meliputi *input* video, pemilihan ROI, konversi video menjadi barisan *frame*, deteksi objek, deteksi pergerakan objek (pelacakan objek) dengan metode Filter Korelasi, dan prediksi atau antisipasi pergerakan objek menggunakan metode Filter Kalman. Berikut penjelasan dari setiap proses yang dilewati.

4.4.1 Implementasi *Input* Video

Pada proses *input* video, *user* diharuskan memasukkan atau memilih berkas video yang telah tersimpan pada komputer. Setelah *user* memasukkan

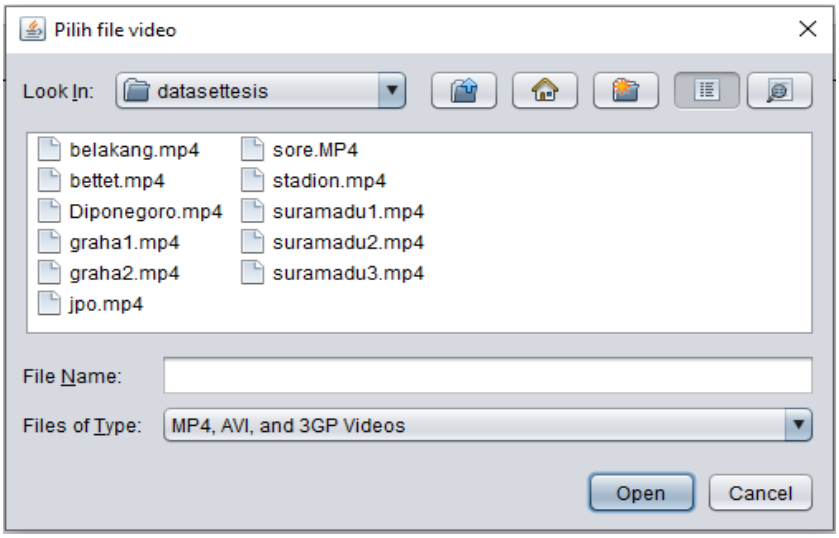
video, panel video akan menampilkan *image* awal dari video tersebut yang nantinya akan digunakan untuk kebutuhan pemilihan daerah ROI. Berikut implementasi dari *input* video pada Java.

Fungsi : *Input* video bertipe *.mp4, *.3gp, *.avi pada form simulasi.
Input : Video bertipe *.mp4, *.3gp, *.avi
Output : Video bertipe *.mp4, *.3gp, *.avi
Deskripsi : Mengambil video bertipe *.mp4, *.3gp, *.avi yang telah disimpan pada komputer.
Source code untuk input video adalah sebagai berikut :

```
public static String Inputfile(){
    JFileChooser chooser = new JFileChooser("Documents:\\A\\dataset");
    FileNameExtensionFilter filter = new FileNameExtensionFilter(
            "MP4, AVI, and 3GP Videos", "mp4", "3gp", "avi");
    chooser.setFileFilter(filter);
    chooser.setDialogTitle("Pilih file video");
    chooser.showOpenDialog(null);
    File file = chooser.getSelectedFile();
    String tempat = file.getAbsolutePath();
    return tempat;
}
```

Gambar 4.28: *Source Code Input* Video

Adapun tampilan antar muka *input* video dapat dilihat pada Gambar 4.29.



Gambar 4.29: Halaman Antar Muka *Input* Video

4.4.2 Implementasi Pemilihan ROI

Region of Interest (ROI) adalah bagian dari *frame* atau citra yang didefinisikan sebagai objek yang akan dideteksi dan dilacak. Pembentukan ROI objek diperoleh dengan membentuk *rectangle*. ROI berfungsi untuk mempersempit ruang proses pengolahan pada video sehingga data yang diolah lebih fokus, mempercepat waktu komputasi, dan menghindari objek yang tidak

diinginkan seperti orang yang berjalan, pohon dan lain-lain. Berikut salah satu implementasi dari tahap pemilihan ROI objek pada Java.

Fungsi: Menentukan objek yang akan dilacak pergerakannya di setiap *frame*.
Input: *Frame* pertama dari urutan *frame*
Output: Citra ROI (koordinat x, y,*perpanjangan* x *dan* y)
Deskripsi: Mendapatkan citra dari objek yang akan dilacak.

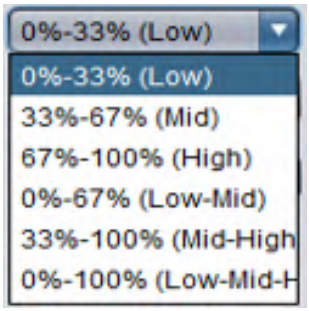
Source code untuk ROI objek adalah sebagai berikut :

```
if (!cbCustomROI.isSelected()) {
    int index = cbROI.getSelectedIndex();
    switch (index) {
        case 0: //low
            point2.y = 1 * (vidpanel.getHeight() / 3);
            break;
        case 1: //mid
            point2.y = 2 * (vidpanel.getHeight() / 3);
            break;
        case 2: //high
            point2.y = 3 * (vidpanel.getHeight() / 3);
            break;
        case 3: //low-mid
            point2.y = 2 * (vidpanel.getHeight() / 3);
            break;
        case 4: //mid-high
            point2.y = 3 * (vidpanel.getHeight() / 3);
            break;
        case 5: //low-mid-high
            point2.y = 3 * (vidpanel.getHeight() / 3);
            break;}}
```

Gambar 4.30: *Source Code* Pemilihan ROI

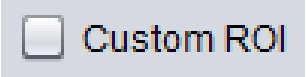
Dalam pemilihan daerah ROI, sistem akan menampilkan *image frame* awal saat *user* memasukkan video, kemudian *user* melakukan *click* dan *drag* pada panel video yang secara langsung akan diikuti oleh kotak biru sebagai tanda daerah ROI, terdapat dua jenis pemilihan daerah ROI yaitu sebagai berikut.

1. Secara *Default* yaitu pengguna hanya menentukan koordinat x_1 dan x_2 atau lebar dari kotak biru yang menandakan daerah ROI sedangkan koordinat y_1 dan y_2 ditentukan oleh sistem sesuai satu pilihan dari lima kemungkinan daerah ROI yang dapat disesuaikan, dimana tampilan antar muka pemilihan ROI secara *default* dapat dilihat pada Gambar 4.31.



Gambar 4.31: Tampilan Antar Muka Pemilihan ROI secara *Default*

2. Secara *Custom* yaitu pengguna menentukan koordinat x_1, y_1, x_2, dan y_2 secara manual. Berikut tampilan antar muka pemilihan ROI secara *custom* dapat dilihat pada Gambar 4.32.



Gambar 4.32: Tampilan Antar Muka Pemilihan ROI Secara *Custom*

Berikut contoh hasil dari pemilihan ROI yang dapat dilihat pada Gambar 4.33, sedangkan parameter yang digunakan dalam pemilihan ROI dapat dilihat pada Tabel 4.3.

Gambar 4.33: Salah Satu Tampilan Pemilihan ROI

Tabel 4.3: *Parameter Region of Interest* (ROI)

No.	Variabel	Tipe Data	Keterangan
1	X1	int	Koordinat x pada titik 1
2	Y1	int	Koordinat y pada titik 1
3	X2	int	Koordinat x pada titik 2
4	Y2	int	Koordinat y pada titik 2

4.4.3 Konversi Video menjadi Barisan *Frame*

Video yang digunakan sebagai *input* selanjutnya akan dirubah menjadi barisan *frame*. Berikut implementasi dari konversi video menjadi barisan *frame* pada Java.

Fungsi: *Input* video bertipe *.mp4, *.3gp, *.avi pada form simulasi.
Input: Video bertipe *.mp4, *.3gp, *.avi
Output: Urutan *frame* bertipe *.jpeg
Deskripsi: Mengkonversi video bertipe *.mp4, *.3gp, *.avi menjadi urutan gambar (*.jpeg)

Source code untuk input video adalah sebagai berikut :

```
if (vc.read(frames)) {
    try {
        Imgproc.resize(frames, frames, new Size(vidpanel.getWidth(),vidpanel.getHeight()));
        Imgproc.rectangle(frames, point1, point2, new Scalar(0, 255, 0), 3);
        frameROI = frames.submat(rect);
        frameROIdisplay=frames.submat(rect);
        contours = new ArrayList();
        System.out.println("Frame ke-" + frame_count);
        lblFrameCount.setText(String.valueOf(frame_count));
        second=(frame_count/fps);
        lblSeconds.setText(String.format("%.3f", second));

        if (cbProcess.getSelectedIndex()==0) {
                Tampil(frames);

                if (Save.isSelected()) {
                    Imgcodecs.imwrite(DirectSaveFrame +"\\"+ frame_count +".jpg",frames);}}
```

Gambar 4.34: *Source Code* Konversi Video menjadi Barisan *Frame*

4.4.4 Implementasi Deteksi Objek

Deteksi objek pada penelitian ini dilakukan dengan cara melakukan segmentasi objek yang meliputi beberapa tahapan diantaranya adalah *prepocessing*, *Background Subtraction* dengan GMM, *smoothing*, *shadow removal*, dan operasi morfologi. Berikut penjelasan dari implementasi masing-masing tahapan tersebut.

1. Implementasi *Preprocessing*
   Sebelum masuk pada tahap segmentasi objek dengan *Background Subtraction*, video di*prepocessing* terlebih dahulu dengan cara perbaikan ketajaman dan kecerahan yang bertujuan untuk mengurangi efek bayangan kendaraan yang terlalu tebal akibat cahaya matahari yang terlalu terang, yang mana akan menjadikan bayangan kendaraan terdeteksi menjadi suatu objek sehingga dapat mengurangi akurasi deteksi objek dan pelacakan objek. Berikut implementasi dari tahap *preprocessing* pada Java.

Fungsi: Objek tanpa bayangan yang terlalu tebal
Input: Urutan *frame* citra dan koordinat ROI
Output: *Preprocessed frame* citra
Deskripsi: *Preprocessing*

Source code untuk *preprocessing* adalah sebagai berikut:

```
start = System.currentTimeMillis();
alpha = Double.valueOf(txtAlpha.getText());
beta = Double.valueOf(txtBeta.getText());
frames.convertTo(frames, -1, alpha, beta);
```

Gambar 4.35: *Source Code Prepocessing*

sedangkan parameter-parameter yang digunakan proses *Prepocessing* pada proses simulasi dapat dilihat pada Tabel 4.4.

Tabel 4.4: Parameter Proses *Prepocessing*

No.	Variabel	Tipe Data	Keterangan
1	Alpha	Double	Nilai pengaturan kontras citra (*contrast adjustment*)
2	Beta	Double	Nilai pengaturan kecerahan citra (*brightness adjustment*)

Adapun contoh dari hasil *prepocessing* dapat dilihat pada Gambar 4.36.

Gambar 4.36: Contoh *Frame* Hasil *prepocessing*

2. Implementasi *Background Subtraction* dengan GMM
   Pada penelitian ini metode yang digunakan untuk *Background Subtraction* adalah metode GMM (*Gaussian Mixture Model*). Proses

Background Subtraction dengan metode GMM dibantu dengan *class* BackgroundSubtractorMOG2 dari *library* OpenCV. Hasil dari *class* tersebut adalah *image grayscale* dengan 3 tingkat keabuan yaitu $0, 127$, dan 255 yang merepresentasikan *background, shadow*, dan *foreground* secara berurutan. Berikut implementasi dari *Background Subtraction* dengan metode GMM pada Java.

Fungsi: Segmentasi objek.
Input: *Preprocessed frame* citra
Output: *Foreground image*
Deskripsi: *Background Subtraction* dengan metode GMM

Source code untuk *Background Subtraction* adalah sebagai berikut:

```
mBGSub.apply(frameROI, mFGMask, learning_rate);
```

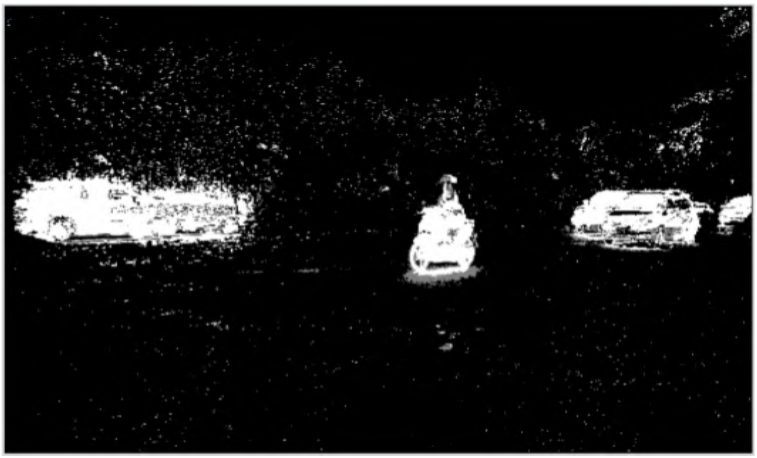
Gambar 4.37: *Source Code Background Subtraction* dengan GMM

Adapun Parameter yang digunakan pada proses *Background Subtraction* dengan GMM dapat dilihat pada Tabel 4.5.

Tabel 4.5: Parameter Proses *Background Subtraction* dengan GMM

No.	Variablel	Tipe Data	Keterangan
1	*Learning rate*	Double	Indikasi seberapa cepat BS mempelajari model

Berikut contoh hasil dari proses *Background Subtraction* dengan metode GMM dari program diberikan pada Gambar 4.38.



Gambar 4.38: Contoh *Frame* Hasil *Background Subtraction* dengan GMM

3. Implementasi *Smoothing*
   Untuk proses *smoothing* pada penelitian ini menggunakan metode Median Filter. Data masukan pada proses *smoothing* adalah *image grayscale* hasil dari proses *Background Subtraction*, sedangkan tipe data dari hasil keluaran proses *smoothing* juga merupakan *image grayscale*. Dalam OpenCV proses Median Filter dilakukan dengan memanggil fungsi medianBlur. Berikut Implementasi dari proses *smoothing* pada Java.

Fungsi: Mengurangi *noise*
Input: *Foreground image*
Output: *Image grayscale smoothing*
Deskripsi: *Smoothing* dengan metode Median Filter

Source code untuk *Smoothing* adalah sebagai berikut:

```
int ksize = Integer.parseInt(txtKSize.getText()).
Imgproc.medianBlur(mFGMask, mFGMask, ksize);
```

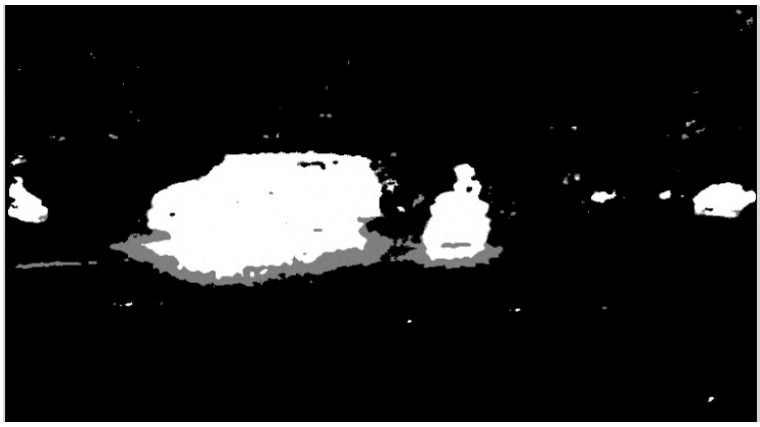
Gambar 4.39: *Source Code Smoothing*

Adapun parameter yang digunakan pada proses *smoothing* dapat dilihat pada Tabel 4.6.

Tabel 4.6: Parameter Proses *Smoothing* dengan Median Filter

No.	Variablel	Tipe Data	Keterangan
1	*Size*	Int	Jangkauan ketetanggaan piksel yang akan dicari nilai mediannya dan bernilai ganjil

Berikut contoh dari hasil *smoothing* dengan Median Filter yang dapat dilihat pada Gambar 4.40.



Gambar 4.40: Contoh *Frame* Hasil *Smoothing*

4. Implementasi *Shadow Removal*
   Salah satu permasalahan yang muncul pada proses *Background Subtraction* adalah munculnya bayangan (*shadow)* dan *ghost*, dimana *ghost* muncul akibat pergerakan kecil dari lingkungan seperti gerak daun yang tertiup angin atau akibat perubahan intensitas cahaya di dalam lingkungan (Solichin dan Harjoko, 2013), sehingga bayangan (*shadow)* dan *ghost* harus diminimalisir dengan *shadow removal*. Data masukan pada proses *shadow removal* berupa *image grayscale* hasil dari *smoothing*, dimana semua piksel yang mempunyai nilai bayangan pada *image grayscale* hasil dari *smoothing* akan dijadikan nilai *background*, sehingga hasil akhir dari proses *shadow removal* berupa *image* biner atau mempunyai dua nilai yaitu *background* dan *foreground*. Berikut implementasi dari *shadow removal* pada Java.

Fungsi: Memperoleh *image biner* untuk mengurangi bayangan (*noise*)
Input: *Image grayscale Smoothing*
Output: *Image biner*
Deskripsi: *Shadow removal*

Source code untuk *Shadow removal* adalah sebagai berikut:

```
Imgproc.threshold(mFGMask, mFGMask, 127, 255, Imgproc.THRESH_BINARY);
```

Gambar 4.41: *Source Code Shadow Removal*

Berikut contoh dari hasil dari *shadow removal* yang dapat dilihat pada Gambar 4.42.

Gambar 4.42: Contoh *Frame* Hasil *Shadow Removal*

5. Implementasi Operasi Morfologi
   Data masukan pada proses morfologi adalah *image biner* hasil dari *shadow removal*. Pada Gambar 4.42 hasil dari *shadow removal* memiliki

lubang (*holes*), sehingga mengganggu proses deteksi objek. Oleh karena itu dilakukan proses *filling holes* dengan menggunakan operasi morfologi *Closing* dengan tujuan untuk meningkatkan akurasi deteksi objek dan pelacakan objek. Berikut implementasi dari operasi morfologi pada Java.

Fungsi: Memperoleh objek lebih utuh dan lubang pada objek berkurang
Input: *Image biner*
Output: *Image biner* Morfologi
Deskripsi: Operasi morfologi dengan operasi *Closing*

Source code untuk Operasi morfologi adalah sebagai berikut:

```
int ksize = Integer.parseInt(txtMSize.getText());
SE = Double.valueOf(txtSE.getText());
if (SE==0) {
Mat kernel_Closing = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(ksize, ksize))
Imgproc.morphologyEx(mFGMask, mFGMask, Imgproc.MORPH_CLOSE, kernel_Closing);
Mat Morph=mFGMask.clone();
```

Gambar 4.43: *Source Code* Operasi Morfologi

Adapun parameter yang digunakan pada proses morfologi dengan operasi *Closing* dapat dilihat pada Tabel 4.7.

Tabel 4.7: Parameter Proses Morfologi dengan Operasi *Closing*

No.	**Variabel**	**Tipe Data**	**Keterangan**
1	*Size*	Int	Ukuran kernel yang akan digunakan untuk proses konvolusi morfologi operasi *closing* dan bernilai ganjil
2	SE	Int	Bentuk dari SE (*Structure element*)

Berikut contoh dari hasil operasi morfologi dengan operasi *Closing* yang dapat dilihat pada Gambar 4.44.

Gambar 4.44: Contoh *Frame* Hasil Operasi Morfologi

4.4.5 Implementasi Pelacakan Objek

Pada proses ini akan dilakukan pelacakan objek dengan metode Filter Korelasi dan Filter Kalman. Berikut penjabaran dari implementasi kombinasi metode Filter Korelasi dan Filter Kalman pada Java.
Fungsi : Memproses setiap *frame* untuk dilakukan pelacakan objek yang telah ditentukan
Input : Barisan *frame* hasil video yang telah diekstrak dan koordinat ROI
Output : Posisi baru dari objek yang dilacak
Deskripsi : Memproses setiap urutan *frame* untuk dilakukan pelacakan objek.

Berikut penjelasan mengenai setiap tahapan dalam proses pelacakan objek menggunakan metode kombinasi Filter Korelasi dan Kalman Filter.

1. Inisialisasi
   Langkah pertama dari metode Filter Korelasi ini diperoleh dari *input* video dengan koordinat ROI, deteksi objek, kemudian inisialisasi pelacakan objek untuk memperoleh koordinat x, y, *width* (x_v), dan *height* (y_v). Empat *input* tersebut digunakan untuk menentukan daerah kalkulasi pada setiap objek. Berikut implementasi dari inisialisasi pada proses pelacakan objek pada Java.

```
List<MatOfPoint> contours = new ArrayList<>();
Mat hierarchy = new Mat();
detect_obj = 1
Imgproc.findContours(mFGMask, contours, hierarchy, Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE);
listObjIn.clear();//untuk list id obj yang ada di roi , setiap frame akan di clear
```

Gambar 4.45: *Source Code* Inisialisasi Filter Korelasi

2. *Update* Model dan *Target Localization*
   Update Model objek dan penghitungan *centroid* dilakukan dengan metode Filter Korelasi seperti yang telah dijelaskan pada Bab 2. Berikut implentasi untuk *update* model dan mendeteksi objek di *frame* selanjutnya (*target localization*) pada Java.

```
if (listObj.isEmpty()) { //objek awal
    System.out.println("objek lainnya " + idObj);
    listObj.add(new Tracking.Obj(boundingBox, idObj));
    // *memanggil class Obj
    listObjIn.add(idObj);
}else {
    boolean check = false;
    for (Tracking.Obj obj : listObj) {
        Rect objRect = obj.getRect();
        if (center.inside(objRect) && !obj.isOut) {
            Point objTL=objRect.tl();

double w = (objRect.x + (objRect.width / 2)) *(objRect.y + (objRect.height / 2));//correlation
w = Luas;
Mat I = Mat.eye(4,4, CvType.CV_32F);
double output_sigma = Math.sqrt((objRect.x*objRect.y)/area*sigma);
double syh   = objRect.y/2;
double sxh = objRect.x/2;
double mult = -0.5 / (output_sigma*output_sigma);
double gausianShape = Math.exp(mult * (objRect.x+objRect.x));
double fz = Core.dft(w, x);
```

Gambar 4.46: *Source Code Update Model*

3. Prediksi dengan Filter Kalman
   Prediksi pergerakan objek memanfaatkan *output* dari metode Filter Korelasi berupa koordinat *centroid* dan koordinat *rectangle*, selanjutnya digunakan sebagai *input* untuk memprediksi dengan metode Filter Kalman. Inisialisasi awal dari Filter Kalman yaitu mendefinisikan matriks A, H, Q, dan R. Implementasi Filter Kalman pada OpenCV sudah tersedia, sehingga hanya membuat *class* untuk mendapatkan sifat-sifat atau atribut dari *class* induk. Berikut implementasi untuk prediksi dan koreksi gerakan dari objek dengan Filter Kalman pada Java.

- Inisialisasi

```
// inisialisai
LastResult = pt;
Mat statePre = new Mat(4, 1, CvType.CV_32F, new Scalar(0));
statePre.put(0, 0, pt.x);
statePre.put(1, 0, pt.y);
statePre.put(2, 0, 0);
statePre.put(3, 0, 0);
kalman.set_statePre(statePre);
Mat statePost = new Mat(4, 1, CvType.CV_32F, new Scalar(0));
statePost.put(0, 0, pt.x);
statePost.put(1, 0, pt.y);
statePost.put(2, 0, 0);
statePost.put(3, 0, 0);
kalman.set_statePost(statePost);
```

- Prediksi

```
public Point getPrediction() {
        Mat prediction = kalman.predict();
        LastResult = new Point(prediction.get(0, 0)[0], prediction.get(1, 0)[0]);
        return LastResult;
}
```

- Koreksi

```
public Point update(Point p, boolean dataCorrect) {
        Mat measurement = new Mat(2, 1, CvType.CV_32F, new Scalar(0)) ;
        if (!dataCorrect) {
                measurement.put(0, 0, LastResult.x);
                measurement.put(1, 0, LastResult.y);
        } else {
                measurement.put(0, 0, p.x);
                measurement.put(1, 0, p.y);
        }
// Koreksi
public Point correction(Point p){
        Mat measurement = new Mat(2, 1, CvType.CV_32F, new Scalar(0));
        measurement.put(0, 0, p.x);
        measurement.put(1, 0, p.y);

        Mat estimated = kalman.correct(measurement);
        LastResult.x = estimated.get(0, 0)[0];
        LastResult.y = estimated.get(1, 0)[0];
        return LastResult;
}
```

Gambar 4.47: *Source Code* Filter Kalman

Berikut tampilan dari pelacakan objek dengan kombinasi metode Filter Korelasi dan Filter Kalman yang dapat dilihat pada Gambar 4.48.

Gambar 4.48: Tampilan Pelacakan Objek dengan Metode FK dan FK

BAB 5
SIMULASI DAN ANALISIS HASIL

Pada bab ini dijelaskan mengenai simulasi dari perancangan proses pelacakan objek yang disebutkan pada bab sebelumya yang telah dibuat dan kemudian membahas hasil dari setiap simulasi. Simulasi yang pertama adalah pengujian pelacakan objek pada video tanpa oklusi dengan menggunakan metode Filter Korelasi dan kombinasi metode Filter Korelasi (FK) dan Filter Kalman (FK). Tujuan dari simulasi ini adalah untuk melihat tingkat persentase akurasi dari pelacakan objek tanpa adanya oklusi. Tahap kedua adalah pengujian pelacakan objek pada video dengan oklusi menggunakan metode Filter Korelasi dan kombinasi metode Filter Korelasi (FK) dan Filter Kalman (FK). Tujuan dari simulasi ini adalah untuk melihat persentase akurasi pelacakan objek dengan adanya oklusi pada kendaraan. Tahap terakhir dari simulasi adalah membandingkan rata-rata persentase akurasi antara metode Filter Korelasi dan metode kombinasi FK+FK. Simulasi yang dilakukan pada penelitian ini berorientasi pada keakuratan dan validasi hasil yang didapatkan dengan melihat visualisasi *frame* yang dapat dilacak secara benar yang diperoleh dari hasil simulasi pada program.

5.1 Data Simulasi

Simulasi pada program dilakukan pada video berekstensi mp4. Video simulasi telah tersimpan dalam penyimpanan komputer yang diperoleh dari akuisisi di jalan raya, penelitian sebelumnya, dan Youtube. Video yang digunakan terdiri dari dua jenis yaitu, video tanpa oklusi dan video dengan oklusi, dimana masing-masing jenis tersebut terdiri dari sepuluh video dengan sudut dan tempat pengambilan yang berbeda. Berikut dataset video yang digunakan untuk simulasi yang dapat dilihat pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1: Dataset Video Simulasi Pelacakan Objek Tanpa Oklusi

No.	**Nama Video**	**Jumlah *Frame***	**Jumlah Kendaraan**	**Pengambilan Video**
1	suramadu1	94	1	Atas depan
2	suramadu2	118	2	Atas depan
3	suramadu3	46	2	Atas depan
4	suramadu4	100	2	Atas depan
5	suramadu5	1020	> 50	Atas depan
6	jpo1	589	12	Atas depan
7	jpo2	359	16	Atas depan
8	zoom	1135	23	Atas depan
9	road	1609	> 50	Atas
10	khusus	462	1	Belakang

Tabel 5.2: Dataset Video Simulasi Pelacakan Objek dengan Adanya Oklusi

No.	**Nama Video**	**Jumlah *Frame***	**Jumlah Kendaraan**	**Pengambilan Video**	**Oklusi**
1	graha1	140	1	Samping	Sebagian
2	bettet	306	2	Atas samping	Total dan sebagian
3	stadion	273	2	Samping	Total dan sebagian
4	Diponegoro	930	> 50	Atas depan	Total dan sebagian
5	duaarah	1796	> 50	Depan dan belakang	Sebagian
6	tigaarah	1496	> 50	Depan dan belakang	Total dan sebagian
7	graha2	1810	> 50	Samping	Total dan sebagian
8	menitA	1500	> 50	Belakang	Total dan Sebagian
9	menit1B	1497	> 50	Depan	Total dan Sebagian
10	sore	1856	> 50	Samping	Total dan sebagian

dari Tabel 5.1 dan 5.2, data yang ada pada No. 10 merupakan kasus khusus, dimana data yang ada di Tabel 5.1 No. 10 merupakan video dengan kondisi kamera bergerak, sedangkan pada Tabel 5.2 No. 10 adalah video yang diambil saat sore hari sehingga terdapat *noise* yang disebabkan oleh lampu kendaraan.

5.2 Akurasi Simulasi Pelacakan Objek

Pada subbab ini dijelaskan mengenai cara menghitung persentase akurasi dari simulasi pelacakan objek yang dilakukan, yaitu

$$PA = \frac{JB}{JB + JS} \times 100\% \tag{5.1}$$

dimana PA adalah persentase akurasi, JB adalah jumlah *frame* yang terlacak benar, yaitu ketika objek berhasil ditandai secara presisi dengan *rectangle*, dan

JS adalah jumlah *frame* yang terlacak salah, yaitu ketika objek kurang presisi, objek terlacak dua kali, dan tidak presisi.

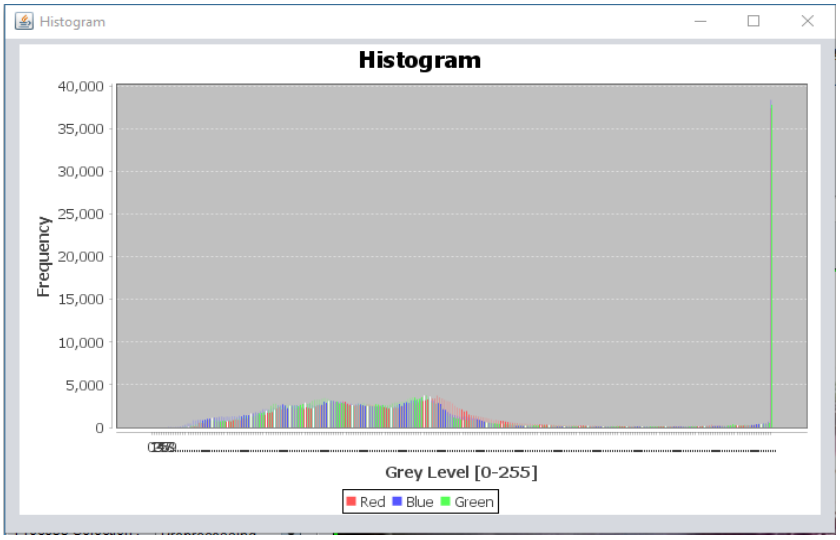
5.3 Simulasi Deteksi Objek

Sebelum masuk pada simulasi *tracking* (pelacakan), data masukan yang berupa video diproses dengan cara mendeteksi objek untuk mengetahui objek yang bergerak dengan cara memisahkan *background* dan *foreground*, sehingga memudahkan pada proses pelacakan objek. Pada proses deteksi objek, sebelum data video masuk ke dalam proses *Background Subtraction*, perlu dilakukan *preprocessing* terlebih dahulu, untuk mengatur ketajaman dan kecerahan dari suatu video. Pada penelitian ini, *preprocessing* dilakukan dengan cara *contrast and brightness anhacement* dengan tujuan untuk mengurangi risiko munculnya bayangan objek yang terlalu tebal akibat cahaya matahari yang terlalu terang, dimana bayangan objek yang terlalu tebal akan mengganggu proses deteksi objek karena dapat menjadi suatu penghubung dengan objek lain dan antara satu objek dengan objek lainnya akan terdeteksi sebagai satu objek karena piksel bayangan yang terdeteksi sebagai objek menyatukan kedua objek yang berbeda, sehingga menyebabkan terjadinya oklusi. Gambar 5.1 menunjukkan data video yang masuk ke dalam proses *Background Subtraction* tanpa melalui *preprocessing* sehingga ketika terjadi bayangan yang terlalu tebal maka akan mempengaruhi hasil dari deteksi objek.

(a) (b)

Gambar 5.1: (a). Citra Asal, (b). Citra Hasil BS tanpa *Preprocessing*

Adapun histogram dari Gambar 5.1 adalah sebagai berikut.



Gambar 5.2: Histogram dari Gambar 5.1

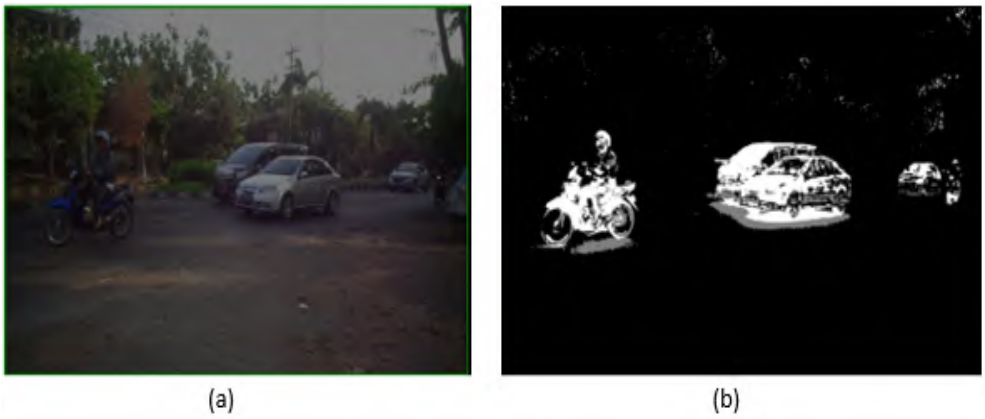
dari Gambar 5.1 dapat dilihat bahwa bayangan dari sekitar objek terlalu tebal sehingga untuk proses deteksi objek tidak cukup hanya dengan *Background Subtraction* akan tetapi, untuk hasil yang lebih baik perlu dilakukan *preprocessing*, *smoothing*, *shadow removal*, dan operasi morfologi. Hal ini dilakukan agar *bounding box* dapat optimal yang juga akan mempengaruhi titik *centroid*.

Pengaturan *contrast and brightness* dilakukan melalui parameter alpha dan beta dimana nilai alpha mempunyai nilai default 1 dan beta mempunyai nilai default 0 (tidak ada perubahan nilai piksel) dan nilainya dapat naik atau turun. Terdapat 8 kemungkinan kondisi nilai alpha dan beta, yaitu:

1. Nilai Alpha 2 dan Beta 0
   Saat nilai alpha naik maka akan meningkatkan *contrast* atau ketajaman warna, semakin tinggi intensitas piksel, maka akan semakin menuju ke warna putih, yangmana piksel dengan intensitas tinggi mempunyai frekuensi yang tinggi dan berakibat objek yang terdeteksi mendekati warna putih begitu juga dengan warna *background* sehingga bayangan akan semakin terlihat jelas dan objek tidak terdeteksi sempurna. Contoh hasil untuk kondisi ini dapat dilihat pada Gambar 5.3.

Gambar 5.3: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

2. Nilai Alpha 0.5 dan Beta 0
   Saat nilai alpha turun maka akan menurunkan *contrast* atau ketajaman warna, sehingga intensitas piksel akan semakin rendah dan akan menuju ke warna hitam serta berakibat objek yang terdeteksi dengan warna bayangan yang awalnya dominan ke hitam akan semakin hitam. Contoh hasil untuk kondisi ini dapat dilihat pada Gambar 5.4.

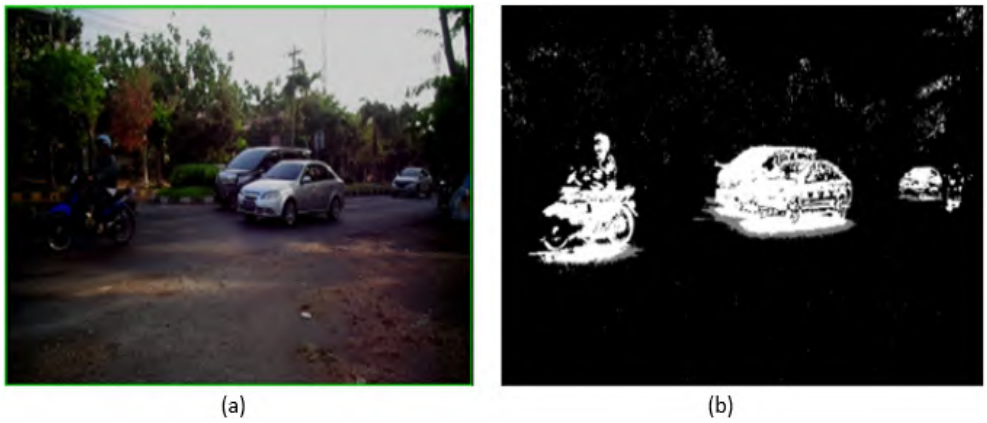


Gambar 5.4: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

3. Nilai Alpha 1 dan Beta 50
   Saat nilai beta naik maka akan meningkatkan *brightness* atau kecerahan warna, sehingga berakibat objek dapat terdeteksi, warna bayangan yang awalnya dominan ke hitam akan berkurang dan terlihat memudar. Contoh hasil untuk kondisi dapat dilihat pada Gambar 5.5.

Gambar 5.5: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

4. Nilai Alpha Tetap 1 dan Beta -50
   Saat nilai beta turun maka kecerahan warna juga menurun, sehingga objek terdeteksi namun warna bayangan yang awalnya dominan ke hitam akan semakin hitam dan terlihat lebih jelas. Contoh hasil dari kondisi ini dapat dilihat pada Gambar 5.6.



Gambar 5.6: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

5. Nilai Alpha 2 dan Beta 50
   Saat nilai alpha dan beta naik maka akan meningkatkan *contrast and brightness* dari video, sehingga mengakibatkan warna objek akan mendekati intensitas tertinggi yaitu warna putih dan bayangan memudar namun warna objek akan sama dengan *background* yang juga memutih. Contoh hasil dari kondisi ini dipat dilihat pada Gambar 5.7.

(a) (b)

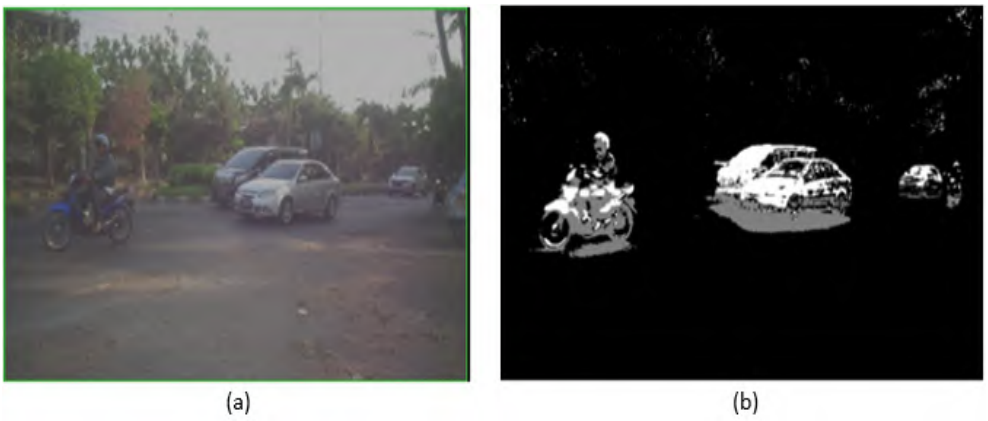
Gambar 5.7: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

6. Nilai Alpha 2 dan Beta -50
   Saat nilai alpha naik maka akan meningkatkan ketajaman warna dan ketika nilai beta turun maka akan menurunkan kecerahan warna, sehingga warna bayangan semakin tajam dan terlihat jelas. Contoh hasil dari kondisi ini dapat dilihat pada Gambar 5.8.

(a) (b)

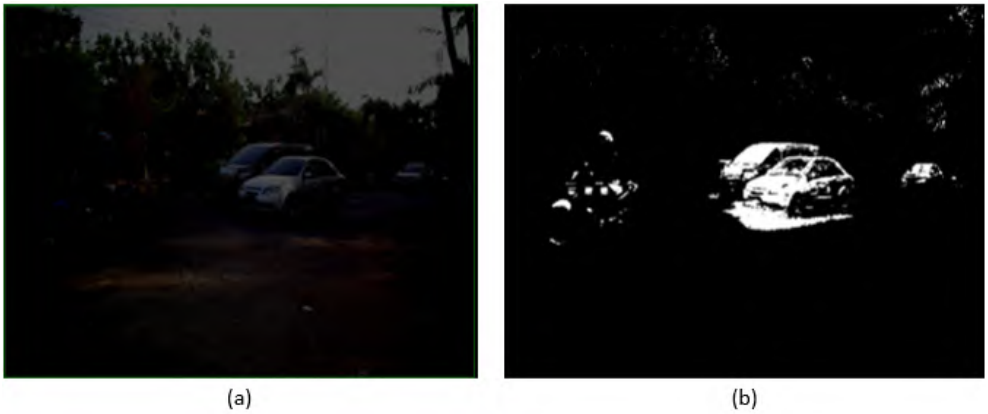
Gambar 5.8: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

7. Nilai Alpha 0.5 dan Beta 50
   Saat nilai alpha turun, maka akan menurunkan ketajaman warna dan ketika nilai beta naik maka akan meningkatkan kecerahan warna, sehingga warna objek memudar dan bayangan juga memudar. Contoh hasil dari kondisi ini dapat dilihat pada Gambar 5.9.



Gambar 5.9: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

8. Nilai Alpha 0.5 dan Beta −50
   Saat nilai alpha dan beta turun, maka akan menurunkan ketajaman dan kecerahan warna, sehingga mengakibatkan warna objek dan bayangan objek akan mendekati intensitas terendah yaitu hitam dan objek yang berwarna gelap tidak terdeteksi. Contoh hasil dari kondisi ini dapat dilihat pada Gambar 5.10.



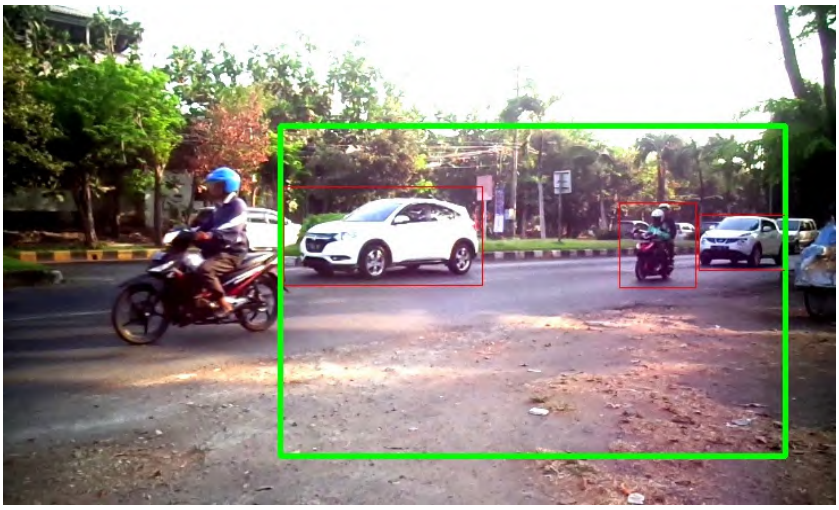
Gambar 5.10: (a). Citra Hasil *Preprocessing*, (b). Citra Hasil BS

Berdasarkan hasil simulasi dan analisis parameter alpha dan beta, untuk dapat mengurangi atau bahkan menghilangkan efek bayangan pada proses *Background Substraction*, maka perlu dilakukan kombinasi antara penurunan ketajaman warna dan peningkatan kecerahan warna dengan cara menurunkan nilai alpha dan meningkatkan nilai beta. Adapun deteksi objek yang optimal dalam penelitian ini adalah bentuk dan jumlah objek yang terdeteksi oleh sistem mendekati atau sama dengan bentuk dan jumlah objek sebenarnya. Deteksi objek merupakan serangkaian proses yang berurutan sehingga proses sebelumnya akan menjadi data untuk proses selanjutnya. Rangkaian proses dalam deteksi objek dapat dilihat pada Tabel 5.3.

Tabel 5.3: Rangkaian Proses Deteksi Objek

No.	Nama Proses	Parameter yang Digunakan	Tujuan	Output yang Diharapkan
1	*Preprocessing*	Alpha dan Beta	Mengatur ketajaman dan kecerahan	Objek tanpa bayangan yang terlalu tebal
2	*Smoothing*	*Size* Median Filter	Mengurangi *noise*	Objek dan *background* tanpa *noise*
3	Operasi Morfologi	*Size* Kernel Morfologi	Merekonstruksi objek	Bentuk objek lebih utuh, lubang dalam objek berkurang
4	Filter Area Objek	Minimum Area	Mengurangi risiko deteksi benda-benda kecil	Objek dengan ukuran lebih dari sama dengan minimum area

Berikut contoh bentuk deteksi objek sesuai dengan bentuk objek sebenarnya, dimana hasil dari deteksi objek tersebut kemudian di *labeling* dengan nilai alpha = 1 dan beta = 0 yang dapat dilihat pada Gambar 5.11

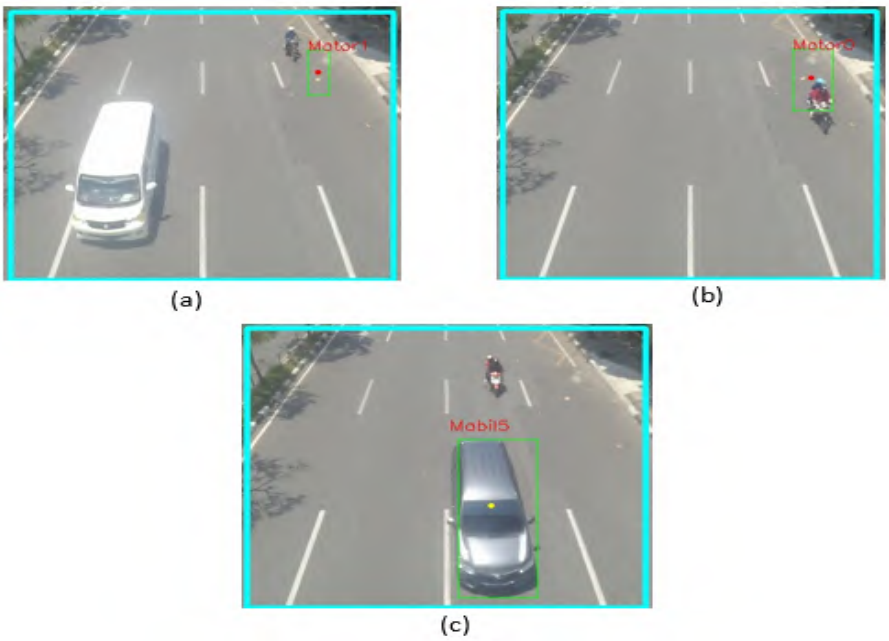


Gambar 5.11: Citra Hasil Deteksi Objek

5.4 Simulasi Pelacakan Objek

Langkah pertama untuk melakukan simulasi adalah *input* video, pemilihan ROI, ekstrak video menjadi barisan *frame*, deteksi objek kemudian dilakukan pelacakan objek. Hasil pelacakan objek berupa deretan *frame*,

dimana setiap *frame* terdapat objek yang ditandai dengan *rectangle* atau *boundingbox*. Pada penelitian ini diberikan 3 definisi tingkat ketepatan pelacakan objek yang dapat dilihat pada Gambar 5.12.



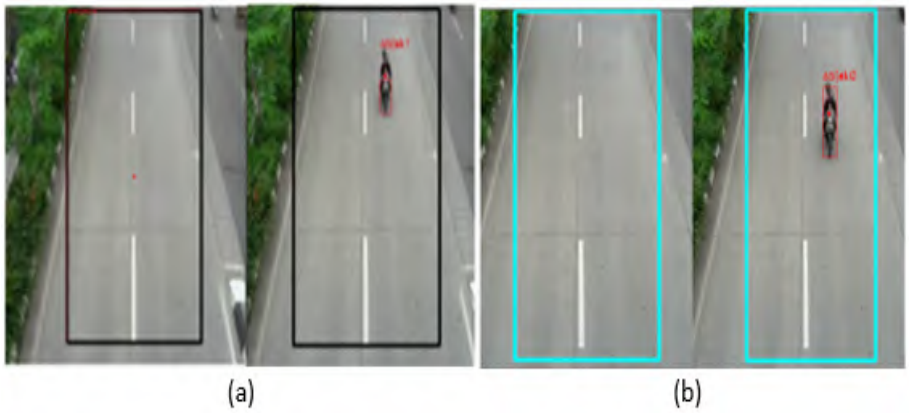
Gambar 5.12: (a). Tidak Presisi, (b). Kurang Presisisi, dan (c). Presisi

dari Gambar 5.12, sebuah pelacakan objek dikatakan presisi, jika sebagian besar objek masih berada dalam area *bounding box*. Suatu pelacakan objek dikatakan kurang presisi jika dalam area *bounding box* hanya terdapat sebagian kecil dari objek yang dilacak dan dikatakan tidak presisi jika tidak ada objek yang dilacak dalam area *bounding box*. Adapun simulasi dari pelacakan objek menggunakan kombinasi metode Filter Korelasi dan Filter Kalman adalah sebagai berikut.

5.4.1 Simulasi Pelacakan Objek Pada Video Tanpa Oklusi

Simulasi pelacakan objek tanpa oklusi menggunakan video yang telah disebutkan pada tabel 5.1, sehingga terdapat 10 sekenario simulasi yang disajikan sebagai berikut:

1. Simulasi pertama dilakukan pada video suramadu1.mp4, dimana video diambil dari atas depan dengan kondisi tanpa oklusi. Ada satu kendaraan yang berupa sepeda motor yang berada dalam ROI dan diperoleh hasil yang dapat dilihat pada Gambar 5.13 dan Tabel 5.4 sebagai berikut:



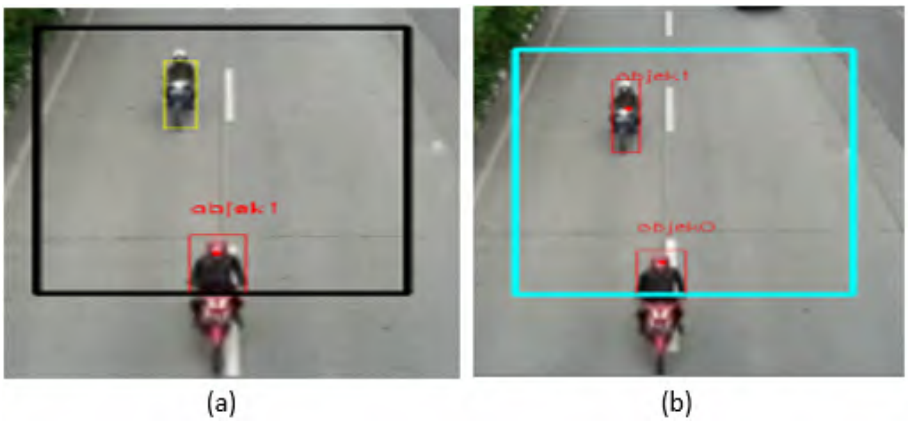
Gambar 5.13: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.4: Hasil Kinerja Pelacakan Objek

Nama Video	**Jumlah *Frame***	**JB**	**JS**	**Persentase Akurasi (PA)**	**Waktu**
Filter Korelasi					
suramadu1	94	88	6	94%	3 s
Filter Korelasi + Filter Kalman (Kombinasi)					
suramadu1	94	94	0	100%	3 s

dari simulasi pada video suramadu1.mp4 tersebut didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 100 %, sedangkan dengan metode Filter Korelasi diperoleh persentase akurasi sebesar 94 %, hal ini disebabkan dengan metode Filter Korelasi pada saat awal melacak program langsung mendeteksi objek yang salah sehingga terdapat beberapa *frame* yang terdefinisi salah. Adapun dengan penambahan metode Filter Kalman membuat hasil pelacakan lebih stabil sehingga tidak ada *frame* yang terdefinisi salah.

2. Simulasi ke-dua dilakukan pada video suramadu2.mp4, dimana video tersebut diambil dari atas depan dengan keadaan tanpa oklusi. Ada dua kendaraan yang berada dalam ROI yang terdiri dari sepeda motor, didapatkan hasil yang dapat dilihat pada Gambar 5.14 dan Tabel 5.5 sebagai berikut.



Gambar 5.14: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.5: Hasil Kinerja Pelacakan Objek

Nama Video	**Jumlah *Frame***	**JB**	**JS**	**Persentase Akurasi (PA)**	**Waktu**
Filter Korelasi					
suramadu2	118	114	4	96%	3 s
Filter Korelasi + Filter Kalman (Kombinasi)					
suramadu2	118	118	0	100%	3 s

dari simulasi pada video suramadu2.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik dikarenakan posisi pengambilan video dan kondisi ukuran objek saat mendekati jangkauan kamera relatif stabil. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 100%, sedangkan dengan metode Filter Korelasi diperoleh hasil 96%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi, sehingga dihasilkan pelacakan yang lebih presisi.

3. Simulasi ke-tiga dilakukan pada video suramadu3.*mp4, dimana video diambil dari atas depan dengan kondisi tanpa oklusi. Ada dua kendaraan yang berada dalam area ROI yang terdiri dari sepeda motor dan mobil, didapatkan hasil yang dapat dilihat pada Gambar 5.15 dan Tabel 5.6 sebagai berikut.

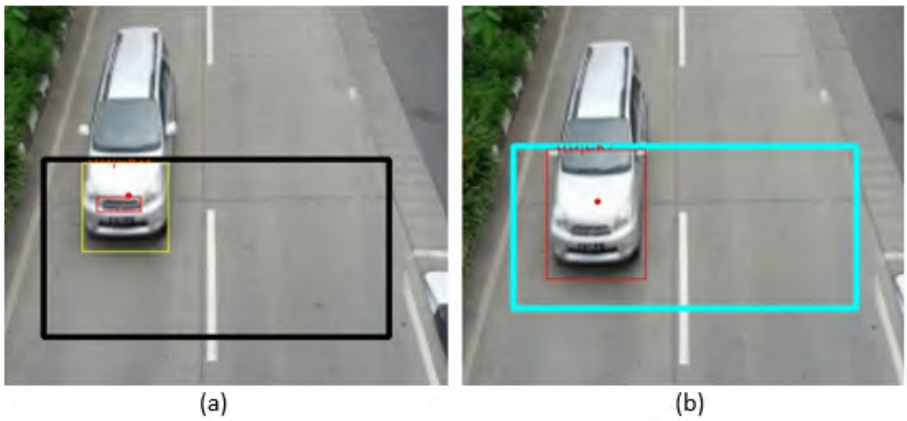
Gambar 5.15: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.6: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
suramadu3	46	43	3	93%	1 s
Filter Korelasi + Filter Kalman (Kombinasi)					
suramadu3	46	46	0	100%	1 s

dari simulasi pada video suramadu3.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan kondisi ukuran objek saat mendekati jangkauan kamera tidak stabil sehingga mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 100%, sedangkan dengan metode Filter Korelasi diperoleh hasil 96%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi, sehingga dihasilkan pelacakan yang lebih presisi.

4. Simulasi ke-empat, dilakukan pada video suramadu4.mp4, dimana video diambil dari atas depan dengan kondisi tanpa oklusi. Ada dua kendaraan yang berada dalam area ROI yang berupa dua buah mobil yang berjalan bergantian, didapatkan hasil yang dapat dilihat pada Gambar 5.16 dan Tabel 5.7 sebagai berikut.



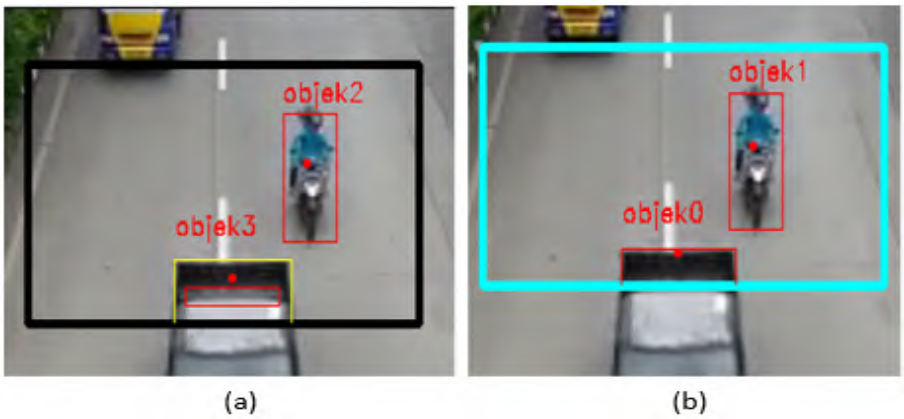
Gambar 5.16: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.7: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
suramadu4	100	98	2	98%	3 s
Filter Korelasi + Filter Kalman (Kombinasi)					
suramadu4	100	100	0	100%	3 s

dari simulasi pada video suramadu4.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan kondisi ukuran objek saat mendekati jangkauan kamera tidak stabil sehingga mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 100%, sedangkan dengan metode Filter Korelasi diperoleh hasil 98%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi sehingga diperoleh hasil pelacakan yang lebih presisi.

5. Simulasi ke-lima, dilakukan pada video suramadu5.mp4, dimana video diambil dari atas depan dengan kondisi tanpa oklusi. Ada > 50 kendaraan yang berada dalam area ROI, didapatkan hasil yang dapat dilihat pada Gambar 5.17 dan Tabel 5.8 sebagai berikut.



Gambar 5.17: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.8: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
suramadu5	1020	999	21	97%	42 s
Filter Korelasi + Filter Kalman (Kombinasi)					
suramadu5	1020	1017	3	99.7%	42 s

dari simulasi pada video suramadu5.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan, terdapat ukuran objek saat mendekati jangkauan kamera tidak stabil yang mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 99.7%, sedangkan dengan metode Filter Korelasi diperoleh hasil 97%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

6. Simulasi ke-enam, dilakukan pada video jpo1.mp4, dimana video diambil dari atas depan dengan jangkauan kamera yang lebih luas, bila dibandingkan dengan video yang digunakan pada simulasi 1-5 serta keadaan objek tanpa oklusi. Ada 12 kendaraan yang berada dalam area ROI, didapatkan hasil yang dapat dilihat pada Gambar 5.18 dan Tabel 5.9 sebagai berikut.

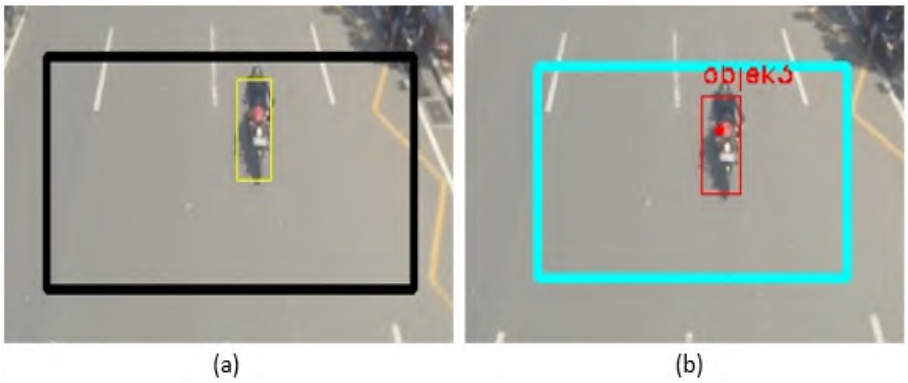
Gambar 5.18: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.9: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
jpo1	589	576	13	97.7%	20 s
Filter Korelasi + Filter Kalman (Kombinasi)					
jpo1	589	584	5	99%	20 s

dari simulasi pada video jpo1.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan kondisi ukuran objek saat mendekati jangkauan kamera tidak stabil yang mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi serta terdapat *noise* bayangan. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 99%, sedangkan dengan metode Filter Korelasi diperoleh hasil 97.7%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

7. Simulasi ke-tujuh, dilakukan pada video jpo2.mp4, dimana video diambil dari atas depan dengan jangkauan kamera yang lebih luas, bila dibandingkan dengan video yang digunakan pada simulasi 1-5 serta keadaan objek tanpa oklusi. Ada 16 kendaraan yang berada dalam area ROI, didapatkan hasil yang dapat dilihat pada Gambar 5.19 dan Tabel 5.10 sebagai berikut.



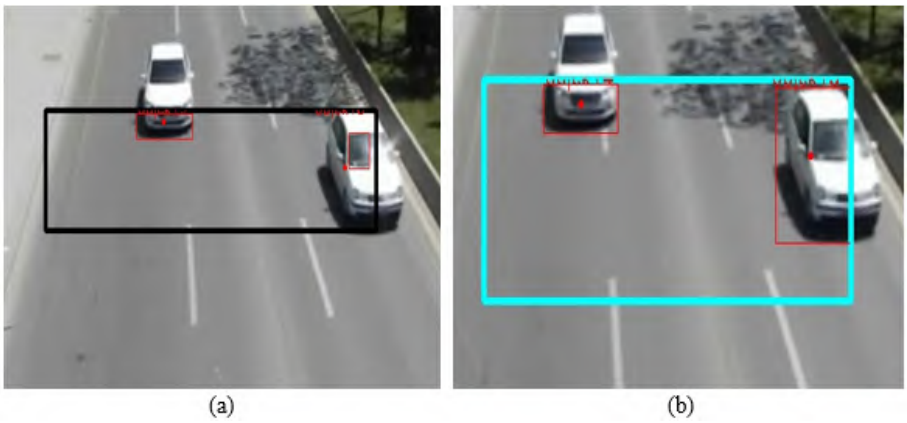
Gambar 5.19: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.10: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
jpo2	359	352	7	98%	12 s
Filter Korelasi + Filter Kalman (Kombinasi)					
jpo2	359	354	5	98.6%	12 s

dari simulasi pada video jpo2.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan kondisi objek yang jauh dari jangkauan kamera sering tidak terdeteksi, ukuran objek saat mendekati jangkauan kamera tidak stabil, serta terdapat *noise* bayangan dan cahaya yang mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 98.6%, sedangkan dengan metode Filter Korelasi diperoleh hasil 98%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi walaupun tidak begitu besar.

8. Simulasi ke-delapan, dilakukan pada video zoom.mp4, dimana video diambil dari atas dengan keadaan objek tanpa oklusi serta ukuran objek yang diperbesar (*zoom*). Ada 23 kendaraan yang berada dalam area ROI, didapatkan hasil yang dapat dilihat pada Gambar 5.20 dan Tabel 5.11 sebagai berikut.



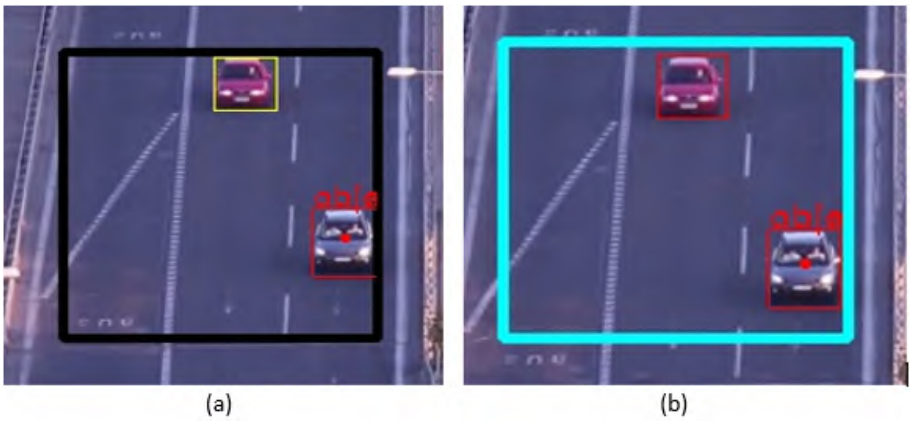
Gambar 5.20: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.11: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
zoom	1135	956	179	84%	39 s
Filter Korelasi + Filter Kalman (Kombinasi)					
zoom	1135	1025	15	90%	39 s

dari simulasi pada video zoom.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat simulasi dilakukan kondisi ukuran objek saat mendekati jangkauan kamera tidak stabil yang mengakibatkan objek berukuran terlalu besar dan sering menyebabkan *bounding box* tidak presisi dan kurang presisi serta terdapat *noise* bayangan. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 90%, sedangkan dengan metode Filter Korelasi diperoleh hasil 84%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

9. Simulasi ke-sembilan, dilakukan pada video road.mp4, dimana video diambil dari atas dengan jangkauan kamera yang lebih luas, bila dibandingkan dengan video yang digunakan pada simulasi 1-8 serta keadaan objek tanpa oklusi. Ada > 50 kendaraan yang berada dalam area ROI, didapatkan hasil yang dapat dilihat pada Gambar 5.21 dan Tabel 5.12 sebagai berikut.



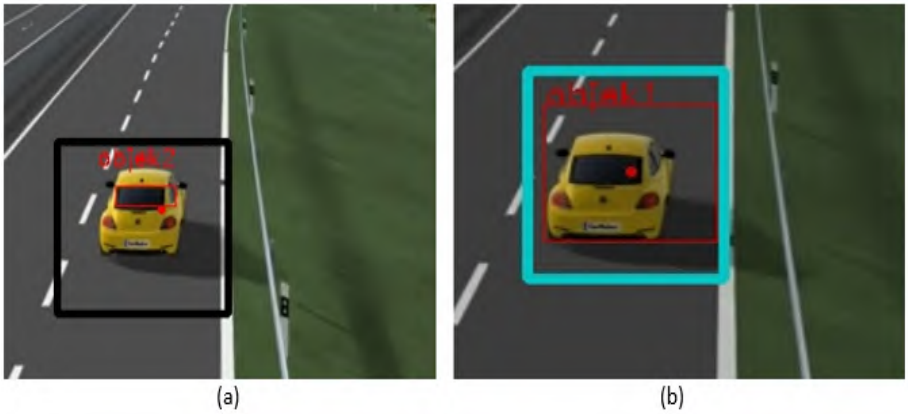
Gambar 5.21: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.12: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
road	1609	1320	289	82%	53 s
Filter Korelasi + Filter Kalman (Kombinasi)					
road	1609	1496	113	93%	53 s

dari simulasi pada video road.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat simulasi dilakukan, garis yang ada di jalan sering terdeteksi sebagai objek sehingga menyebabkan banyaknya *frame* yang terlacak salah. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 93%, sedangkan dengan metode Filter Korelasi diperoleh hasil 82%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

10. Simulasi ke-sepuluh, dilakukan pada video khusus.mp4, dimana video ini merupakan kondisi khusus. Ada 1 kendaraan yang berada dalam area ROI yang berupa mobil, dimana video tersebut dengan kondisi kamera bergerak sehingga mengikuti kemanapun objek tersebut berjalan, didapatkan hasil yang dapat dilihat pada Gambar 5.22 dan Tabel 5.13 sebagai berikut.



Gambar 5.22: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.13: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
khusus	462	235	227	51%	15 s
Filter Korelasi + Filter Kalman (Kombinasi)					
khusus	462	283	169	63%	15 s

dari simulasi pada video khusus.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat simulasi dilakukan, banyak *frame* yang terlacak salah akibat objek yang tidak terdeteksi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 63%, sedangkan dengan metode Filter Korelasi diperoleh hasil 51%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

5.4.2 Simulasi Pelacakan Objek Pada Video dengan Oklusi

Jenis Oklusi dalam pelacakan objek kendaraan terdiri dari dua jenis yang pertama oklusi oleh benda diam seperti oklusi yang disebabkan oleh pohon, lampu lalu lintas, tiang listrik, semak-semak, dan lain-lain. Kedua oklusi yang disebabkan oleh benda yang bergerak seperti antara satu kendaraan dengan kendaraan lainnya berjejer atau saling berhimpitan. Langkah pertama untuk melakukan simulasi sama seperti sub-bab sebelumnya, yaitu *input* video, pemilihan ROI, konversi video menjadi barisan *frame*. Pemilihan ROI bertujuan untuk mengetahui apakah perangkat lunak berhasil memperoleh citra objek yang akan dilacak atau tidak. Simulasi pelacakan objek dengan oklusi menggunakan video yang telah disebutkan pada tabel 5.2, sehingga terdapat 10 sekenario simulasi yang disajikan sebagai berikut:

1. Simulasi pertama dilakukan pada video graha1.mp4, dimana video diambil dari samping, ada satu kendaraan yang berada di ROI dan

kendaraan tersebut dalam kondisi berjalan namun tertutup oleh semak-semak sehingga bisa dikategorikan oklusi sebagian oleh benda diam, didapatkan hasil yang dapat dilihat pada Gambar 5.23 dan Tabel 5.14 sebagai berikut.

Gambar 5.23: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.14: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
graha1	140	136	4	97%	3 s
Filter Korelasi + Filter Kalman (Kombinasi)					
graha1	140	138	2	98.6%	3 s

dari simulasi pada video graha1.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat simulasi objek tertutup oleh semak-semak yang menyebabkan *bounding box* tidak sempurna sehingga terdapat beberapa pelacakan tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 98.6%, sedangkan dengan metode Filter Korelasi diperoleh hasil 97%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

2. Simulasi ke-dua dilakukan pada video bettet.mp4, dimana video diambil dari atas samping, ada dua kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh pohon sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda diam, didapatkan hasil yang dapat dilihat pada Gambar 5.24 dan Tabel 5.15 sebagai berikut.

Gambar 5.24: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.15: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
bettet	306	224	82	73%	10s
Filter Korelasi + Filter Kalman (Kombinasi)					
bettet	306	300	6	98%	10s

dari simulasi pada video bettet.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat dilakukan simulasi, objek tertutup oleh pohon yang menyebabkan *bounding box* tidak sempurna sehingga terdapat beberapa pelacakan tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 98%, sedangkan dengan metode Filter Korelasi diperoleh hasil 73%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

3. Simulasi ke-tiga dilakukan pada stadion.mp4, dimana video diambil dari samping, ada dua kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh tiang dan saling bersalipan sehingga bisa dikategorikan oklusi sebagian oleh benda diam dan keseluruhan oleh benda bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.25 dan Tabel 5.16 sebagai berikut.

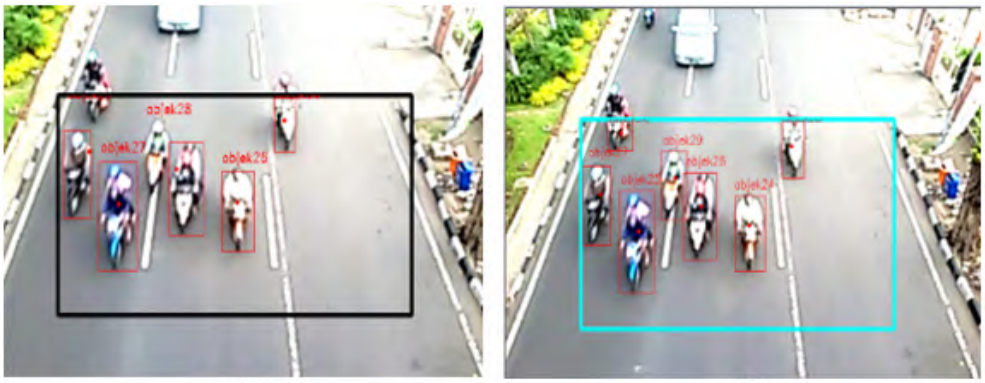
Gambar 5.25: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.16: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
stadion	273	271	2	99%	10s
Filter Korelasi + Filter Kalman (Kombinasi)					
stadion	273	273	0	100%	10s

dari simulasi pada video stadion.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan, objek tertutup oleh tiang dan objek saling bersalipan sehingga menyebabkan oklusi sebagian dan keseluruhan. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 100%, sedangkan dengan metode Filter Korelasi diperoleh hasil 99%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

4. Simulasi ke-empat dilakukan pada video Diponegoro.mp4, dimana video diambil dari atas depan, ada > 50 kendaraan yang berada dalam ROI dalam kondisi berjalan namun terkadang tertutup kendaraan lainnya, sehingga bisa dikategorikan oklusi sebagian oleh benda bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.26 dan Tabel 5.17 sebagai berikut.



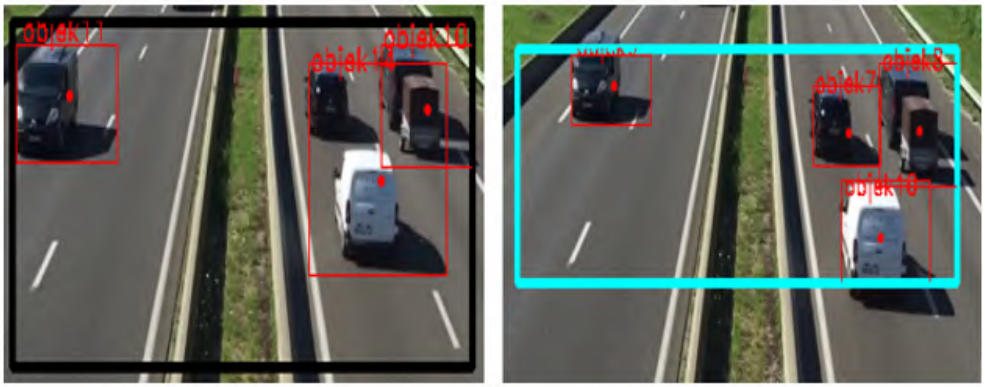
Gambar 5.26: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.17: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
Diponegoro	930	866	64	93%	31s
Filter Korelasi + Filter Kalman (Kombinasi)					
Diponegoro	930	908	22	97.3%	31s

dari simulasi pada video Diponegoro.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi dilakukan, tekadang objek tertutup oleh kendaraan lainnya sehingga terdapat beberapa pelacakan objek yang tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 97.3%, sedangkan dengan metode Filter Korelasi diperoleh hasil 93%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

5. Simulasi ke-lima dilakukan pada video duaarah.mp4, dimana video diambil dari depan dan belakang, dimana objek mendekati kamera dan menjauhi kamera. Simulasi tersebut digunakan untuk pelacakan kendaraan yang berada dalam area ROI dengan jumlah objek sebanyak > 50, didapatkan hasil yang dapat dilihat pada Gambar 5.27 dan Tabel 5.18 sebagai berikut.



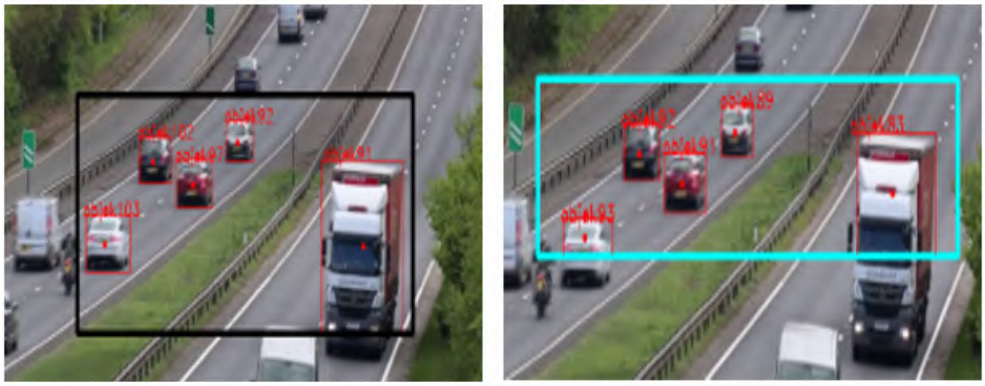
Gambar 5.27: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.18: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
duaarah	1796	1775	21	98.8%	59s
Filter Korelasi + Filter Kalman (Kombinasi)					
duaarah	1796	1782	14	99.2%	59s

dari simulasi pada video duaarah.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi objek tertutup kendaraan yang menyebabkan *bounding box* tidak sempurna sehingga terdapat beberapa pelacakan tidak presisi dan kurang presisi dan dapat dikategorikan oklusi sebagian dan total oleh benda bergerak. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 99.2%, sedangkan dengan metode Filter Korelasi diperoleh hasil 98.8%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

6. Simulasi ke-enam dilakukan pada video tigaarah.mp4, dimana video diambil dari depan dan belakang dengan tiga arah, ada > 50 kendaraan yang berada dalam ROI dengan kondisi berjalan namun tertutup oleh pohon dan kendaraan lainnya, sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda diam dan bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.28 dan Tabel 5.19 sebagai berikut.



Gambar 5.28: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.19: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
tigaarah	1496	1300	196	86.9%	49s
Filter Korelasi + Filter Kalman (Kombinasi)					
tigaarah	1496	1315	181	87.9%	49s

dari simulasi pada video tigaarah.mp4, didapatkan hasil bahwa program melakukan pelacakan tidak cukup baik, dikarenakan banyak objek yang mengalami oklusi, objek bergerak terlalu cepat dan warna objek yang menyerupai *background* sehingga banyak *frame* yang terlacak salah. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 87.9%, sedangkan dengan metode Filter Korelasi diperoleh hasil 86.9%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi walaupun tidak begitu besar pengaruhnya.

7. Simulasi ke-tujuh dilakukan pada video graha2.*mp4, dimana video diambil dari samping, ada > 50 kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh pohon dan kendaraan lainnya, sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda diam dan bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.29 dan Tabel 5.20 sebagai berikut.

Gambar 5.29: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.20: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
graha2	1810	682	1128	37%	59s
Filter Korelasi + Filter Kalman (Kombinasi)					
graha2	1810	712	1098	39%	59s

dari simulasi pada video graha2.mp4, didapatkan hasil bahwa program tidak dapat melakukan pelacakan dengan baik, dikarenakan banyaknya oklusi total pada objek dan posisi pengambilan video yang tidak tepat. Hal ini mengakibatkan pada saat pelacakan ukuran objek tidak stabil dan menjadi terdeteksi beberapa kali. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 39%, sedangkan dengan metode Filter Korelasi diperoleh hasil 37%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

8. Simulasi ke-delapan dilakukan pada video menitA.mp4, dimana video diambil dari belakang, ada > 50 kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh kendaraan lainnya sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.30 dan Tabel 5.21 sebagai berikut.

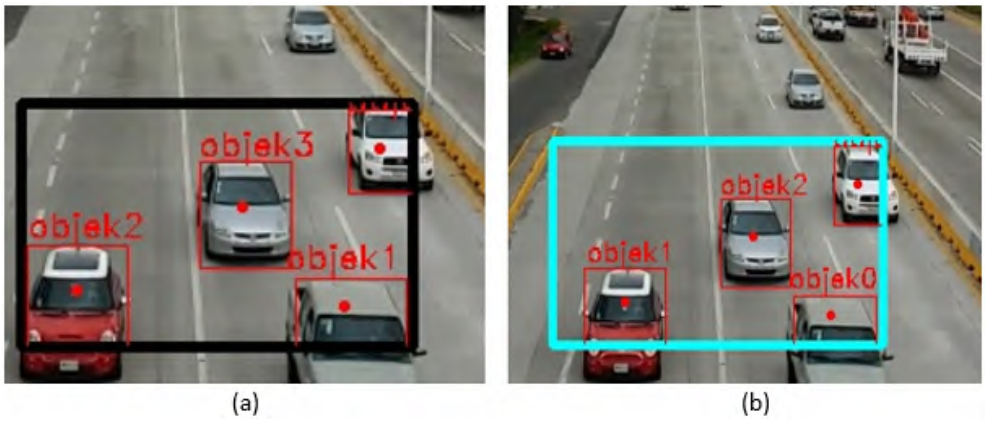
Gambar 5.30: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.21: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
menitA	1500	909	591	61%	59s
Filter Korelasi + Filter Kalman (Kombinasi)					
menitA	1500	1326	174	88%	59s

dari simulasi pada video menitA.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan cukup baik, walaupun saat simulasi objek tertutup oleh kendaraan lainnya yang menyebabkan *bounding box* tidak sempurna sehingga terdapat beberapa pelacakan tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 84%, sedangkan dengan metode Filter Korelasi diperoleh hasil 59%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

9. Simulasi ke-sembilan dilakukan pada video menit1B.mp4, dimana video diambil dari depan, ada $>$ 50 kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh kendaraan sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.31 dan Tabel 5.22 sebagai berikut.



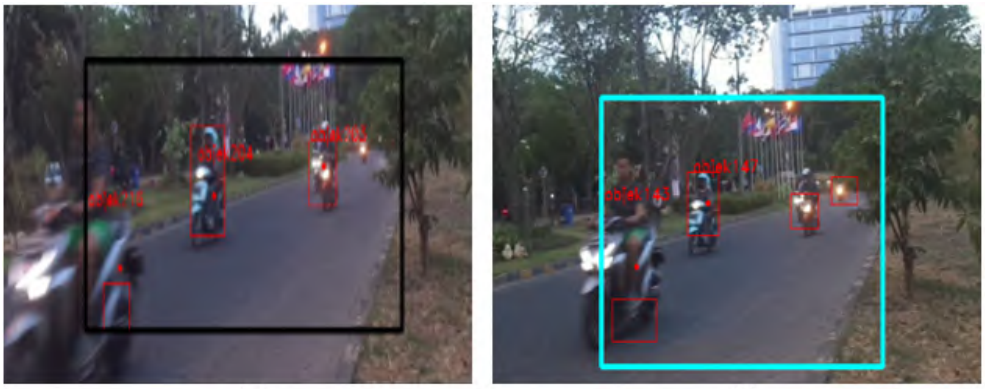
Gambar 5.31: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

Tabel 5.22: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
menit1B	1497	1377	120	91%	59s
Filter Korelasi + Filter Kalman (Kombinasi)					
menit1B	1497	1422	75	95%	59s

dari simulasi pada video menit1B.mp4, didapatkan hasil bahwa program dapat melakukan pelacakan dengan baik, walaupun saat simulasi objek tertutup oleh kendaraan lainnya dan ukuran objek tidak stabil saat objek mendekati jangkauan kamera yang menyebabkan *bounding box* tidak sempurna, sehingga terdapat beberapa pelacakan tidak presisi dan kurang presisi. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 96%, sedangkan dengan metode Filter Korelasi diperoleh hasil 93%. Penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

10. Simulasi ke-sepuluh dilakukan pada video sore.mp4 yang merupakan video dengan kondisi khusus, pengambilan video dari samping dan waktu sore hari, ada > 50 kendaraan yang berada dalam ROI dalam kondisi berjalan namun tertutup oleh pohon dan kendaraan lainnya sehingga bisa dikategorikan oklusi sebagian dan keseluruhan oleh benda diam dan bergerak, didapatkan hasil yang dapat dilihat pada Gambar 5.32 dan Tabel 5.23 sebagai berikut.



Gambar 5.32: (a). Citra Hasil Pelacakan Objek dengan Metode Fk, (b). Citra Hasil Pelacakan Objek dengan Metode FK+FK

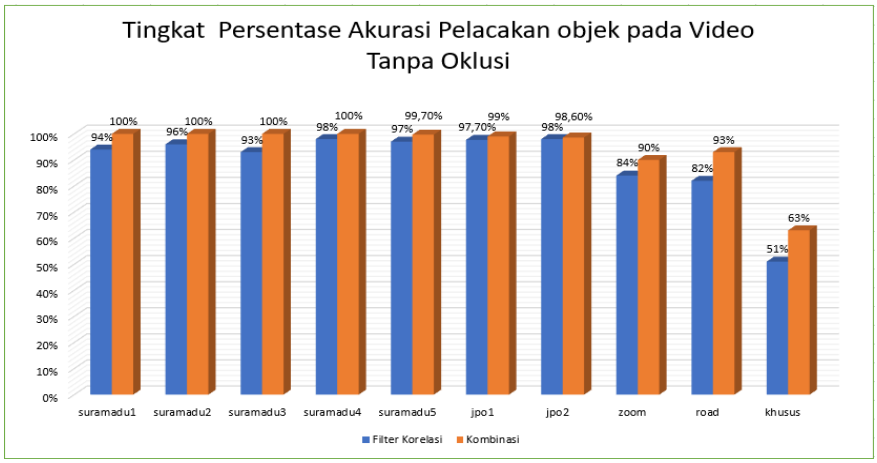
Tabel 5.23: Hasil Kinerja Pelacakan Objek

Nama Video	Jumlah *Frame*	JB	JS	Persentase Akurasi (PA)	Waktu
Filter Korelasi					
sore	1856	755	1101	40%	61s
Filter Korelasi + Filter Kalman (Kombinasi)					
sore	1856	755	1101	40%	61s

dari simulasi pada video sore.mp4, didapatkan hasil bahwa program tidak dapat melakukan pelacakan dengan baik, dikarenakan posisi pengambilan video yang tidak tepat dan lampu dari kendaraan yang menyebabkan terjadinya pelacakan yang kurang presisi. Hal ini mengakibatkan pada saat pelacakan ukuran objek tidak stabil dan menjadi terdeksi beberapa kali. Simulasi dengan metode kombinasi menghasilkan persentase akurasi sebesar 40%, sedangkan dengan metode Filter korelasi diperoleh hasil 40%. Penambahan metode Filter Kalman tidak memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

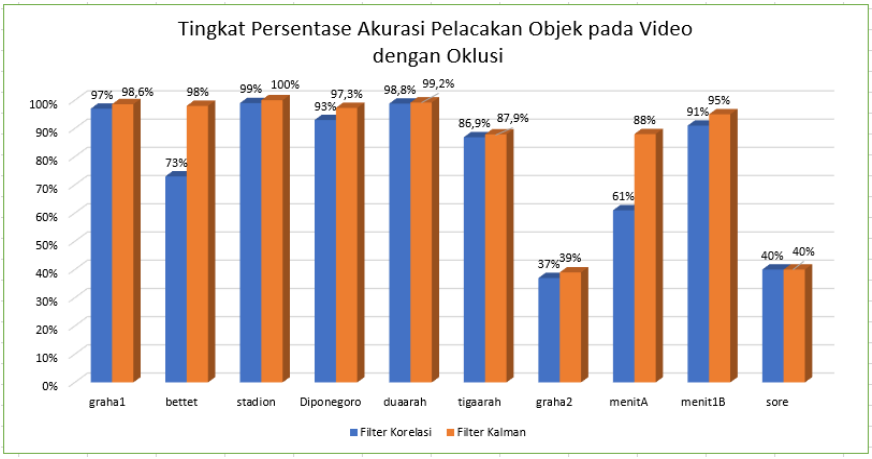
5.5 Pembahasan Hasil Pengujian

Pembahasan pada sub-bab ini difokuskan pada tingkat akurasi pelacakan dengan metode Filter Korelasi dan metode kombinasi (Filter Korelasi+Filter Kalman). Hasil pelacakan memberikan hasil akurasi pelacakan sama atau lebih bagus pada setiap metode. Berikut disajikan perbandingan persentase akurasi pelacakan dengan metode Filter Korelasi dan metode kombinasi.



Gambar 5.33: Tingkat Persentase Akurasi Pelacakan Objek pada Video Tanpa Oklusi

dari Gambar 5.33, dapat dilihat dari persentase akurasi masing-masing video, bahwa hasil pelacakan dengan metode kombinasi menghasilkan akurasi yang lebih tinggi bila dibandingkan dengan metode Filter Korelasi, sehingga penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak objek dengan metode Filter Korelasi.



Gambar 5.34: Tingkat Persentase Akurasi Pelacakan Objek pada Video dengan Oklusi

dari Gambar 5.34, dapat dilihat dari persentase akurasi masing-masing video, penambahan metode Filter Kalman mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi, hanya saja untuk video graha2.mp4 dan sore.mp4 hasil akurasi pelacakan relatif kecil atau terjadi penurunan akurasi disebabkan oleh arah pengambilan video yang tidak tepat yaitu diambil dari samping sehingga menyebabkan penskalaan objek yang tidak stabil saat dilacak. Adapun pada video sore.mp4 selain disebabkan oleh arah pengambilan video, penurunan akurasi juga disebabkan oleh waktu

pengambilan video yang tidak tepat yaitu sore hari sehingga banyak *noise* dari lampu kendaraan, serta penambahan metode Filter Kalman tidak mampu memperbaiki kesalahan saat melacak dengan metode Filter Korelasi.

Dari berbagai video yang telah dilakukan simulasi yangmana disebutkan pada Tabel 5.1 dan 5.2, berikut rata-rata persentasi dari hasil pelacakan pada video tanpa oklusi dan video dengan oklusi baik menggunakan metode Filter Korelasi maupun metode kombinasi Filter Korelasi dan Filter Kalman.

1. Rata - Rata Persentase Akurasi pada Video Tanpa Oklusi
   Rata-rata persentase akurasi pelacakan dengan metode Filter Korelasi dan metode kombinasi dapat dilihat pada Tabel 5.24 dan Tabel 5.25.

Tabel 5.24: Rata-Rata Persentase Akurasi dengan Metode Filter Korelasi

No.	**Nama Video**	**Jumlah *Frame***	**JB**	**JS**	**PA** (%)
1	suramadu1	94	88	6	94
2	suramadu2	118	114	4	96
3	suramadu3	46	43	3	93
4	suramadu4	100	98	2	98
5	suramadu5	1020	999	21	97
6	jpo1	589	576	13	97.7
7	jpo2	359	352	7	98
8	zoom	1135	956	179	84
9	road	1609	1320	289	82
10	khusus	462	235	227	51
11	Rata-rata				89

Tabel 5.25: Rata-Rata Persentase Akurasi dengan Metode Kombinasi

No.	**Nama Video**	**Jumlah *Frame***	**JB**	**JS**	**PA** (%)
1	suramadu1	94	94	0	100
2	suramadu2	118	118	0	100
3	suramadu3	46	46	0	100
4	suramadu4	100	100	0	100
5	suramadu5	1020	1017	3	99.7
6	jpo1	589	584	5	99
7	jpo2	359	354	5	98.6
8	zoom	1135	1025	15	90
9	road	1609	1496	113	93
10	khusus	462	283	169	63
11	Rata-rata				94

berdasarkan Tabel 5.24 dan 5.25, rata-rata persentase akurasi pelacakan

diperoleh sebesar 89% dengan metode Filter Korelasi dan 94% dengan metode kombinasi.

2. Rata - Rata Persentase Akurasi pada Video dengan Oklusi
   Rata-rata persentase akurasi pelacakan dengan metode Filter Korelasi dan metode kombinasi dapat dilihat pada Tabel 5.26 dan Tabel 5.27.

Tabel 5.26: Rata-Rata Persentase Akurasi dengan Metode Filter Korelasi

No.	**Nama Video**	**Jumlah *Frame***	**JB**	**JS**	**PA** (%)
1	graha1	140	136	4	97
2	bettet	306	224	82	73
3	stadion	273	271	2	99
4	Diponegoro	930	866	64	93
5	duaarah	1796	1775	21	98.8
6	tigaarah	1496	1300	196	86.9
7	graha2	1810	682	1128	37
8	menitA	1500	909	591	61
9	menit1B	1497	1377	120	91
10	sore	1856	755	1101	40
11	Rata-rata				78

Tabel 5.27: Rata-Rata Persentase Akurasi dengan Metode Kombinasi

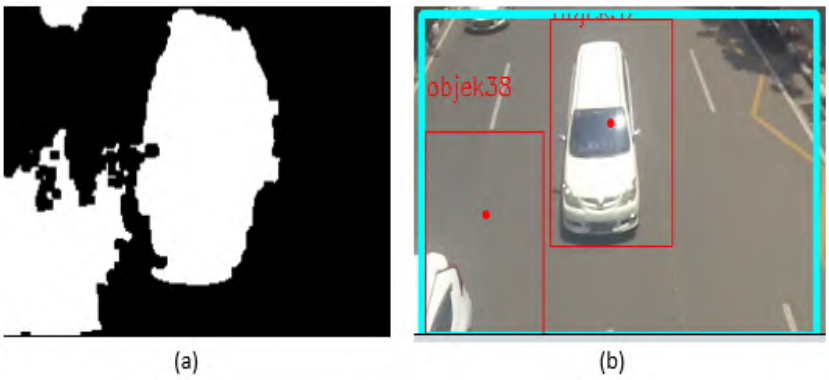
No.	**Nama Video**	**Jumlah *Frame***	**JB**	**JS**	**PA** (%)
1	graha1	140	138	2	98.6
2	bettet	306	300	6	98
3	stadion	273	273	0	100
4	Diponegoro	930	908	22	97.3
5	duaarah	1796	1782	14	99.2
6	tigaarah	1496	1315	181	87.9
7	graha2	1810	712	1098	39
8	menitA	1500	1326	174	88
9	menit1B	1497	1422	75	95
10	sore	1856	755	1101	40
11	Rata-rata				84.3

berdasarkan Tabel 5.26 dan Tabel 5.27, rata-rata persentase akurasi pelacakan diperoleh sebesar 78% dengan metode Filter Korelasi dan 84.3% dengan metode kombinasi.

5.6 Pembahasan Penyebab Besar Kecilnya Kinerja

Penyebab besar kecilnya prosentase kinerja disebabkan oleh beberapa hal sebagai berikut :

1. Pencahayaan.
   Pencahayaan adalah faktor paling penting karena hasil yang diberikan sangat bergantung pada pencahayaan, selain itu perubahan intensitas cahaya yang masuk ke kamera yang diakibatkan oleh pantulan cahaya oleh kaca kendaraan atau perubahan intensitas cahaya matahari dapat mempengaruhi kinerja dari pelacakan objek karena perubahan intensitas citra yang tinggi secara tiba-tiba. Pencahayaan dalam penelitian ini juga bergantung dari waktu pengambilan video, seperti video malam, siang, dan sore. Hasil pelacakan dan deteksi objek dengan faktor pantulan cahaya ke kamera disajikan pada Gambar 5.35.



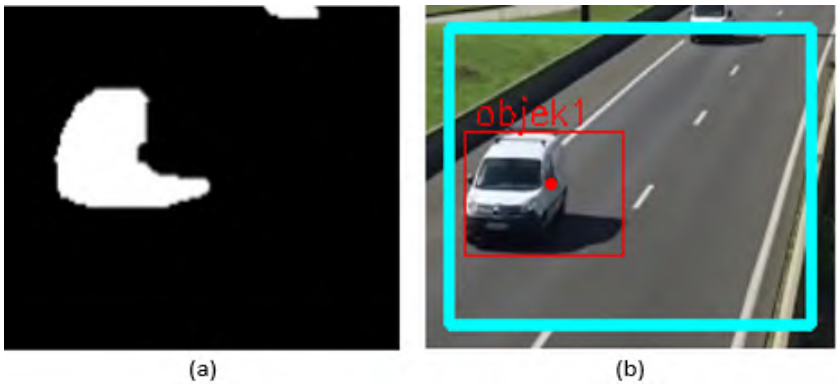
Gambar 5.35: Pengaruh Pencahayaan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

Adapun pengaruh dari waktu pengambilan video dapat dilihat pada Gambar 5.36, dimana pengambilan video tersebut dilakukan pada sore hari.

Gambar 5.36: Pengaruh Waktu Pengambilan Video pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

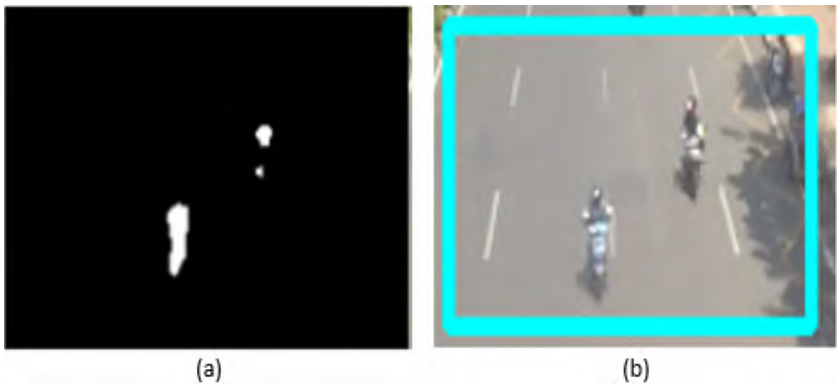
dari Gambar 5.35 dan 5.36 dapat dilihat bahwa pengaruh dari intensitas cahaya dapat menyebabkan pelacakan objek menjadi kurang presisi.

2. Bayangan.
   Bayangan memberikan gangguan yang signifikan pada *tracking*. Seperti bayangan dari objek kendaraan yang melintas menyebabkan terbentuknya kumpulan *noise* dengan ukuran piksel yang tidak sesuai dengan batasan piksel dari objek yang sebenarnya, sehingga terkadang menyebabkan pelacakan objek menjadi kurang presisi bahkan jika terdapat bayangan dari objek selain objek kendaraan juga terdeteksi sebagai objek kendaraan. Hasil pelacakan dan deteksi objek dengan faktor bayangan yang menyebabkan terjadinya *noise* dapat dilihat pada Gambar 5.37.



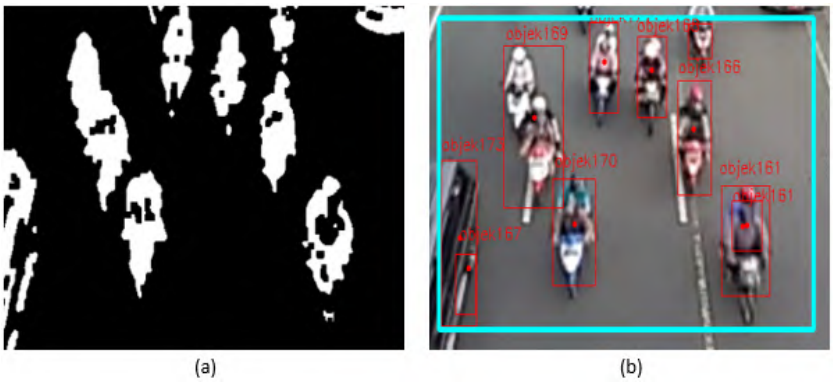
Gambar 5.37: Pengaruh Bayangan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

3. Warna Kendaraan.
   Warna kendaraan yang menyerupai *background* akan terdeteksi sebagai *background*, dan jika warna kendaraan antara satu kendaraan dengan kendaraan lainnya sama menyebabkan objek tidak terdeteksi sempurna. Hasil deteksi objek dengan faktor warna kendaraan dapat dilihat pada Gambar 5.38.



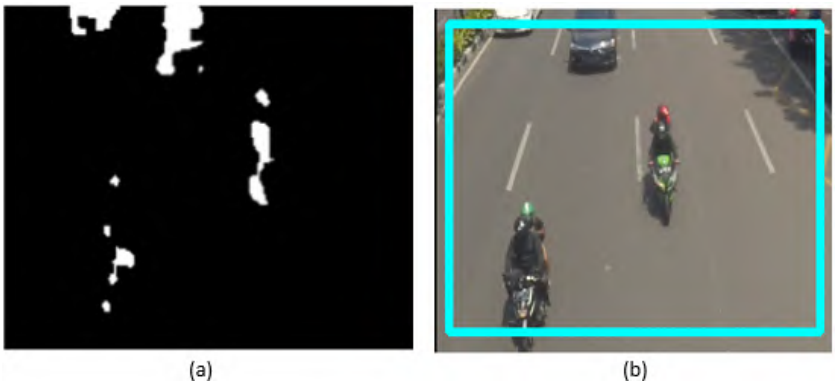
Gambar 5.38: Pengaruh Warna Kendaraan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

4. Kepadatan Arus Lalu Lintas.
   Arus lalu lintas yang cukup padat dengan kecepatan rendah menghasilkan bertumpuknya kendaraan dan berhimpitan, sehingga kendaraan yang berhimpit tersebut dikenali sebagai satu kendaraan yang utuh atau menyebabkan terjadinya oklusi dan hal tersebut juga menyebabkan objek terdeteksi dua kali serta hasil pelacakan kurang bahkan tidak presisi. Hasil pelacakan dan deteksi objek dengan faktor kepadatan arus lalu lintas disajikan pada Gambar 5.39.



Gambar 5.39: Pengaruh Kepadatan Arus Lalu Lintas pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

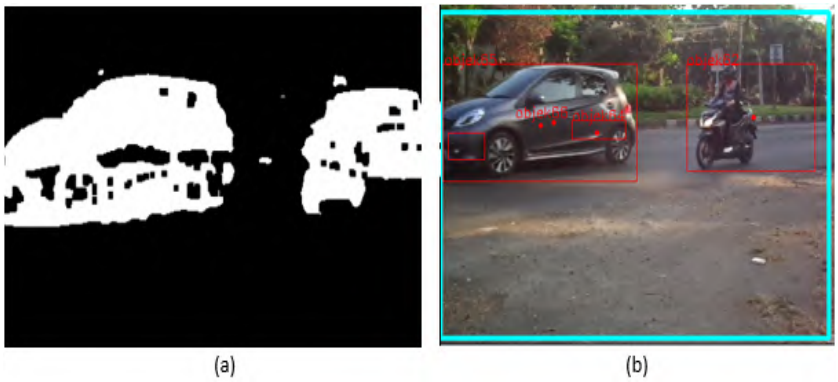
5. Kecepatan Kendaraan.
   Kendaraan yang melesat sangat cepat memberikan masalah pada proses *tracking*, dimana objek tidak terdeteksi sehingga sulit untuk dilacak dan menyebabkan hasil pelacakan tidak presisi. Hasil Pelacakan dan deteksi objek dengan faktor kecepatan kendaraan dapat dilihat pada Gambar 5.40.



Gambar 5.40: Pengaruh Kecepatan Kendaraan pada Kinerja Pelacakan Objek. (a). Segmentasi Objek, (b). Pelacakan Objek

6. Sudut Pengambilan Video
   Sudut pengambilan video sangat berpengaruh karena dengan sudut

pandang kamera yang berbeda menghasilkan perspekstif yang berbeda sehingga mempengaruhi ukuran kendaraan. Hal tersebut juga mengakibatkan hasil pelacakan objek kurang presisi, tidak presisi, dan menjadikan objek terlacak beberapa kali, serta area penghitungan saat *frame* i yang digunakan sebagai acuan untuk *frame* $i + 1$ berbeda bentuk objeknya. Hasil Pelacakan dan deteksi objek dengan faktor sudut pengambilan video dapat dilihat pada Gambar 5.41.



Gambar 5.41: Pengaruh Sudut Pengambilan Video pada Kinerja Pelacakan objek. (a). Segmentasi Objek, (b). Pelacakan Objek

BAB 6
KESIMPULAN DAN SARAN

Dari analisis dan pembahasan yang sudah dilakukan, dapat ditarik kesimpulan serta saran untuk pengembangan dan perbaikan penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil simulasi yang telah dilakukan, dapat disimpulkan bahwa penelitian ini telah berhasil mengkombinasikan metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak pada beberapa video baik yang bersifat *single object* dan *multiple objects*, serta video dengan oklusi maupun tanpa oklusi. Pelacakan dilakukan dengan beberapa tahapan, yaitu: *input* video, pemilihan ROI, ekstrak video menjadi barisan *frame*, deteksi objek, pelacakan objek dengan Filter Korelasi, dan prediksi objek dengan Filter Kalman.

Dari hasil simulasi pelacakan yang dilakukan pada sepuluh video tanpa oklusi dan sepuluh video dengan oklusi yang disebut pada Tabel 5.1 dan 5.2, dihasilkan rata-rata persentase akurasi sebesar 89% dengan metode Filter Korelasi dan 94% dengan metode kombinasi pada video tanpa oklusi, sedangkan rata-rata persentase akurasi pada video dengan oklusi diperoleh sebesar 78% dengan metode Filter Korelasi dan 84.3% dengan metode kombinasi. Hasil persentase akurasi tersebut berdasarkan simulasi yang dilakukan sepuluh kali percobaan pada masing-masing video, baik video tanpa oklusi maupun video dengan oklusi menggunakan metode Filter Korelasi dan metode kombinasi. Kombinasi metode Filter Korelasi dan Filter Kalman untuk pelacakan kendaraan bergerak dapat meningkatkan persentase akurasi pelacakan kendaraan baik untuk video tanpa oklusi maupun dengan oklusi, hanya saja untuk video dengan oklusi belum mampu sepenuhnya memperbaiki kesalahan saat terjadi oklusi secara keseluruhan hal ini disebabkan karena metode Filter Korelasi pada primal domain dan metode Filter Kalman cenderung digunakan pada pergerakan objek yang linear, sehingga akan sulit mendeteksi saat kehilangan objek. Adapun waktu komputasi baik menggunakan metode Filter Korelasi dan metode Kombinasi tidak ada perbedaan.

6.2 Saran

1. Metode Filter Korelasi hanya bekerja pada masalah primal domain, bisa dikembangkan ke dual domain, kernel korelasi, dan adaptive Filter Korelasi, sehingga bisa menangani pergerakan objek yang tidak linear dan penanganan oklusi total.

2. Pada penelitian ini penerapan Metode Filter Korelasi dan Filter Kalman hanya sampai pada masalah pelacakan dan untuk memudahkan dalam perhitungan kendaraan, sehingga untuk penelitian selanjutnya bisa dikembangkan sampai masalah otomasi *counting* dan klasifikasi kendaraan baik menggunakan *machine learning* atau *deep learning* seperti SVM, CNN dan lain-lain.

3. Hal yang sangat mempengaruhi akurasi pelacakan adalah waktu dan sudut pandang kamera dalam pengambilan video, sehingga jika ingin memperoleh akurasi yang lebih baik, maka waktu dan sudut pengambilan video benar-benar diperhatikan.

4. Metode Filter Korelasi bersifat pelacakan visual yaitu hanya untuk pelacakan satu objek maka untuk akurasi terbaik bahkan saat oklusi pada pelacakan *multiple objects* bisa dikombinasikan dengan *deep learning*.

DAFTAR PUSTAKA

Amaluddin, F., Muslim, M.A., Naba, A. (2015), "Klasifikasi Kendaraan Menggunakan Gaussian Mixture Model (GMM) dan Fuzzy Cluster Mean (FCM)", Jurnal EECCIS Vol. 9, No. 1.

Badan Pusat Statisitik (2018), *Statisitik Transportasi Darat 2017*, ISSN: 2598-5612, BPS RI/BPS-Statistic Indonesia.

Bovik, A.C. (2000) *Handbook of Image Video Processing*, Academic Press Publisher, Sand Diego.

Bukey, C.M., Chavan, R.A., Kulkarni, S.V. (2017), "Multi-Object Tracking Using Kalman Filter and Particle Filter", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017).

Cabido, R., Montemayor, A.S., dan Pantrigo, J.J. (2012), "High performance memetic algorithm particle filter for multiple object tracking on modern GPUs", Soft Comput. 2012, 16, hal. 217-230.

Gonzalez, R.C. dan Woods, R.E (2008), *Digital Image Processing*, 3rd edition, Prentice Hall, New Jersey.

Hamdi, A dan Ghanem, B. (2017), "Learning Rotation for Kernel Correlation Filter", https://www.researchgate. net/publication/319122054.

Henriques, F.J., Caseiro, R., Martins, P., dan Batista, J. (2014), "High-Speed Tracking with Kernelized Correlation Filter", IEEE Transaction On Pattern Analysis and Machine Intelligence, hal. 1-14.

Huang, S. dan Hong, J. (2011), "Moving Object Tracking Based on Camshift Approach and Kalman Filter", IEEE International Conference on Consumer Electronics, Communications and Network (CECNet), hal. 1423-1426.

Kleeman, Lindsay. (2007), *Understanding and Applying Kalman Filtering*, Clayton: Monash University.

Lee, B.Y, Liew, L.H, Cheah, W.S and Wang, Y. C (2014), "Occlusion Handling in Videos Object Tracking : A Survey", 8th International Symposium of the Digital Earth (ISDE8).

Mueller, M., Smith, N., dan Ghanem, B. (2017), "Context-Aware Correlation Filter Tracking", *Proceedings of the 30th IEEE Conference on Computer*

Vision and Pattern Recognition, Honolulu, HI, USA. 21-26 July 2017; New York, NY, USA: IEEE; 2017, hal. 1387-1395.

Patel, H.A. dan Thakore, D.G., (2013), "Moving Object Tracking Using Kalman Filter", Internatioanal Journal of Computer Science and Mobile Computing, Vol. 2, No. 4, hal.326-332.

Rizky, M.S. (2017), *Aplikasi Metode Particle Filter pada Pelacakan serta Penghitungan Orang*. Tugas Akhir., Jurusan Matematika, Institut Tekhnologi Sepuluh November, Surabaya.

Sarifuddin, M. (2016), *Pengolahan Citra dan Video Digital*, Erlangga, Jakarta.

Solichin, A., dan Harjoko, A. (2013), "Metode Background Subtraction untuk Deteksi Objek Pejalan Kaki pada Lingkungan Statis", Seminar Nasional Teknologi Informasi (SNATI).

Stauffer, C., dan Grimson, W.E.L. (1999), *Adaptive Background Mixture Model for Real Time Tracking*, Computer Vision and Pattern Recognition, Vol.2, hal. 252-258.

Suzuki, S., dkk. (1985), "Topological structural analysis of digitized binary images by border following", Computer Vision, Graphics, and Image Processing, 30(1):32-46.

Swalaganata, Galandaru. (2017), *Penerapan Citra Super-Resolusi Pada Pelacakan Objek Bergerak Menggunakan Metode Camshift dan Kalman Filter*, Tesis Magister Institut Tekhnologi Sepuluh November, Surabaya.

Wang, Ying, (2018), "A Novel Vehicle Tracking Algorithm Using Video Image Processing", 2018 International Conference on Virtual Reality and Intelligent Systems.

Wicaksono, D.W., (2017), *Pengembangan Sistem Estimasi Kecepatan Pada Kendaraan Bergerak Berbasis Pengolahan Citra Digital*,Tesis Magister Institut Tekhnologi Sepuluh November, Surabaya.

Yilmaz, A., Javed, O., dan Shah, M., (2006), *Object Tracking: A Survey*, ACM Comput. Surv. 38, (4), Article 13.

Yue, F. dan Li, X. (2017), "Improved Kernelized Correlation Filter Algorithm and Application in The Optoelectronic Tracking System", International Journal of Advanced Robotic Systems, hal. 1-10.

Zheng, B., Xu, X., Dai, Y., dan Lu, Y. (2012), "Object Tracking Algorithm Based on Combination of Dynamic Template Matching and Kalman Filter", 4th International Conference on Intelligent Human-Machine Systems and Cybernetics. Nanchang, Jiangxi, hal 136-139.

Zhou, B. dan Wang, T. (2019), "Adaptive Context-Aware and Structural Correlation Filter for Visual Tracking", Applied Sciences. 2019, www.mpdi.com/journal/applsci, hal. 1-17.

BIODATA PENULIS

Penulis bernama ***Luluk Sarifah***, lahir di Pamekasan, 14 Mei 1995, merupakan anak ke-empat dari lima bersaudara. Penulis menempuh pendidikan formal di SDN ***Mapper I***, Pamekasan, SMP ***Al-Azhar***, Pamekasan dan MA **Miftahul Ulum Bettet**, Pamekasan. Setelah lulus dari SMA penulis melanjutkan studi di **Jurusan Matematika** Universitas Islam Madura (UIM) dengan Tugas Akhir bidang **Jaringan Syaraf Tiruan** dan diterima sebagai mahasiswa angkatan 2013 lulus tahun 2017. Kemudian penulis melanjutkan S2 **Jurusan Matematika** di Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2018 dengan Tesis pada bidang **"Pengolahan Citra Digital"** dan lulus pada tahun 2020. Penulis dapat dihubungi melalui *e-mail* : syariefa.lk@gmail.com.