



TESIS-SM 142501

**PENERAPAN CITRA SUPER-RESOLUSI PADA
PELACAKAN OBJEK BERGERAK MENGGUNAKAN
METODE CAMSHIFT DAN KALMAN FILTER**

Galandaru Swalaganata
1214 201 201

DOSEN PEMBIMBING
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT
Dr. Budi Setiyono, S.Si, MT

PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



THESIS-SM 142501

**SUPER-RESOLUTION IMAGING APPLIED TO
MOVING OBJECT TRACKING USING CAMSHIFT
METHOD AND KALMAN FILTER**

Galandaru Swalaganata
1214 201 201

SUPERVISORS
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT
Dr. Budi Setiyono, S.Si, MT

MASTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017

PENERAPAN CITRA SUPER-RESOLUSI PADA PELACAKAN OBJEK BERGERAK MENGGUNAKAN METODE CAMSHIFT DAN KALMAN FILTER

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sains (M.Si.)

di

Institut Teknologi Sepuluh Nopember

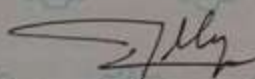
oleh:

GALANDARU SWALAGANATA
NRP. 1214 201 201

Tanggal Ujian : 06 Januari 2017

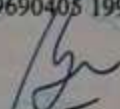
Periode Wisuda : Maret 2017

Disetujui oleh:



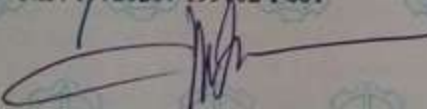
Dr. Dwi Ratna Sulistyanningrum, S.Si., M.T.
NIP. 19690405 199403 2 003

(Pembimbing I)



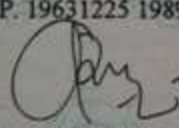
Dr. Budi Setiyono, S.Si, M.T.
NIP. 19720207 199702 1 001

(Pembimbing II)



Prof. Dr. Mohammad Isa Irawan, M.T.
NIP. 19631225 198903 1 001

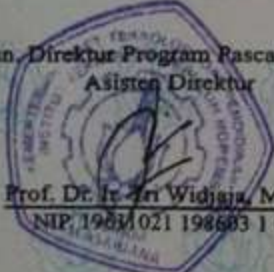

(Penguji)



Dr. Imam Mukhlash, S.Si., M.T.
NIP. 19700831 199403 1 003

(Penguji)

an, **Direktur Program Pascasarjana**
Asisten Direktur



Prof. Dr. Ir. Iri Widjaja, M.Eng.
NIP. 19631021 198603 1 001

Direktur Program Pascasarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D
NIP. 19601202 198701 1 001

PENERAPAN CITRA SUPER-RESOLUSI PADA PELACAKAN OBJEK BERGERAK MENGGUNAKAN METODE CAMSHIFT DAN KALMAN FILTER

Nama mahasiswa : Galandaru Swalaganata
NRP : 1214 201 201
Pembimbing : 1. Dr. Dwi Ratna S., S.Si, MT
2. Dr. Budi Setiyono, S.Si, MT

ABSTRAK

Pelacakan objek bergerak dalam suatu video merupakan suatu cara yang digunakan untuk mendeteksi dan menganalisa perubahan-perubahan yang terjadi pada suatu objek yang sedang diamati. Kualitas visual dan tingkat presisi pelacakan objek sangat diharapkan pada era pelacakan modern. Pada kenyataannya objek yang di-*tracking* tidak selalu terlihat jelas, sehingga hasil pelacakan yang diperoleh kurang presisi. Salah satu penyebabnya adalah objek yang berukuran kecil. Super-resolusi citra merupakan salah satu teknik yang dapat dilakukan sebelum proses pelacakan untuk memperoleh hasil pelacakan yang presisi. Super-resolusi citra merupakan teknik mendapatkan citra resolusi tinggi dari citra resolusi rendah. Super-resolusi *single frame* merupakan salah satu jenis super-resolusi yang memiliki keunggulan waktu komputasi yang cepat. Pada penelitian ini, metode super-resolusi menggunakan interpolasi Directional Bicubic yang memiliki kelebihan dalam waktu komputasi cepat dan mempertahankan ketajaman tepi (*edge*). Proses pelacakan objek menggunakan metode Camshift. Metode Camshift memiliki kelebihan dalam pelacakan berbagai kondisi histogram dan warna objek yang bervariasi. Metode Kalman Filter digunakan untuk memprediksi pergerakan objek saat *frame i + 1* berdasarkan *frame ke-i*. Dari hasil uji coba pelacakan keseluruhan video menghasilkan rata-rata pelacakan presisi sebesar 71% tanpa proses super-resolusi dan 92% dengan super-resolusi dengan metode Camshift. Sedangkan pelacakan dengan metode kombinasi menghasilkan rata-rata pelacakan presisi 73% tanpa proses super-resolusi dan 93% dengan super-resolusi. Sehingga dapat disimpulkan penambahan proses super-resolusi dan prediksi Kalman Filter membuat hasil pelacakan objek menjadi semakin presisi. Penambahan metode Kalman Filter membuat rata-rata waktu pelacakan objek lebih cepat daripada waktu pelacakan dengan metode Camshift saja pada keseluruhan video.

Kata kunci: deteksi objek, Kalman Filter, metode Camshift, pelacakan objek, super-resolusi

SUPER-RESOLUTION IMAGING APPLIED TO MOVING OBJECT TRACKING USING CAMSHIFT METHOD AND KALMAN FILTER

By : Galandaru Swalaganata
Student Identity Number : 1214 201 201
Supervisor : 1. Dr. Dwi Ratna S., S.Si, MT
2. Dr. Budi Setiyono, S.Si, MT

ABSTRACT

Moving object tracking in a video is a method used to detect and analyze changes that occur in an object that being observed. Visual quality and the precision of the tracked target are highly wished in modern tracking system. The fact that the tracked object does not always seem clear causes the tracking result less precise. The main reason was small object. Super-resolution image is one of technique which can be used before tracking approach. Super-resolution image is a technique to obtain high-resolution images from low-resolution images. Single frame super-resolution was a kind of super-resolution that it has the advantage of fast computation time. In this research Directional Bicubic interpolation was used for super-resolution technique. It has advantages of less computation time and keep edge sharpness. Camshift method is for tracking approach. It has advantages of simple calculation based on HSV color that use its histogram for some condition and color of the object varies. The Kalman Filter is used for predicted the object in frame $i + 1$ based on frame i . From the test results tracking the entire video generated an average precision tracking by Camshift method 71% without super-resolution process and 92% with the super-resolution. While the average of tracking result by combine method between Camshift and Kalman Filter tracked 73% precisely without super-resolution process and 93% with super-resolution process. Thus, adding super-resolution and prediction step with Kalman Filter made tracking result more precision. The Kalman Filter added before tracking approach made average tracking time faster than tracking time without the Kalman Filter.

Key words: Camshift method, Kalman Filter, object detection, object tracking, super-resolution

KATA PENGANTAR

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

Penerapan Citra Super-Resolusi pada Pelacakan Objek Bergerak Menggunakan Metode Camshift dan Kalman Filter

Tesis ini disusun sebagai salah satu syarat kelulusan Program Studi Strata 2 (S-2) Program Magister Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Terselesainya Tesis ini tidak terlepas dari bantuan dan dukungan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua, Bapak Agus Sunaryo dan Ibu Insiyatun, serta adik tercinta Gandazeta Wikandanan yang telah memberikan do'a dan dukungan yang tak pernah henti sampai terselesainya Tesis ini.
2. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T. dan Bapak Dr. Budi Setiyono, S.Si, M.T., selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan, perhatian, arahan, nasehat, dan motivasi kepada penulis, sehingga penulis dapat menyelesaikan Tesis ini.
3. Bapak Dr. Imam Mukhlash, S.Si., M.T., selaku ketua Jurusan Matematika ITS dan dosen penguji, atas kritik, saran, dan bantuan yang membantu penulis untuk memperbaiki Tesis ini.
4. Bapak Prof. Dr. Mohammad Isa Irawan, M.T., selaku dosen penguji, atas kritik, saran, dan bantuan yang membantu penulis untuk memperbaiki Tesis ini.
5. Yessi Affriyeni beserta keluarga di Malang atas segala bentuk dukungan, do'a, nasihat, dan motivasi selama penulis menyelesaikan Tesis ini.
6. Ibu Endah Rokhmati M.P., Ph.D., selaku dosen wali yang telah membimbing dan memotivasi selama menempuh pendidikan magister.
7. Bapak Prof. Dr. Basuki Widodo, M.Sc. selaku Dekan MIPA Institut Teknologi Sepuluh Nopember Surabaya.

8. Bapak Dr. Mahmud Yunus, M.Si., selaku Koordinator Program Studi Pascasarjana Matematika Institut Teknologi Sepuluh Nopember Surabaya yang telah memberi bimbingan selama menempuh pendidikan magister.
9. 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.
10. Bayu, Reza, Yabunayya Habibi, Gede, Tita, Yunita, Vita, Ayu, Mas Rahmat, Mbak Ija, Mas Alfian, dan Mbak Rita atas segala bentuk dukungan, saran, dan kritik yang diberikan kepada penulis selama mengerjakan Tesis hingga terselesaikannya Tesis ini.
11. Teman-teman S2 Matematika ITS khususnya angkatan 2014 ~ 2015 atas persahabatan dan kenangan selama penulis menempuh pendidikan di Pascasarjana Matematika ITS dan seluruh pihak yang tak bisa penulis sebutkan satu per satu, atas segala bantuan baik langsung maupun tidak langsung.

Penulis menyadari bahwa tulisan ini jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan 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, Januari 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
TITLE PAGE	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Penelitian yang Pernah Dilakukan	6
2.2 Super-resolusi <i>single frame</i>	8
2.2.1 Interpolasi Directional Bicubic	7
2.3 Pelacakan Objek.....	9
2.4 Metode Camshift.....	11
2.5 Metode Kalman Filter	13
BAB 3 METODA PENELITIAN.....	15
3.1 Objek dan Peralatan Penelitian	15
3.2 Tahapan Penelitian	15
3.3 Block Diagram Pelacakan Objek berbasis Citra SR	17

BAB 4 SUPER-RESOLUSI *SINGLE FRAME* DAN

PELACAKAN OBJEK	21
4.1 Perancangan Proses	21
4.1.1 Super-resolusi	22
4.1.1 Pelacakan Objek	26
4.2 Perancangan Perangkat Lunak.....	32
4.2.1 Lingkungan Perancangan.....	32
4.2.2 Perancangan Proses Akuisisi Data.....	32
4.2.2.2 Data Masukan	33
4.2.2.3 Data Proses	33
4.2.2.2 Data Keluaran	34
4.3 Implementasi Perangkat Lunak pada MATLAB.....	34
4.4 Implementasi Proses Super-resolusi Citra dan Pelacakan Objek.....	38
4.4.1 Proses Pengambilan <i>Input</i> Video dan Urutan <i>Frame</i>	38
4.4.2 Proses Konversi Video ke Urutan <i>Frame</i>	39
4.4.3 Proses Pemilihan ROI Objek	41
4.4.4 Proses Pemotongan dan Super-resolusi <i>Single Frame</i>	42
4.4.5 Proses Pelacakan Objek dengan Metode Camshift dan Kalman Filter	45
BAB 5 UJI COBA DAN ANALISA HASIL.....	49
5.1 Data Uji Coba	49
5.2 Uji Coba Pelacakan tanpa Proses Super-resolusi	52
5.3 Uji Coba Pelacakan dengan Proses Super-resolusi.....	55
5.4 Pembahasan Hasil Pengujian	58
5.5 Uji Coba Pelacakan Berdurasi Panjang	70
BAB 6 KESIMPULAN DAN SARAN	73
6.1 Kesimpulan	73
6.2 Saran	74

DAFTAR PUSTAKA	77
BIODATA PENULIS.....	81
LAMPIRAN.....	83
1 Kode program super-resolusi interpolasi Directional Bicubic	83
2 Kode program pelacakan objek dengan metode Camshift.....	87
3 Kode program pelacakan objek dengan metode kombinasi.....	91
4 Centroid hasil pelacakan video soccer.avi	95
5 Centroid hasil pelacakan video soccer.avi dengan SR.....	97

DAFTAR GAMBAR

Gambar 2.1 Super-resolusi dengan perbesaran $2\times$ pada citra asli berukuran 4×4 piksel	7
Gambar 2.2 Taksonomi metode pelacakan [14]	10
Gambar 2.3 Metode Camshift [6]	12
Gambar 2.4 Siklus diskrit Kalman Filter [12].....	13
Gambar 3.1 Diagram tahapan pelacakan objek berbasis citra super-resolusi.	18
Gambar 4.1 Diagram alir perancangan proses	22
Gambar 4.2 (a) <i>Frame</i> sebelum dilakukan pemotongan.....	23
(b) Hasil pemotongan <i>frame</i>	23
Gambar 4.3 Diagram alir super-resolusi <i>single frame</i>	23
Gambar 4.4 Diagram alir proses pelacakan dengan metode kombinasi	27
Gambar 4.5 Pemilihan objek yang akan dilacak	29
Gambar 4.6 (a) Citra hasil konversi RGB ke HSV	29
(b) Citra HSV dengan nilai Hue (1)	29
Gambar 4.7 Proses akuisisi video	33
Gambar 4.8 Rancangan halaman awal perangkat lunak	35
Gambar 4.9 Halaman awal perangkat lunak	35
Gambar 4.10 Perancangan halaman utama perangkat lunak	37
Gambar 4.11 Halaman utama perangkat lunak	37
Gambar 4.12 Halaman antar muka <i>input</i> video	39
Gambar 4.13 Halaman antar muka konversi video ke urutan <i>frame</i>	41
Gambar 4.14 Halaman antar muka pemilihan ROI objek.....	42
Gambar 4.15 Halaman antar muka pemotongan dan super-resolusi.....	44
Gambar 4.16 Halaman pelacakan objek dengan metode Camshift dan Kalman Filter	48
Gambar 5.1 Citra objek yang dilacak.....	53

Gambar 5.2 (a) Citra hasil pelacakan objek <i>frame</i> 33 pada video paralayang.avi.....	54
(b) Citra hasil pelacakan objek <i>frame</i> 18 pada video bicycle.avi.	54
(c) Citra hasil pelacakan objek <i>frame</i> 75 pada video soccer.avi ..	54
Gambar 5.3 (a) Citra hasil pelacakan objek <i>frame</i> 57 pada video paralayang.avi.....	55
(b) Citra hasil pelacakan objek <i>frame</i> 45 pada video ball.avi	55
(c) Citra hasil pelacakan objek <i>frame</i> 108 pada video surveillance.avi	55
Gambar 5.4 Citra objek super-resolusi yang dilacak.....	56
Gambar 5.5 (a) Citra hasil pelacakan objek <i>frame</i> 41 pada video bicycle.avi	57
(b) Citra hasil pelacakan objek <i>frame</i> 51 pada video ball.avi	57
(c) Citra hasil pelacakan objek <i>frame</i> 73 pada video soccer.avi .	57
Gambar 5.6 (a) Citra hasil pelacakan objek <i>frame</i> 63 pada video paralayang.avi.....	58
(b) Citra hasil pelacakan objek <i>frame</i> 110 pada video surveillance.avi	58
(c) Citra hasil pelacakan objek <i>frame</i> 148 pada video bicycle.avi.....	58
Gambar 5.7 Potongan <i>frame</i> hasil pelacakan bicycle.avi (a) tanpa SR dan (b) dengan SR	58
Gambar 5.8 Potongan <i>frame</i> hasil pelacakan paralayang.avi (a) tanpa SR dan (b) dengan SR	59
Gambar 5.9 Tingkat presisi pelacakan dengan metode Camshift	62
Gambar 5.10 Tingkat presisi pelacakan dengan metode kombinasi	62
Gambar 5.11 Tingkat presisi pelacakan pada kasus khusus.....	63
Gambar 5.12 Persentase rata-rata hasil pelacakan objek tanpa proses SR.....	64
Gambar 5.13 Persentase rata-rata hasil pelacakan objek dengan proses SR...	65

Gambar 5.14	Persentase rata-rata hasil pelacakan objek dengan metode Camshift	66
Gambar 5.15	Persentase rata-rata hasil pelacakan objek dengan metode Kombinasi	66
Gambar 5.16	Rata-rata waktu pelacakan objek per <i>frame</i> dengan proses super-resolusi (dalam detik)	67
Gambar 5.17	Faktor kecepatan pergerakan objek dari (a) <i>frame</i> 52 ke (b) <i>frame</i> 53.....	68
Gambar 5.18	Faktor pergerakan kamera dari (a) <i>frame</i> 116 ke (b) <i>frame</i> 117	68
Gambar 5.19	Faktor pergerakan kamera dari (a) <i>frame</i> 58 ke (b) <i>frame</i> 59..	69
Gambar 5.20	Faktor kondisi lingkungan dari (a) <i>frame</i> 76 ke (b) <i>frame</i> 77...	69
Gambar 5.21	Potongan <i>frame</i> hasil pelacakan video berdurasi panjang (a) orang.avi dan (b) lari.avi.	71

DAFTAR TABEL

Tabel 4.1 Lingkungan Perancangan Perangkat.....	32
Tabel 4.2 Tabel Data Proses.....	34
Tabel 5.1 <i>Dataset</i> Video Uji Coba Pelacakan.....	50
Tabel 5.2 Definisi Tingkat Ketepatan Pelacakan Objek	51
Tabel 5.3 Cara Menghitung Tingkat Ketepatan Pelacakan Objek	52
Tabel 5.4 Hasil Pelacakan Objek dengan Metode Camshift.....	53
Tabel 5.5 Hasil Pelacakan Objek dengan Metode Kombinasi.....	54
Tabel 5.6 Hasil Pelacakan Video Super-Resolusi dengan Metode Camshift ..	56
Tabel 5.7 Hasil Pelacakan Video Super-Resolusi dengan Metode Kombinasi	57
Tabel 5.8 Hasil Pelacakan Objek Video paralayang.avi	60
Tabel 5.9 Hasil Pelacakan Objek Video soccer.avi	60
Tabel 5.10 Hasil Pelacakan Objek Video ball.avi.....	60
Tabel 5.11 Hasil Pelacakan Objek Video surveillance.avi	61
Tabel 5.12 Hasil Pelacakan Objek Video jogging.avi	61
Tabel 5.13 Hasil Pelacakan Objek Video bicycle.avi.....	63
Tabel 5.14 Proses <i>Looping</i> Menentukan <i>Centroid</i> saat Proses Pelacakan Objek	67
Tabel 5.15 <i>Dataset</i> Video Uji Coba Berdurasi Panjang	70
Tabel 5.16 Hasil Pelacakan Video Berdurasi Panjang dengan Metode Kombinasi	71

BAB I

PENDAHULUAN

1.1 Latar Belakang

Moving object tracking atau pelacakan objek bergerak merupakan suatu cara yang digunakan untuk mendeteksi dan menganalisa perubahan-perubahan yang terjadi pada suatu objek dari suatu *frame* ke *frame* berikutnya di dalam video [1]. Tujuan dari pelacakan objek bergerak adalah memperkirakan lintasan objek pada bidang gambar yang berpindah pada suatu *scene* [14]. Pelacakan objek bergerak dapat membantu untuk mendeteksi, mengidentifikasi, dan menganalisa objek tersebut.

Terdapat tiga langkah mendasar dalam pelacakan objek bergerak. Langkah pertama adalah pemilihan objek bergerak pada suatu video. Langkah kedua adalah pelacakan objek yang telah dipilih dari langkah pertama dari beberapa bagian *frame*. Langkah ketiga adalah analisa pergerakan objek untuk mengenali perilaku objek tersebut [14].

Dalam bidang militer pelacakan video *automatic surveillance* untuk melakukan pemantauan terhadap suatu lokasi tertentu. Misalnya penjagaan daerah perbatasan negara dan untuk memantau pergerakan musuh di dalam medan tempur. *CCTV* juga memanfaatkan hal yang sama pada keramaian orang, misalnya *mall*, pasar, dan lain sebagainya. Pada video olahraga sepak bola, pelacakan objek digunakan untuk mengetahui posisi pemain yang sedang bermain di lapangan. Sehingga seorang analis olahraga dapat memberikan informasi mengenai pola permainan, formasi, dan strategi yang seharusnya dilakukan. Secara keseluruhan, kegiatan tersebut melibatkan pelacakan objek bergerak. Objek yang diamati berukuran relatif kecil (*small object*).

Metode atau pendekatan yang digunakan untuk pelacakan objek bergerak diantaranya adalah SIFT, *Mean-shift*, *Camshift*, Kalman Filter, dan *Particle Filter*. Dari metode yang telah disebutkan memiliki kelebihan dan kekurangan masing-masing, misalnya Kalman Filter hanya dapat digunakan pada sistem yang linear atau stabil dan Gaussian *noise* [4].

Metode Camshift merupakan pengembangan dari Mean-shift. Camshift dapat digunakan untuk tujuan umum pelacakan objek bergerak. [7]. Akan tetapi metode ini tidak dapat mendeteksi objek yang terhalang oleh suatu benda lain dalam citra tersebut. Kalman Filter dapat memprediksi pergerakan objek yang terhalang objek lain. Sehingga dengan mengkombinasikan Camshift dan Kalman Filter, pelacakan objek bergerak dapat dilakukan dalam berbagai macam kondisi [6].

Pada kenyataannya objek yang di-*tracking* tidak selalu terlihat jelas, sehingga hasil pelacakan yang diperoleh kurang presisi. Dengan memanfaatkan citra super-resolusi sebelum proses pelacakan, diharapkan akan mendapatkan hasil pelacakan objek yang bergerak di dalam video lebih presisi.

Super-resolusi merupakan teknik mengkonstruksi citra beresolusi tinggi dari citra yang beresolusi rendah [10]. Terdapat dua jenis super-resolusi berdasarkan *input frame*, yaitu super-resolusi *single frame* dan super-resolusi *multi frame*. Super-resolusi *single frame* memiliki kelebihan dari sisi waktu komputasi yang lebih cepat daripada super-resolusi *multi frame* [8]. Metode yang cukup populer untuk melakukan super-resolusi *single frame* adalah *Nearest neighbor*, *Bilinear*, dan *Bicubic* [13].

Penelitian mengenai pelacakan objek bergerak dengan Kalman Filter telah dilakukan dengan judul *Moving Object Tracking Using Kalman Filter* [12]. Penelitian tersebut dilakukan pada objek dengan latar belakang bervariasi dengan target objek tunggal. Penelitian lainnya yang berjudul *Moving Target Tracking Based on Camshift Approach and Kalman Filter* [6] berhasil mengkombinasikan Camshift dan Kalman Filter. Dari kedua penelitian tersebut tidak menggunakan *small object*. Penelitian yang dilakukan oleh Olegs Mise dan Toby P. Breckonb yang berjudul *Super-Resolution Imaging Applied to Moving Targets in High Dynamic Scenes* [10] berhasil menerapkan citra super-resolusi untuk mendapatkan hasil pelacakan objek bergerak. Super-resolusi menggunakan super-resolusi *multi frame*. Sedangkan pelacakan objek menggunakan metode *block matching correlation*.

Terdapat kesulitan untuk melacak objek yang relatif kecil pada video. Kesulitan pertama adalah objek yang relatif kecil akan sulit untuk dibedakan antara objek dan

noise [3]. Kesulitan lainnya adalah *small object* direpresentasikan oleh piksel (antara 10 sampai 100 piksel), sehingga informasi yang diperoleh sedikit [9].

Berdasarkan uraian yang telah diberikan, pada penelitian ini diimplementasikan kombinasi metode Camshift dan Kalman Filter untuk pelacakan objek kecil bergerak berbasis super-resolusi *single frame*. Metode Kalman Filter digunakan sebagai alat untuk memprediksi pergerakan objek pada *frame* selanjutnya. Sehingga diharapkan hasil pelacakan dapat lebih presisi. Sebelum proses pelacakan objek, proses super-resolusi *single-frame* dilakukan terlebih dahulu dengan cara memotong bagian *frame*. Pemotongan *frame* dilakukan sebelum super-resolusi dengan tujuan untuk mempersempit area pencarian saat proses pelacakan. Selain itu juga, keseluruhan proses dan perancangan perangkat lunak diimplementasikan pada MATLAB.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diberikan, maka permasalahan yang diusulkan untuk dibahas dalam penelitian ini adalah sebagai berikut.

1. Bagaimana menerapkan super-resolusi *single-frame* pada video?
2. Bagaimana menerapkan dan mengimplementasikan metode Camshift dan Kalman Filter untuk pelacakan objek bergerak berbasis citra super-resolusi pada video di MATLAB?
3. Bagaimana analisa dan akurasi metode Camshift dan Kalman Filter untuk pelacakan objek bergerak berbasis citra super-resolusi?

1.3 Batasan Masalah

Untuk menghasilkan suatu hasil yang efektif, maka batasan masalah yang digunakan dari penelitian yang diusulkan adalah sebagai berikut.

1. Data video yang akan digunakan telah disimpan dalam penyimpanan komputer.
2. Objek yang akan diamati adalah objek tunggal yang ditentukan oleh pengguna.

3. Objek yang diamati atau dilacak maksimal berukuran $10\sim 100 \times 10\sim 100$ piksel.
4. Posisi kamera saat merekam video adalah statis atau tidak bergerak.
5. Pada pengamatan yang dilakukan tidak terdapat kasus oklusi atau objek terhalang dengan objek lainnya.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang diberikan, tujuan dari penelitian adalah menerapkan metode Camshift dan Kalman Filter untuk pelacakan objek berbasis citra super-resolusi *single-frame* pada video dan mengimplementasikannya di MATLAB. Selanjutnya analisa akurasi metode Camshift dan Kalman Filter untuk pelacakan objek bergerak berbasis citra super-resolusi.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian yang diusulkan, maka manfaat yang ingin diperoleh sebagai berikut.

1. Untuk memudahkan pengamatan dalam pelacakan suatu video yang di dalam video terdapat objek bergerak dengan memanfaatkan citra super-resolusi sehingga akan diperoleh hasil pelacakan objek yang sangat presisi.
2. Dapat membantu pihak-pihak dalam menganalisis pelacakan objek bergerak yang beresolusi tinggi. Misal untuk mengetahui posisi setiap orang yang sedang bermain di lapangan, sehingga seorang analis olahraga dapat memberikan informasi mengenai pola permainan, formasi, dan strategi yang seharusnya dilakukan.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bagian ini membahas uraian singkat mengenai beberapa teori yang digunakan untuk memudahkan dalam pembahasan, meliputi pelacakan objek, super-resolusi *single frame*, metode Camshift, dan metode Kalman Filter. Terlebih dahulu diberikan uraian mengenai penelitian yang telah dilakukan sebelumnya.

2.1 Penelitian yang Pernah Dilakukan

Berikut beberapa penelitian yang telah dilakukan sebelumnya berkaitan dengan pelacakan objek bergerak.

1. “*Moving Object Tracking Using Kalman Filter*”, International Journal of Computer Science and Mobile Computing oleh Hitesh A. Patel dan Darshak G. Thakore tahun 2013. Pembahasan mengenai pelacakan objek pada video *surveillance*. Tahapan pelacakan objek yang pertama adalah deteksi objek, sedangkan yang selanjutnya adalah pelacakan objek tersebut. Video yang digunakan adalah video static. Penelitian tersebut dilakukan pada objek dengan latar belakang bervariasi dengan target objek tunggal dan bukan objek yang berukuran relatif kecil.
2. “*Moving Target Tracking Based on Camshift Approach and Kalman Filter*”, An International Journal Applied Mathematics & Information Sciences oleh Kuo-Hsien Hsia, Shao-Fan Lien dan Juhng-Perng Su tahun 2012. Penelitian ini membahas pelacakan objek dengan kombinasi dua metode yaitu metode Camshift dan Kalman Filter. Penelitian ini berhasil mendeteksi objek pada video yang tertutup oleh objek lainnya. Pelacakan pada penelitian ini berbasis Kalman Filter. Sehingga metode Camshift terletak diantara langkah prediksi dan koreksi pada Kalman Filter.
3. “*Detection and Tracking of Very Small Low Contrast Objects*” oleh D. Daviesy, P. Palmery, dan M. Mirmehdiz tahun 1998. Penelitian ini berhasil

mendeteksi *small object* menggunakan Wavelets untuk objek tunggal dan Kalman Filter untuk *multiple* objek.

4. “*Super-Resolution Imaging Applied to Moving Targets in High Dynamic Scenes*”, International Society for Optics and Photonics oleh Olegs Mise dan Toby P. Breckonb pada tahun 2013. Penelitian ini menerapkan citra super-resolusi untuk mendapatkan hasil pelacakan objek bergerak. Pada penelitian ini super-resolusi menggunakan metode *Sum of the Absolut Differences* (SAD) dan perkiraan *gradient-descent motion*. Sedangkan pelacakan objek menggunakan metode *block matching correlation*.

Berdasarkan beberapa penelitian yang telah dilakukan tersebut, pada laporan penelitian ini dibahas mengenai pelacakan objek bergerak berbasis citra super-resolusi *single frame* dengan metode kombinasi Camshift dan Kalman Filter. Pelacakan pada penelitian ini berbasis Metode Camshift. Metode Camshift digunakan untuk melakukan pelacakan objek, sedangkan Kalman Filter untuk prediksi objek yang dilacak.

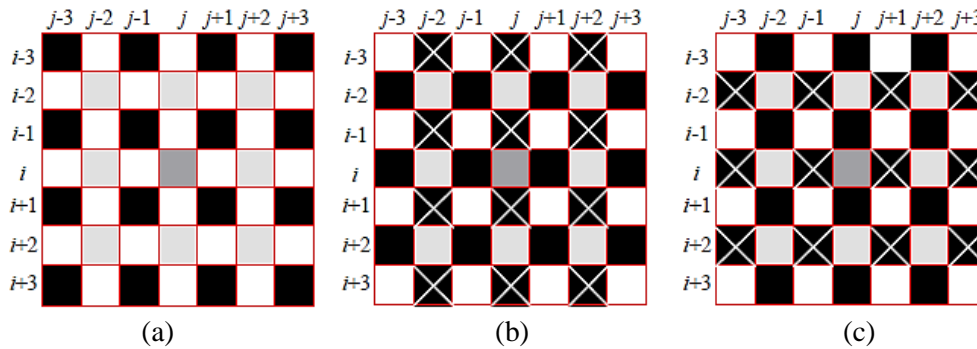
2.2 Super-Resolusi *Single Frame*

Metode super-resolusi secara umum mengkonstruksi citra beresolusi tinggi dari beberapa citra yang beresolusi rendah [10]. Terdapat dua jenis super-resolusi, yaitu super-resolusi *single frame* dan super-resolusi *multi frame*. Pada penelitian ini memakai super-resolusi *single frame*. Sehingga untuk mendapatkan citra beresolusi tinggi cukup dengan mendapatkan *single frame* dari ROI yang telah ditentukan, karena telah menyediakan informasi yang cukup untuk rekonstruksi.

Secara umum super-resolusi terdiri dari dua tahapan, yaitu registasi citra dan rekonstruksi citra. Pada proses pembentukan citra beresolusi tinggi pada *single frame* secara sederhana dengan cara membesarkan citra target sedemikian sehingga dalam proses tersebut tidak menghilangkan dan tetap mengandung informasi sebelum dilakukan super-resolusi. Metode yang cukup populer untuk melakukan super-resolusi *single frame* adalah *Nearest neighbor*, *Bilinear*, dan *Bicubic* [13].

2.2.1 Interpolasi *Directional Bicubic*

Pada penelitian ini akan menggunakan pengembangan metode interpolasi *Bicubic* yaitu *Directional Bicubic Interpolation* [8]. Metode interpolasi *Directional Bicubic* dapat melakukan interpolasi secara fleksibel ke segala arah, tidak seperti 1D *Bicubic* melakukan interpolasi pada arah horizontal dan vertikal. Gambar 2.1 menjelaskan super-resolusi dengan perbesaran $2\times$ pada citra asli berukuran 4×4 piksel. Piksel yang belum terisi diinterpolasi menjadi dua tahap. Piksel berwarna hitam pada Gambar 2.1(a) menandakan piksel yang diketahui dari citra asli. Piksel dengan tanda “ \times ” dan berwarna abu-abu adalah piksel yang akan diisi.



Gambar 2.1. Super-resolusi dengan perbesaran $2\times$ pada citra asli berukuran 4×4 piksel.

Interpolasi *Directional Bicubic* menghasilkan kompleksitas komputasi rendah dan mempertahankan ketajaman tepi (*edge*) dan detail lebih baik [8]. Berikut ini langkah-langkah interpolasi *Directional Bicubic*.

Input : citra resolusi rendah (I_{LR})

Output : citra resolusi tinggi (I_{HR})

Begin

- Inisialisasi. Memperbesar I_{LR} ke I_{HR} dengan $I_{HR}(2i - 1, 2j - 1) = I_{LR}(i, j)$, $1 \leq i \leq M, 1 \leq j \leq N$, dimana M, N adalah ukuran dari citra *low resolution*.

- Lakukan interpolasi *Bicubic* dengan persamaan berikut

$$p(i, j) = \sum_{m=0}^3 \sum_{n=0}^3 a_{mn} i^m j^n \quad (2.1)$$

dengan $p(i, j)$ adalah nilai piksel interpolasi *Bicubic* dan a_{ij} adalah nilai koefisien 16 tetangga yang dipertimbangkan [8].

- Langkah ketiga adalah melakukan interpolasi pada bagian yang berwarna abu-abu (Gambar 2.1(a)). Sebelum mengisi piksel yang kosong, dihitung terlebih dahulu gradien 45° dan 135° sebanyak k , dimana $0 < k < M$.

$$G_1^{45^\circ}(k) = \sum_{m=3, \pm 1} \sum_{n=3, \pm 1} |I_{HR}(i+m, j-n) - I_{HR}(i+m-2, j-n+2)| \quad (2.2)$$

$$G_2^{135^\circ}(k) = \sum_{m=3, \pm 1} \sum_{n=3, \pm 1} |I_{HR}(i+m, j+n) - I_{HR}(i+m-2, j+n+2)| \quad (2.3)$$

Pengisian nilai piksel yang kosong, menggunakan sebagai berikut.

$$\max(G_1) = \text{nilai maksimum } (G_1^{45^\circ}(k))$$

$$\max(G_2) = \text{nilai maksimum } (G_2^{135^\circ}(k))$$

$$\text{if } \frac{(1 + \max(G_1))}{(1 + \max(G_2))} > \text{Threshold}$$

$$\text{piksel}(i, j) = p(i, j)$$

$$\text{else if } \frac{(1 + \max(G_2))}{(1 + \max(G_1))} > \text{Threshold}$$

$$\text{piksel}(i, j) = p(i, j)$$

else

$$w_1 = \frac{1}{1 + \max(G_1)}, w_2 = \frac{1}{1 + \max(G_2)}$$

$$\text{piksel}(i, j) = \frac{(w_1 * p(i, j) + w_2 * p(i, j))}{w_1 + w_2}$$

end (2.4)

- Langkah keempat adalah melakukan interpolasi pada bagian yang bertanda silang (Gambar 2.1(b) dan (c)). Sebelum mengisi piksel yang kosong,

dihitung terlebih dahulu gradien horizontal dan vertikal sebanyak k , dimana $0 < k < M$.

$$G_1^{90^\circ}(k) = \sum_{m=\pm 1} \sum_{n=0,2} |I_{HR}(i+m, j-n) - I_{HR}(i+m, j-n+2)| \\ + \sum_{m=0,\pm 2} |I_{HR}(i+m, j-1) - I_{HR}(i+m, j+1)| \quad (2.5)$$

$$G_2^{0^\circ}(k) = \sum_{m=0,2} \sum_{n=\pm 1} |I_{HR}(i-m, j+n) - I_{HR}(i-m+2, j+n)| \\ + \sum_{n=0,2} |I_{HR}(i-1, j+n) - I_{HR}(i+1, j+n)| \quad (2.6)$$

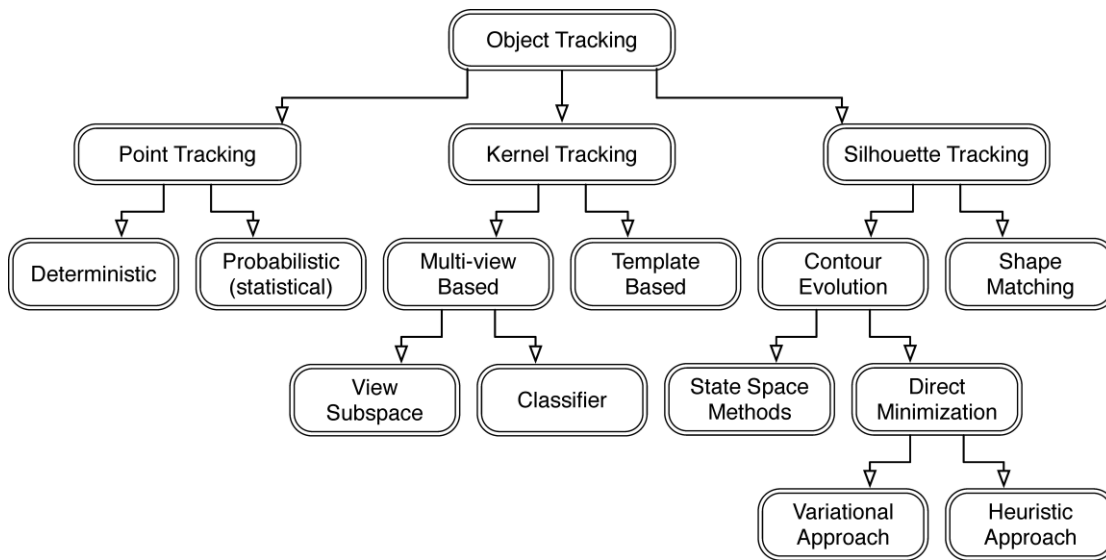
Untuk mengisi nilai piksel yang kosong, menggunakan pertimbangan yang sama seperti persamaan 2.4.

End

2.3 Pelacakan Objek

Dalam bahasa yang sederhana, pelacakan dapat didefinisikan sebagai suatu masalah dalam memperkirakan pergerakan benda pada suatu bidang gambar dimana benda bergerak di dalam suatu *scene*. Dengan kata lain, suatu jejak ditandakan secara konsisten kepada objek tersebut pada *frame* yang berbeda dari suatu video. Terdapat tiga langkah mendasar dalam pelacakan objek bergerak, yaitu pemilihan objek bergerak yang diinginkan, pelacakan objek yang telah dipilih, dan analisa pergerakan objek [14].

Objek pelacakan dapat direpresentasikan dalam bentuk titik/ *point*, *primitive geometric shapes* seperti persegi atau elipse, *silhouette*, bagian tubuh manusia atau objek lainnya, dan *skeletal models*. Reperesentasi objek, deteksi objek, segmentasi objek, dan pelacakan objek adalah kegiatan yang dilakukan dalam pelacakan. Sehingga tergantung kepada pengguna dalam mengambil suatu informasi dari pelacakan tersebut [14]. Taksonomi tentang pembagian pelacakan objek dapat dilihat pada Gambar 2.2.



Gambar 2.2. Taksonomi metode pelacakan [14]

Pelacakan *point tracking* adalah objek terdeteksi di *frame* secara berturut-turut yang diwakili oleh titik. Titik yang digunakan sebagai tanda objek saling berhubungan dari *frame* pertama sampai *frame* terakhir membentuk suatu lintasan pelacakan. Titik-titik tersebut mewakili posisi objek dan pergerakan objek. MGE tracker dan GOA tracker merupakan contoh metode deterministik. Kalman Filter, JPDAF dan PMHT merupakan contoh 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*. SVM tracker dan Eigentracking merupakan contoh metode berdasarkan *multi-view*.

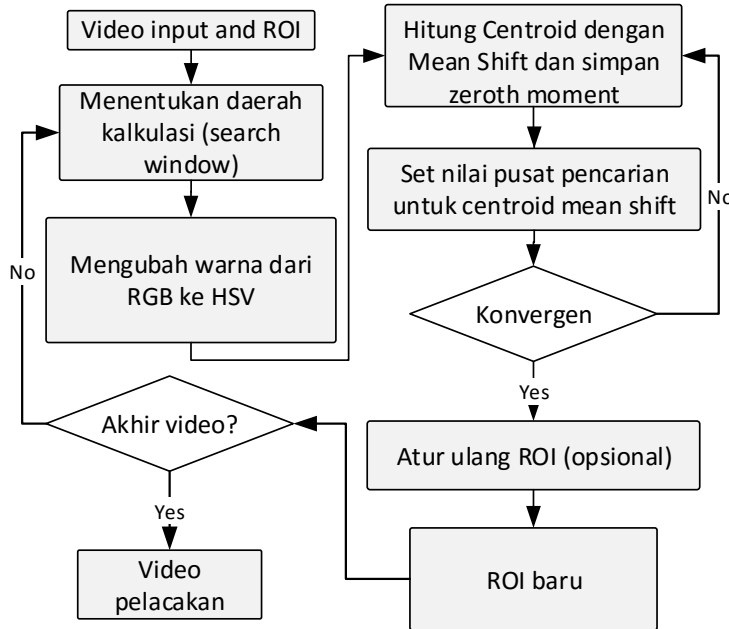
Silhouette tracking dilakukan dengan memperkirakan wilayah objek di setiap *frame*. Metode pelacakan siluet menggunakan informasi yang ditandai dalam bentuk atau tanda berupa wilayah objek. State space models dan Variational methods merupakan contoh metode *contour evolution*. Hausdorff dan Hough transform merupakan contoh metode *matching shape*.

2.4 Metode Camshift

Metode Camshift (*Continuously Adaptive Mean Shift*) merupakan pengembangan dari metode *Mean-shift* [6]. Metode Camshift dapat digunakan untuk tujuan umum pelacakan objek bergerak. Selain itu metode Camshift dapat digunakan dalam berbagai kondisi histogram dan warna objek yang bervariasi [7]. Kelebihan lainnya dari metode Camshift adalah tidak adanya ketergantungan pada jenis objek yang diamati, baik objek yang bersifat *rigid* maupun *non-rigid*. Sebelum membahas metode Camshift, terlebih dahulu diberikan langkah-langkah dari metode *Mean-shift*.

- a. Mengatur *region of interest* (ROI) dari *input* gambar.
- b. Tentukan lokasi awal *search window*.
- c. Hitung daerah *mean* dalam *search window*.
- d. Posisikan *search window* ke tengah daerah *mean* seperti dihitung pada langkah (c).
- e. Ulangi langkah (c) dan (d) hingga konvergen (atau hingga pergeseran daerah *mean* kurang dari *threshold* / batas yang ditentukan).

Pengembangan metode *Mean-shift* menjadi Camshift terletak pada langkah terakhir. Dari langkah (e) dari metode *Mean-shift*, koordinat dari *matching block* terbaik dijadikan nilai awal dari posisi pencarian pada gambar berikutnya yang terekam. Prosedur dari metode Camshift seperti Gambar 2.3.



Gambar 2.3. Metode Camshift [6]

Untuk citra berdistribusi probabilitas warna, daerah *mean (centroid)* didalam *search window* bisa dicari dengan persamaan berikut ini.

Zeroth moment bisa dicari dengan persamaan:

$$M_{00} = \sum_x \sum_y I(x, y) \quad (2.7)$$

First moment untuk x dan y bisa dicari dengan persamaan :

$$M_{10} = \sum_x \sum_y xI(x, y) \quad (2.8)$$

$$M_{01} = \sum_x \sum_y yI(x, y) \quad (2.9)$$

Maka lokasi mean dalam *search window (centroid)* adalah :

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \quad (2.10)$$

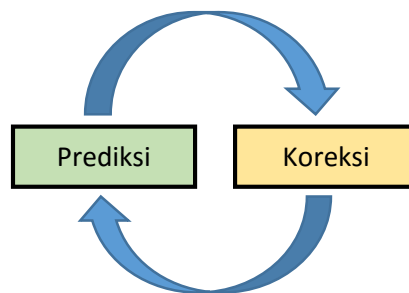
dengan $I(x, y)$ adalah nilai piksel di posisi (x, y) pada citra, sedangkan x, y berada di dalam *search window*. M_{10} adalah *first moment* untuk x , M_{01} adalah *first moment* untuk y , dan M_{00} adalah *zeroth moment*.

2.5 Metode Kalman Filter

Kalman Filter digunakan untuk memperkirakan suatu keadaan dari suatu sistem yang linear dan berdistribusi Gaussian [12]. Kalman Filter merupakan metode statistikal yang berbasis pada *point tracking* yang memiliki kelebihan selalu memberikan solusi optimal berupa hasil prediksi yang presisi [12]. Selain itu, waktu komputasi metode Kalman Filter yang cepat karena proses penghitungan berupa titik dan *output* berupa titik. Metode Kalman Filter juga dapat diaplikasikan untuk pelacakan *single object* dan *multiple object*.

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

Metode Kalman Filter memperkirakan lokasi objek yang paling mungkin pada *frame* saat ini berdasarkan hasil dari pelacakan target pada *frame* sebelumnya, selanjutnya mencari lokasi target di sekitaran lokasi sebelumnya [7]. Dua langkah utama metode Kalman Filter adalah prediksi (*prediction*) dan koreksi (*correction*) [6]. Gambar 2.4 menunjukkan siklus diskrit Kalman Filter pada *image processing*.



Gambar 2.4 Siklus diskrit Kalman Filter [12]

Didefinisikan vektor *state* $X_k = [x, y, v_x, v_y]$, 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 ini persamaan prediksi (*prediction*) dan koreksi (*correction*) dari metode Kalman Filter.

a. Tahap prediksi

$$\text{Persamaan prediksi 1 : } X_{pred_k} = A \cdot X_{k-1} \quad (2.11)$$

$$\text{Persamaan prediksi 2 : } P_{pred_k} = A \cdot P_{k-1} \cdot A^T + Q \quad (2.12)$$

b. Tahap koreksi

Persamaan Kalman-gain :

$$K_k = P_{pred_k} \cdot H^T \cdot (H \cdot P_{pred_k} \cdot H^T + R)^{-1} \quad (2.13)$$

$$\text{Persamaan update 1 : } X_k = X_{pred_k} + K_k \cdot (Z_k - H \cdot X_{pred_k}) \quad (2.14)$$

$$\text{Persamaan update 2 : } P_k = P_{pred_k} - K_k \cdot H \cdot P_{pred_k} \quad (2.15)$$

Nilai dari matriks transisi A , matriks ukuran (*measurement*) H , matriks kovarian *noise* proses Q , dan matriks kovarian *noise* ukuran (*measurement*) R sebagai berikut.

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

BAB III

METODA PENELITIAN

Pada bagian ini membahas uraian singkat mengenai tahapan penelitian yang akan dilakukan beserta *block* diagram pelacakan objek berbasis citra super-resolusi. Terlebih dahulu diberikan uraian mengenai objek penelitian dan peralatan yang akan digunakan.

3.1 Objek dan Peralatan Penelitian

Pada penelitian ini objek yang digunakan adalah benda bergerak dengan ukuran objek kecil (*small object*) yang berukuran $10\sim 100 \times 10\sim 100$ piksel dan *single object* atau objek tunggal dalam suatu rekaman video. Peralatan yang digunakan pada penelitian ini adalah MATLAB sebagai *software* utama untuk implementasi pelacakan objek beserta antar muka. Serta kamera untuk pengambilan video sebagai *input* untuk percobaan.

3.2 Tahapan Penelitian

Berikut diberikan beberapa tahapan penelitian yang akan dilakukan dalam penelitian yang diusulkan.

1. Studi literatur

Mengkaji teori mengenai pengolahan video pelacakan yang akan diimplementasikan pada video pelacakan tersebut. Kajian ini dilakukan dengan membaca jurnal *video tracking* yang berkaitan dengan metode Camshift dan Kalman Filter, jurnal *video tracking* untuk *small object*, dan jurnal super-resolusi *single frame* untuk mendapatkan gambar beresolusi tinggi dari objek.

2. Akuisi data

Pada tahap ini akan dilakukan pengumpulan data video. Video diperoleh dari akuisi di lingkungan kampus ITS, Youtube *dataset* dan VIRAT *dataset* [11].

Video yang dibuat sendiri melibatkan bola yang digelindingkan secara vertikal maupun horizontal. Sedangkan pada VIRAT *dataset* berupa video surveillance pada suatu daerah tertentu. Video lainnya didapatkan dari situs youtube.

3. Penerapan metode Camshift dan Kalman Filter berbasis citra super-resolusi
Pada tahap ini akan dilakukan penerapan metode pelacakan objek menggunakan kombinasi metode Camshift dan Kalman Filter. Pada tahapan ini diawali dengan proses super-resolusi *single frame* untuk mendapatkan citra yang lebih baik. Proses super resolusi *single frame* menggunakan metode interpolasi *Directional Bicubic*. Metode Camshift digunakan untuk melakukan pelacakan objek, sedangkan Kalman Filter untuk prediksi pergerakan objek.
4. Implementasi pada MATLAB
Implementasi pada MATLAB bertujuan untuk mempermudah melakukan percobaan dan analisa hasil karena *input* video yang bermacam-macam ukuran. Dengan demikian diharapkan proses pengerjaan pada tahap ini dapat dilakukan lebih cepat.
5. Percobaan dan analisa hasil
Percobaan dari video yang telah didapatkan dari langkah akuisisi data. Akan dilakukan pengukuran kinerja dari hasil implementasi pada MATLAB, yaitu perbandingan waktu komputasi dari *input* video yang berbeda dan tingkat ketepatan atau presisi pelacakan objek. Ukuran tingkat ketepatan pelacakan objek dihitung dari berapa kali objek tersebut dapat terdeteksi dalam serangkaian *frame* dari suatu video dibagi dengan jumlah keseluruhan *frame*. Dari beberapa video yang akan digunakan nantinya, dapat dibandingkan tingkat ketepatan pelacakan objek yang bergerak. Sedangkan untuk perbandingan waktu komputasi dihitung mulai saat proses pelacakan sampai berakhir pelacakan.

6. Publikasi

Pada tahapan ini penelitian yang telah dilakukan selanjutnya diseminasikan pada suatu seminar internasional, yaitu The Asian Mathematical Conference (AMC) pada tanggal 25-29 Juli 2016 di Bali.

7. Penulisan Tesis.

3.3 Block Diagram Pelacakan Objek berbasis Citra Super-resolusi

Block diagram dari metode penelitian yang digunakan untuk pelacakan objek berbasis citra super-resolusi ditunjukkan seperti Gambar 3.1. Dimulai dari *input*/masukan berupa video *offline* hingga video hasil pelacakan.

1. *Input* video yang telah tersimpan pada direktori

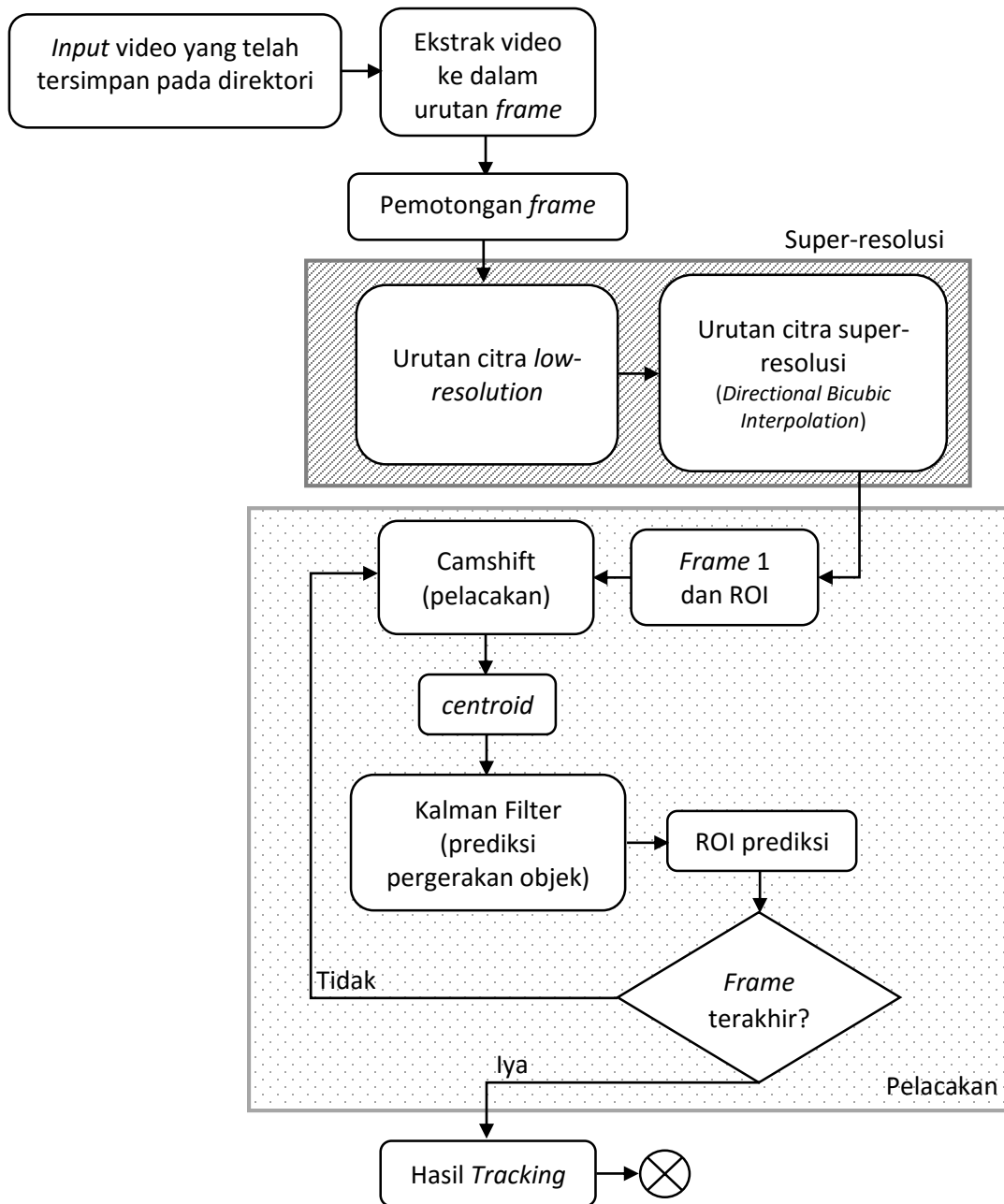
Video diperoleh dari akuisisi di lingkungan kampus ITS dan VIRAT *dataset* [11]. Video yang dibuat sendiri melibatkan bola yang digelindingkan secara vertikal maupun horizontal. Sedangkan pada VIRAT *dataset* berupa video surveillance pada suatu daerah tertentu. Video lainnya didapatkan dari situs youtube.

2. Ekstrak video ke dalam urutan *frame*

Input video yang digunakan akan dilakukan proses pemecahan video menjadi beberapa rangkaian citra yang sering disebut *frame*. Proses ini bertujuan untuk mempermudah penentuan ROI dan proses-proses berikutnya.

3. Pemotongan *frame* (*Cropping frame*)

Pada langkah ini dilakukan pemotongan *frame* yang didalamnya terdapat lintasan objek yang akan dilacak. Proses pemotongan ini dilakukan untuk meminimalkan ruang pencarian objek untuk proses pelacakan. Hasil pemotongan pada setiap *frame* akan disuper-resolusi untuk mendapatkan detail objek yang lebih baik.



Gambar 3.1. Diagram tahapan pelacakan objek berbasis citra super-resolusi

4. Super-resolusi *single frame*

Super-resolusi dilakukan pada hasil pemotongan *frame* yang telah ditentukan. Metode yang digunakan untuk super-resolusi, yaitu interpolasi *Directional Bicubic*. Proses super-resolusi dikenakan pada semua *frame*. Tujuannya adalah

untuk membuat objek semakin terlihat jelas. Proses ini dilakukan diluar proses pelacakan karena waktu komputasi super-resolusi yang lama. Pada proses ini dapat dilakukan ekualisasi histogram untuk memperoleh citra yang lebih bagus. Namun, proses ekualisasi histogram bersifat tidak harus, karena tidak semua video jika diekualisasi akan semakin bagus hasilnya.

5. Menentukan ROI (*Region of Interest*)

Region of Interest atau ROI akan 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 di dalam ROI tidak akan terpilih.

6. Pelacakan objek (Metode Camshift)

Pelacakan objek dilakukan berdasarkan ROI dari citra yang telah di super-resolusi. Selanjutnya metode Camshift diimplementasikan untuk mendeteksi perpindahan objek pada video. Target yang teridentifikasi akan ditandai dengan *bounding box* dan dihitung *centroid*-nya. *Centroid* yang didapatkan akan menjadi *input* dari Kalman Filter untuk dilakukan prediksi benda di *frame* berikutnya.

7. Prediksi pergerakan objek (Metode Kalman Filter)

Output yang didapatkan dari metode Camshift menjadi *input* dari metode Kalman Filter untuk melakukan prediksi pergerakan objek. Proses penghitungan yang dilakukan pada metode Kalman Filter menghasilkan prediksi letak objek di *frame* selanjutnya. *Output* metode Kalman Filter tersebut digunakan sebagai *input* untuk perhitungan pada metode Camshift pada *frame* selanjutnya.

8. Hasil *tracking*

Hasil *tracking* berupa video yang merupakan rangkaian dari *frame-frame* yang terdapat objek bergerak yang dibatasi oleh *bounding box*.

BAB IV

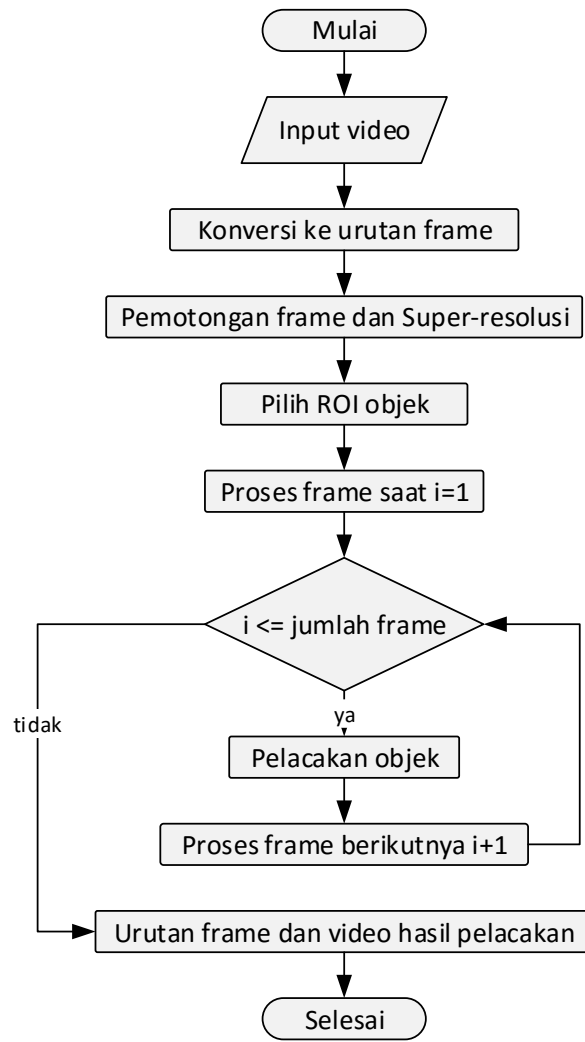
SUPER-RESOLUSI *SINGLE FRAME* DAN PELACAKAN OBJEK

Pada bab ini menjelaskan mengenai perancangan proses penerapan citra super-resolusi pada pelacakan objek bergerak. Perancangan proses diawali dengan proses super-resolusi citra dengan interpolasi *Directional Bicubic*, proses pemilihan ROI (*Region of Interest*), dan proses pelacakan objek menggunakan metode Camshift dan Kalman Filter. Tahapan selanjutnya adalah perancangan dan implementasi perangkat lunak. Tujuannya adalah untuk mempermudah proses uji coba dari perancangan proses yang telah dibuat pada bab selanjutnya.

4.1 Perancangan Proses

Perancangan proses bertujuan untuk menjaga agar proses penerapan citra super-resolusi dan pelacakan objek bergerak berjalan dengan lancar dan teratur, sehingga menghasilkan informasi yang dibutuhkan untuk setiap prosesnya. Gambar 4.1 menjelaskan diagram alir perancangan proses pelacakan yang telah dikerjakan.

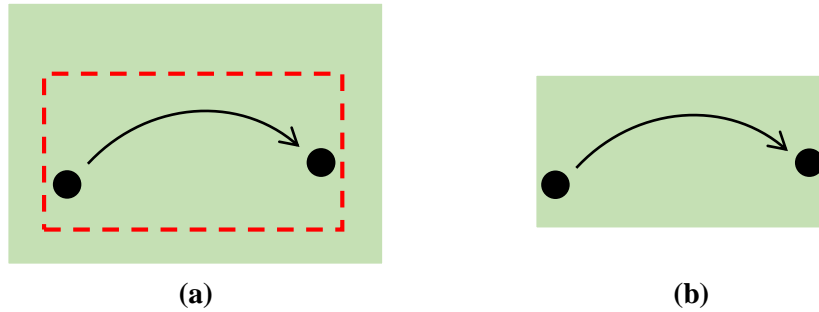
Proses diawali dengan pemilihan *input* video dari direktori komputer yang selanjutnya dikonversi ke urutan *frame*. Langkah berikutnya adalah pemotongan *frame* sebesar pergerakan benda dimana hasilnya akan disuper-resolusi *single frame*. Selanjutnya pemilihan ROI (*Region of Interest*) objek yang merupakan citra objek yang akan dilacak. Pada tahap berikutnya dilakukan proses pelacakan objek pada *frame* hasil super-resolusi menggunakan metode Camshift. Metode Camshift menggunakan 1 *channel* warna HSV untuk menentukan *centroid* dari ROI. Prediksi pergerakan benda di *frame* selanjutnya menggunakan metode Kalman Filter. Proses pelacakan objek dilakukan sampai *frame* terakhir dari video.



Gambar 4.1 Diagram alir perancangan proses.

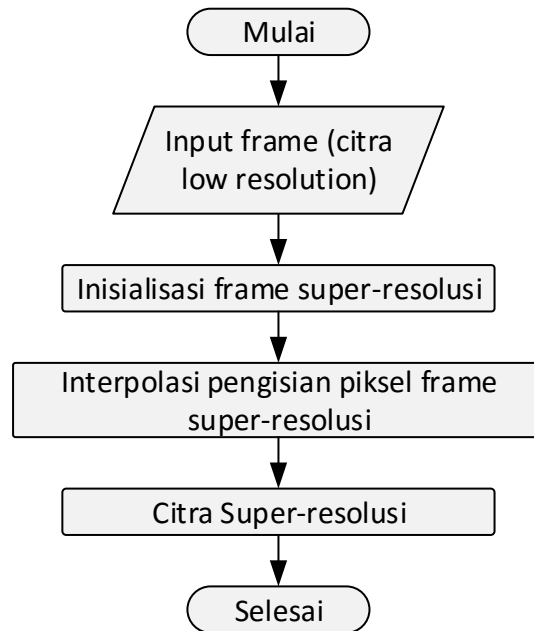
4.1.1 Super-resolusi *Single Frame*

Proses super-resolusi dimuali dengan proses pemotongan *frame*. Proses pemotongan *frame* pada dasarnya sama halnya memelih ROI, namun ukuran dari pemotongan *frame* lebih besar dari pemilihan ROI. Tujuan dari pemotongan *frame* adalah untuk memperkecil *search window* dari objek yang akan dilacak, sehingga diharapkan akan mampu mengurangi waktu komputasi. Proses pemotongan *frame* yang dilakukan dalam penelitian ini disajikan pada Gambar 4.2.



Gambar 4.2 (a) *Frame* sebelum dilakukan pemotongan, (b) Hasil pemotongan *frame*

Proses pemilihan pemotongan *frame* akan sekaligus diikuti dengan proses super-resolusi. Langkah ini dilakukan dengan tujuan juga mengurangi waktu komputasi, sehingga setelah koordinat ditentukan, proses super-resolusi *single frame* akan dilakukan. Proses super-resolusi *single frame* yang dilakukan dalam penelitian ini disajikan pada Gambar 4.3.

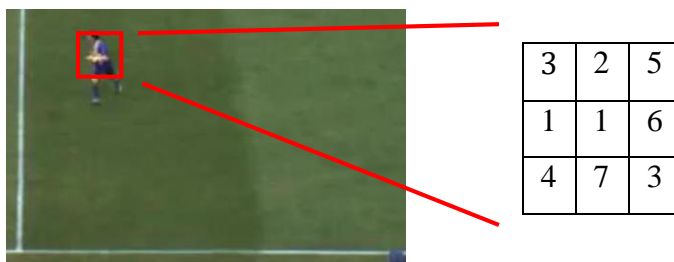


Gambar 4.3 Diagram alir super-resolusi *single frame*.

Ekualisasi histogram dapat dilakukan pada *frame* yang telah disuper-resolusi. Proses ekualisasi histogram bertujuan untuk membuat distribusi histogram pada *frame* menjadi distribusi normal.

Tahapan super-resolusi *single frame* dimulai dari *input* berupa hasil pemotongan *frame*. Super-resolusi dilakukan untuk beberapa *frame* atau untuk keseluruhan *frame*. Inisialisasi berupa pemetaan dari *input* berupa citra beresolusi rendah ke citra beresolusi tinggi. Pada super-resolusi *single frame*, piksel-piksel yang kosong atau belum ada nilainya akan diisi dengan cara interpolasi seperti yang telah dijelaskan pada Bab 2. Proses ini akan menghasilkan urutan *frame* baru dengan ukuran $2 \times$ yang lebih besar dari *frame* hasil pemotongan. Hasil dari urutan *frame* baru yang akan digunakan pada langkah selanjutnya yaitu pelacakan objek.

Sebagai ilustrasi proses super-resolusi, misalkan diberikan citra *A* berukuran 3×3 dari citra asal *B*. Citra *A* akan disuper-resolusi $2 \times$ menggunakan interpolasi *Directional Bicubic*.



- Langkah pertama adalah memperbesar I_{LR} ke I_{HR} dengan pemetaan

$$I_{HR}(2i - 1, 2j - 1) = I_{LR}(i, j), 1 \leq i \leq M, 1 \leq j \leq N$$

dimana M, N adalah ukuran dari citra *low resolution*. Sehingga diperoleh citra super-resolusi yang berukuran $I_{HR}(5 \times 5)$ dengan isi sebagai berikut.

	$j-2$	$j-1$	j	$j+1$	$j+2$
$i-2$	3		2		5
$i-1$					
i	1		1		6
$i+1$					
$i+2$	4		7		3

- Langkah kedua adalah melakukan interpolasi *Bicubic* sesuai persamaan (2.1) terlebih dahulu. Sehingga diperoleh hasil sebagai berikut

$$P = \begin{array}{|c|c|c|c|c|c|} \hline 3 & 3 & 2 & 3 & 4 & 5 \\ \hline 3 & 2 & 2 & 2 & 5 & 6 \\ \hline 1 & 1 & 0 & 2 & 5 & 6 \\ \hline 1 & 2 & 2 & 3 & 5 & 6 \\ \hline 3 & 4 & 5 & 5 & 4 & 3 \\ \hline 4 & 5 & 7 & 7 & 4 & 2 \\ \hline \end{array}$$

- Langkah ketiga adalah melakukan interpolasi pada bagian yang berwarna abu-abu. Sebelum mengisi piksel yang kosong, dihitung terlebih dahulu gradien 45° dan 135° sebanyak k , dimana $0 < k < M$ dengan persamaan 2.2 dan 2.3.

Dari persamaan tersebut tersebut diperoleh nilai $G_1^{45^\circ} = [4 \ 1 \ 1 \ 3]$ dan $G_2^{135^\circ} = [2 \ 4 \ 5 \ 2]$. Menggunakan persamaan 2.4 dan ambang batas atau *threshold* sebesar 1, diperoleh $\frac{6}{5} > 1$ bernilai benar. Sehingga $piksel(2,2) = p(2,2) = 2$. Nilai tersebut pembulatan dari nilai 2,2625. Berikut ini hasil dengan cara yang sama.

3		2		5
	2		2	
1		1		6
	2		3	
4		7		3

- Langkah keempat adalah melakukan interpolasi pada bagian yang berwarna kuning dan biru. Sebelum mengisi piksel yang kosong, dihitung terlebih dahulu gradien horizontal ($G_1^{90^\circ}$) dan vertical ($G_2^{0^\circ}$) sebanyak k , dimana $0 < k < M$.

Dari persamaan 2.5 dan 2.6 diperoleh nilai $G_{1_kuning}^{90^\circ} = [4 \ 1]$, $G_{2_biru}^{0^\circ} = [2 \ 4]$, $G_{1_biru}^{90^\circ} = [7 \ 4]$, dan $G_{2_kuning}^{0^\circ} = [2 \ 4]$. Untuk mengisi nilai piksel yang kosong, menggunakan persamaan 2.4.

Dengan menggunakan ambang batas atau *threshold* sebesar 1, diperoleh $\frac{5}{5} > 1$

bernilai salah. Diperoleh nilai $w_1 = \frac{1}{5}$, $w_2 = \frac{1}{5}$, sehingga

$$piksel(1,2) = \frac{(w_1 * p(i,j) + w_2 * p(i,j))}{w_1 + w_2} = 3$$

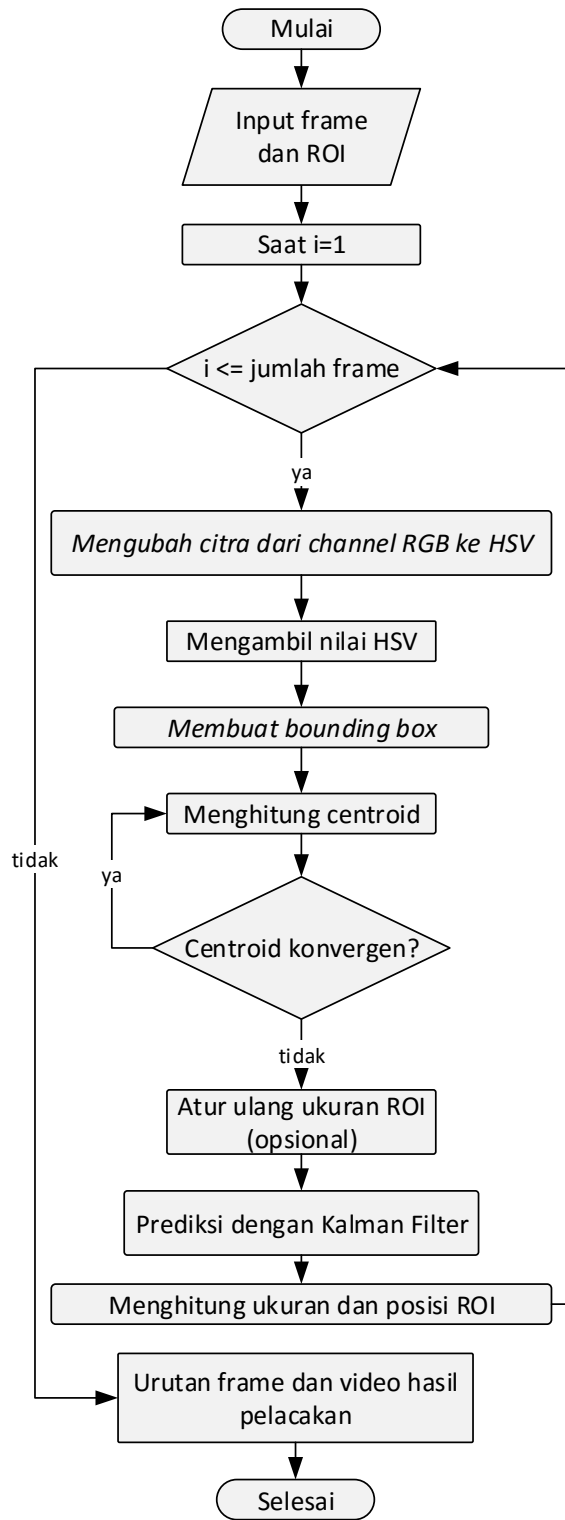
Nilai tersebut pembulatan dari nilai sebenarnya 2,8562. Langkah tersebut diulangi hingga piksel terisi sebagai berikut

3	3	2	3	5
3	2	3	2	5
1	1	1	2	6
1	2	2	3	5
4	4	7	5	3

4.1.2 Pelacakan Objek

Proses pelacakan objek dilakukan dengan menggunakan metode Camshift. Sedangkan prediksi pergerakan objek menggunakan metode Kalman Filter. Metode Kalman Filter tidak berdiri sendiri, tetapi terletak didalam proses pelacakan pada metode Camshift. Sehingga *input* metode Kalman bergantung pada *output* metode Camshift dan begitu sebaliknya.

Urutan *frame* untuk pelacakan objek adalah *frame* hasil super-resolusi. Objek yang akan dilacak telah didefinisikan melalui pemilihan ROI objek. Meskipun letak objek pada *frame* pertama telah diketahui, proses pelacakan 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 Camshift dan Kalman Filter dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram alir proses pelacakan dengan metode kombinasi.

Pelacakan objek dimulai dengan mengubah warna RGB ke *channel* warna HSV. Nilai HSV yang sering digunakan untuk proses penghitungan probability distribution adalah *channel* 1 (H) atau *channel* 2 (S). Pemilihan *channel* warna tersebut berdasarkan dari penelitian-penelitian sebelumnya. Langkah selanjutnya adalah membuat *bounding box* atau *search window* pada *frame* atau gambar yang sedang dilakukan pelacakan. Setelah didapatkan *bounding box*, langkah selanjutnya adalah menghitung *centroid* dari ROI yang telah dipilih, sehingga didapat pusat pencarian dari objek yang dilacak. Nilai *centroid* tersebut digunakan sebagai informasi untuk menentukan prediksi objek pada *frame* selanjutnya.

Dalam menentukan letak objek pada *frame* selanjutnya dengan metode Kalman Filter. *Output* metode Camshift berupa *centroid* dan koordinat *bounding box*. Kedua *output* tersebut digunakan untuk *input* dari metode Kalman Filter. *Output* metode kalman Filter juga berupa *centroid* dan koordinat *bounding box*. Kedua *output* tersebut menjadi *input* Camshift untuk *frame* selanjutnya. Proses ini dilakukan pada semua *frame* dari video.

Langkah-langkah pengerjaan pelacakan objek dengan metode Camshift dan Kalman Filter adalah sebagai berikut:

1. Menentukan ukuran dan lokasi awal serta daerah kalkulasi *search window*.

Langkah pertama dari metode Camshift diperoleh dari *input* berupa ROI. Dalam pemilihan ROI objek diperoleh koordinat x , y , perpanjangan x , dan perpanjangan y . Dari keempat *input* tersebut digunakan untuk menentukan daerah kalkulasi dengan ukuran lebih besar dari *search window*. Gambar 4.5 adalah contoh objek yang telah ditandai dengan *rectangle* memiliki koordinat *frame* ke dua $x = 474.982$, $y = 305$, perpanjangan $x = 15$, dan perpanjangan $y = 28$.



Gambar 4.5 Pemilihan objek yang akan dilacak.

2. Konversi citra dari RGB ke HSV.

HSV (hue, saturation, value) adalah model warna yang lebih banyak menggambarkan fitur warna daripada model RGB (red, green, blue). Kelebihan dari model warna HSV adalah tidak sensitif terhadap cahaya. Gambar 4.6 adalah hasil konversi warna dari Gambar 4.5 ke bentuk warna HSV.



Gambar 4.6 (a) Citra hasil konversi RGB ke HSV, (b) Citra HSV dengan nilai Hue (1)

3. Menghitung *centroid* dari objek yang dilacak.

Lakukan algoritma mean shift pada langkah ini (satu atau banyak iterasi) dengan *input* berupa ukuran dan lokasi *search window* serta citra distribusi probabilitas warna dan simpan *zeroth moment* seperti yang telah dijelaskan di Bab 2.

Dari persamaan 2.10 dengan *input* berupa $x = 474.9820, y = 305$, perpanjangan $x = 15$, dan perpanjangan $y = 28$ diperoleh *centroid* dengan koordinat $x = 318.982, y = 483$. Koordinat tersebut digunakan untuk prediksi pergerakan objek dengan Kalman Filter.

4. Prediksi gerakan dari objek dengan Kalman Filter.

Prediksi pergerakan objek memanfaatkan *output* dari metode Camshift, selanjutnya digunakan sebagai *input* untuk memprediksi dengan metode Kalman Filter. Merujuk pada Bab 2, langkah pertama adalah tahapan prediksi dan tahap kedua adalah koreksi. Berikut ini penyelesaiannya.

a. Tahap prediksi

Persamaan prediksi 1 : $X_{pred_k} = A \cdot X_{k-1}$

$$X_{pred_k} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 318.982 \\ 483 \\ 0.9371 \\ 0 \end{bmatrix} = \begin{bmatrix} 319.9191 \\ 483 \\ 0.9371 \\ 0 \end{bmatrix}$$

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

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

b. Tahap koreksi

Persamaan Kalman-gain : $K_k = P_{pred_k} \cdot H^T \cdot (H \cdot P_{pred_k} \cdot H^T + R)^{-1}$

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

$$K_k = \begin{bmatrix} 0.8841 & 0 \\ 0 & 0.8841 \\ 0.546 & 0 \\ 0 & 0.546 \end{bmatrix}$$

$$\text{Persamaan update 1 : } X_k = X_{pred_k} + K_k \cdot (Z_k - H \cdot X_{pred_k})$$

$$X_k = \begin{bmatrix} 319.9191 \\ 483 \\ 0.9371 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.8841 & 0 \\ 0 & 0.8841 \\ 0.546 & 0 \\ 0 & 0.546 \end{bmatrix} \cdot \left(\begin{bmatrix} 319 \\ 483 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 319.9191 \\ 483 \\ 0.9371 \\ 0 \end{bmatrix} \right)$$

$$X_k = \begin{bmatrix} 319.9906 \\ 483 \\ 0.9813 \\ 0 \end{bmatrix}$$

$$\text{Persamaan update 2 : } P_k = P_{pred_k} - K_k \cdot H \cdot P_{pred_k}$$

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

$$P_k = \begin{bmatrix} 7.6317 & 0 & 4.7126 & 0 \\ 0 & 7.6317 & 0 & 4.7126 \\ 4.7126 & 0 & 4.7755 & 0 \\ 0 & 4.7126 & 0 & 4.7755 \end{bmatrix}$$

$$P_k = \begin{bmatrix} 0.8841 & 0 & 0.546 & 0 \\ 0 & 0.8841 & 0 & 0.546 \\ 0.546 & 0 & 2.2026 & 0 \\ 0 & 0.546 & 0 & 2.2026 \end{bmatrix}$$

Hasil penghitungan diperoleh nilai *centroid* dengan koordinat $x = 319.9906$, $y = 483$ dari nilai X_k baris pertama dan kedua. *Input frame* selanjutnya adalah koordinat *rectangle* (L) dari penghitungan *centroid* ditentukan dengan cara sebagai berikut.

$$L_x = centroidx(i) - (centroidx(i-1) - L_{x-1}) = 320 - (319 - 306) = 307$$

$$L_y = centroidy(i) - (centroidy(i-1) - L_{y-1}) = 483 - (483 - 475) = 475$$

Dengan nilai perpanjangan koordinat x dan y yang sama seperti *input* awal yaitu, perpanjangan $x = 15$, dan perpanjangan $y = 28$.

Langkah-langkah pelacakan objek dengan metode Camshift dan Kalman Filter 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

Setelah perancangan proses, tahapan selanjutnya adalah perancangan perangkat lunak. Perancangan perangkat lunak meliputi lingkungan perancangan dan perancangan data perangkat lunak.

4.2.1 Lingkungan Perancangan

Lingkungan perancangan dalam Tesis ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasinya disajikan dalam Tabel 4.1.

Tabel 4.1 Lingkungan Perancangan Perangkat

Perangkat keras	<ol style="list-style-type: none">1. Intel Pentium CPU 2020M @2.40GHz dengan AMD Radeon R5 M200/HD 8500M Series, memory 2048 MB2. Memory 4096 MB RAM3. Web Camera Logitech
Perangkat lunak	<ol style="list-style-type: none">2. Sistem operasi Microsoft Windows 10 Pro with Media Center 64 bit3. Tools menggunakan MATLAB R2015a.

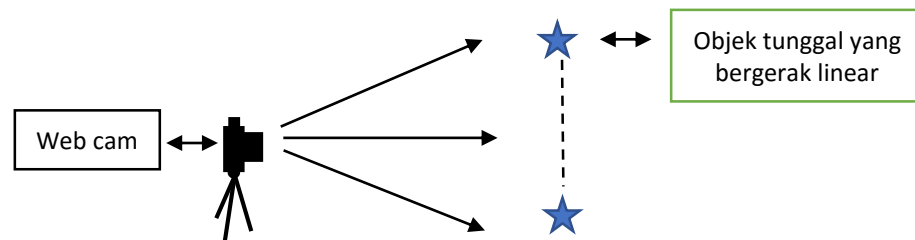
Dengan perangkat keras dan lunak seperti dijelaskan pada Tabel 4.1 diharapkan memperoleh hasil perangkat lunak yang efisien dari segi efisiensi waktu komputasi dan tentunya ketepatan pelacakan objek.

4.2.2 Perancangan Proses Akuisisi Data

Data yang digunakan untuk pelacakan objek bergerak terdiri dari data masukan, data proses, dan data keluaran. Data masukan merupakan video rekaman *offline* yang dilakukan secara mandiri oleh peneliti dan data yang telah disebutkan pada bab sebelumnya. 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.

4.2.2.1 Data Masukan

Data video yang digunakan memiliki spesifikasi sesuai dengan batasan masalah yang terdapat dalam penelitian ini. Pengambilan rekaman video dilakukan di sekitar kampus ITS. Data lainnya didapatkan dari *dataset* yang telah disediakan pada VIRAT *dataset* [11] dan situs Youtube. Proses akuisisi video yang dilakukan dalam penelitian ini disajikan pada Gambar 4.7.



Gambar 4.7 Proses akuisisi video

Hasil rekaman video tersebut akan dijadikan data masukan dari 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 karena akan mempengaruhi banyak *frame* saat konversi dari video ke urutan *frame*. Kamera digital (*web camera*) yang digunakan untuk proses akuisisi data memiliki ± 25 FPS.

4.2.2.2 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 Tabel Data Proses

No	Tahapan	Input	Output
1.	Input data	Video	Frame citra
2.	Konversi video ke urutan <i>frame</i>	Video	Frame citra
3.	Menentukan ROI	Frame citra super-resolusi	Citra ROI
4.	Pemotongan <i>frame</i> dan Super-resolusi	Frame citra	Frame citra super-resolusi
5.	Proses pelacakan dengan metode Camshift	Frame <i>i</i> dan koordinat ROI	Centroid dan koordinat ROI
6.	Proses prediksi dengan metode Kalman Filter	Centroid dan koordinat ROI dari langkah 5	Centroid, koordinat ROI prediksi, dan hasil pelacakan

4.2.2.3 Data Keluaran

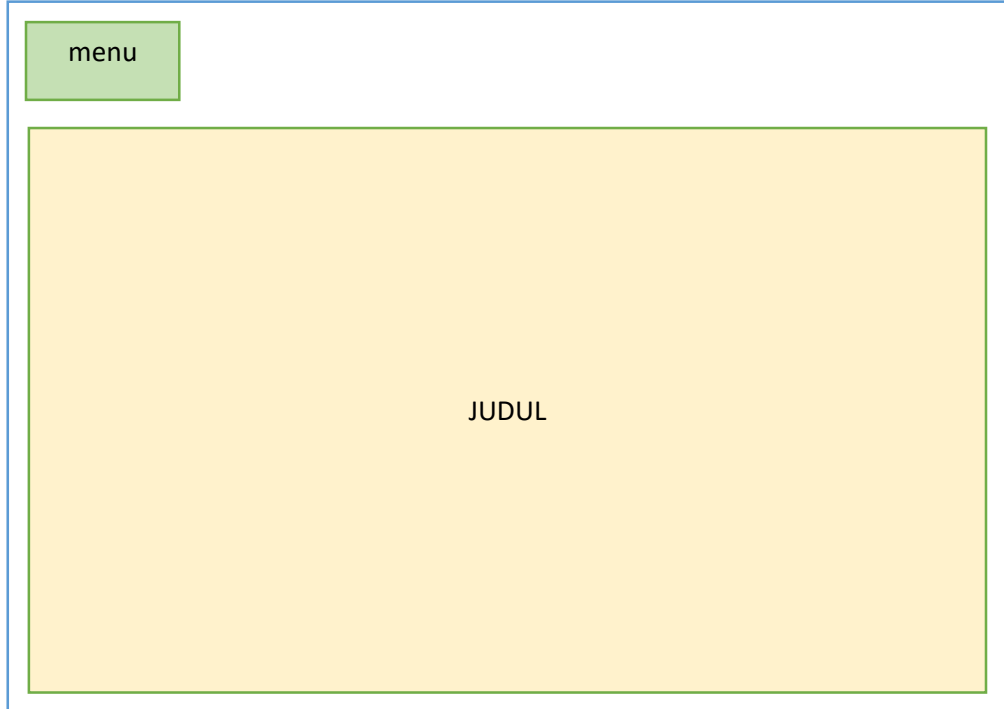
Data keluaran yang dihasilkan dari rancangan perangkat lunak ini adalah berupa semua *frame* dan video hasil pelacakan objek dengan penandaan posisi target (objek) dengan *bounding box* di setiap *frame*.

4.3 Implementasi Perangkat Lunak pada MATLAB

Tampilan *user interface* atau antar muka sistem dibuat semenarik dan semudah mungkin. Tujuannya adalah untuk mempermudah dalam mengoperasikan perangkat lunak. Berikut ini tampilan desain antar muka dan implementasinya yang telah dibuat.

a. Halaman Awal

Perancangan antar muka halaman awal ditunjukkan pada Gambar 4.8. Form ini adalah form awal dari keseluruhan perangkat lunak. Pada halaman ini hanya terdapat satu menu yaitu **Tracking** yang berfungsi membawa pengguna ke halaman utama seperti ditunjukkan pada Gambar 4.9.



Gambar 4.8 Rancangan halaman awal perangkat lunak.

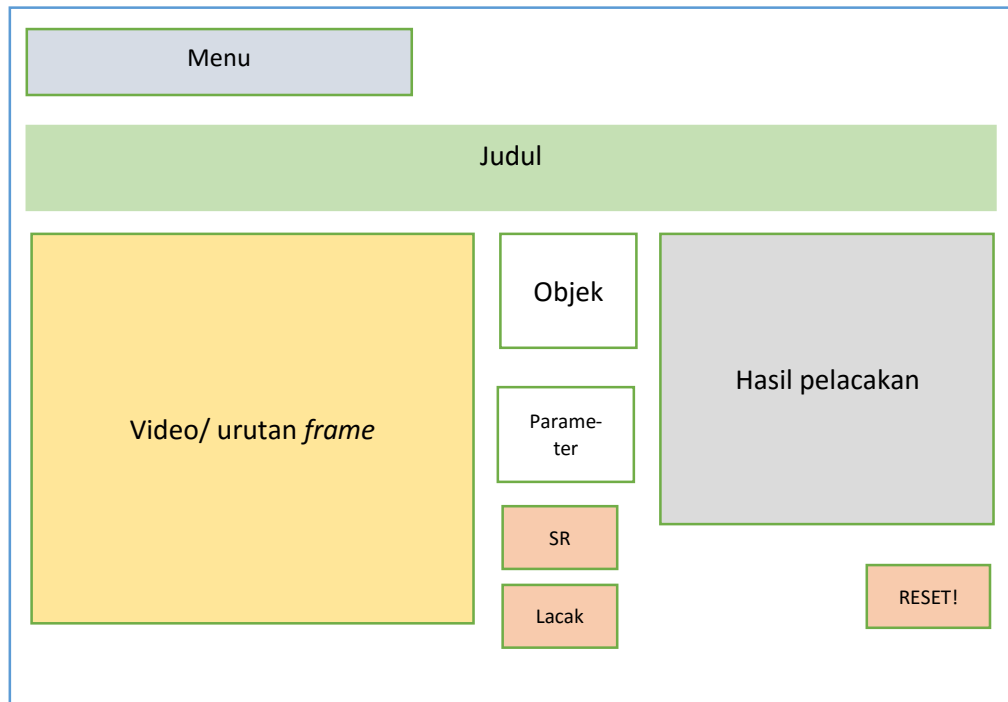


Gambar 4.9 Halaman awal perangkat lunak.

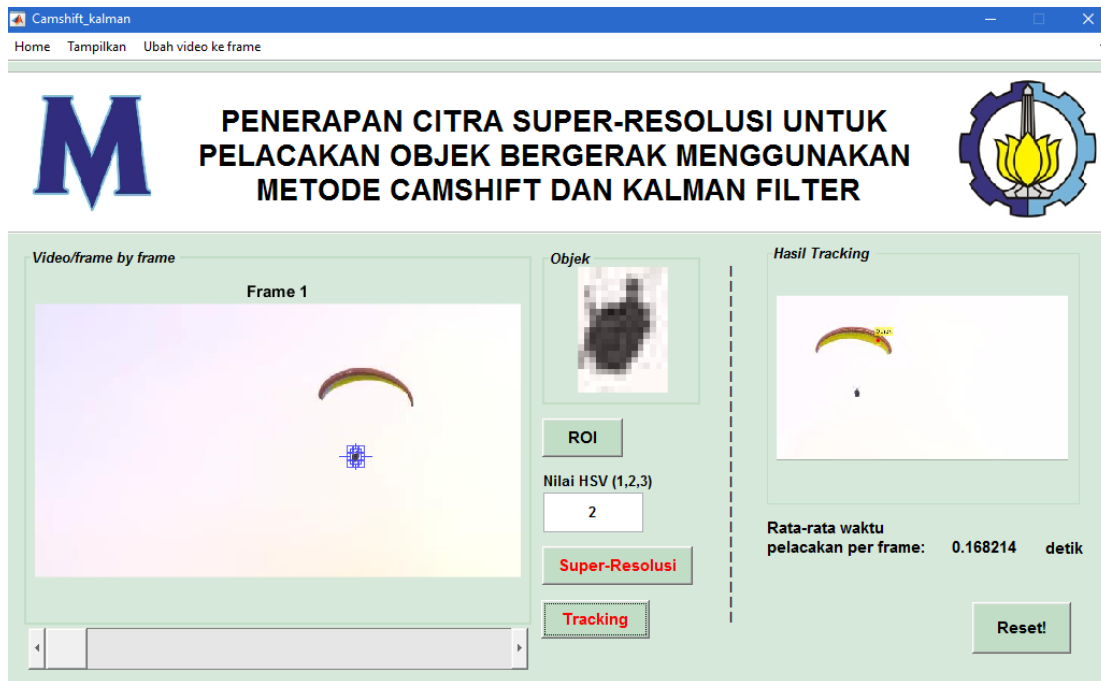
b. Halaman Utama

Perancangan antar muka halaman utama ditunjukkan pada Gambar 4.10. Pada halaman Gambar 4.11 dilakukan proses pemilihan video, pra pengolahan video, pemilihan ROI, proses super-resolusi, pengisian parameter warna, dan proses pelacakan. Pada halaman ini juga menampilkan video hasil pelacakan objek. Antar muka halaman utama terdiri dari:

1. Menu **Tampilkan** memiliki dua sub-menu, yaitu **video** dan **frame by frame**. Sub-menu tersebut berfungsi untuk memilih video/frame by frame yang akan dilacak.
2. Menu **Ubah video ke frame**, berfungsi untuk mengubah video menjadi urutan **frame** (citra).
3. **Axes1**, berfungsi untuk menampilkan video atau urutan **frame** dari video yang telah dipilih oleh pengguna.
4. **Axes2**, berfungsi untuk menampilkan objek yang akan dilacak pergerakannya. Objek ini diperoleh setelah pengguna memilih ROI objek pada **frame** pertama yang ditampilkan pada axes1.
5. **Axes3**, berfungsi untuk menampilkan hasil pelacakan objek.
6. Push button **ROI**, berfungsi untuk memilih objek yang akan dilacak atau menentukan wilayah yang akan dilakukan super-resolusi.
7. Edit text **Nilai HSV (1, 2, 3)**, berfungsi agar pengguna dapat menentukan nilai HSV yang digunakan. Setiap video memiliki karakteristik tersendiri dalam menentukan nilai HSV yang dipakai.
8. Push button **Super-resolusi** digunakan untuk melakukan proses super-resolusi **single frame** dari koordinat yang dipilih pada langkah 4.
9. Push button **Tracking** digunakan untuk melakukan proses pelacakan objek dengan metode Camshift dan Kalman Filter dari koordinat yang dipilih pada langkah 4.
10. Push button **Reset** digunakan untuk menghapus semua historis dari proses sebelumnya.



Gambar 4.10 Perancangan halaman utama perangkat lunak.



Gambar 4.11 Halaman utama perangkat lunak.

4.4 Implementasi Proses Super-resolusi Citra dan Pelacakan Objek

Implementasi proses super-resolusi dan pelacakan objek terdiri dari beberapa tahapan meliputi proses pengambilan *input* video, proses super-resolusi, proses pemilihan ROI, proses pelacakan objek dengan metode Camshift, dan proses prediksi pergerakan benda menggunakan Kalman Filter. Berikut ini penjelasan pada setiap proses yang dilewati.

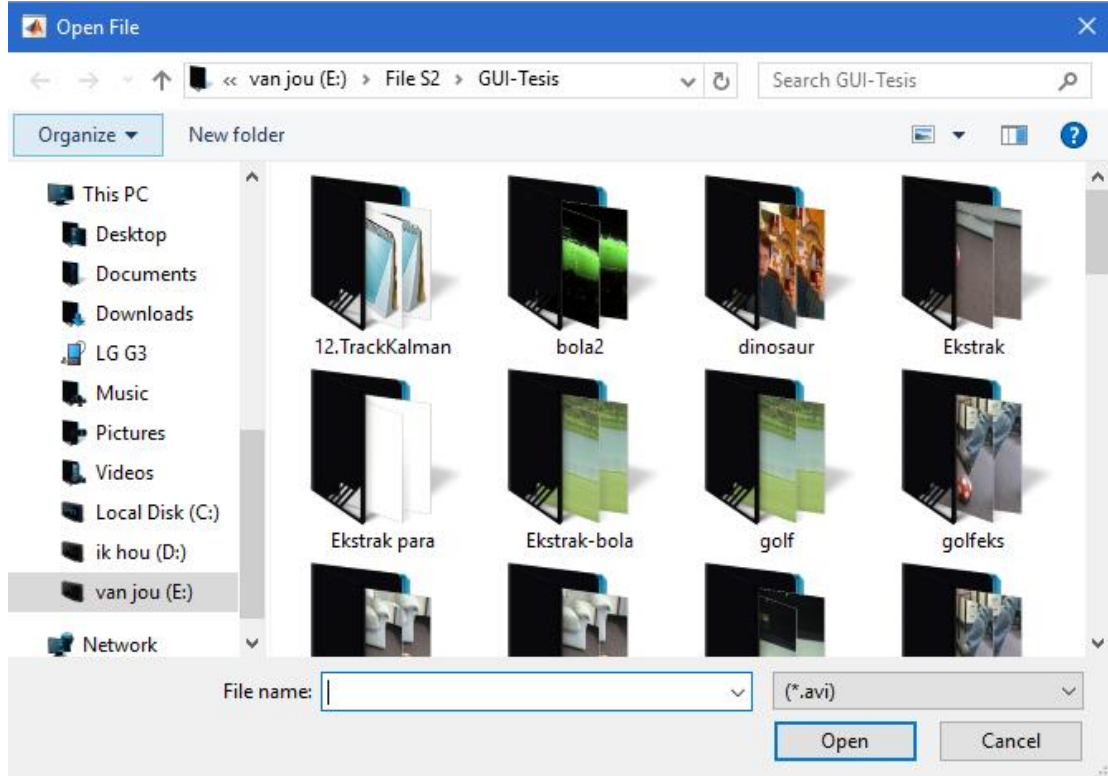
4.4.1 Proses Pengambilan *Input* Video dan Urutan *Frame*

Pengguna akan memilih video yang telah tersimpan pada komputer untuk digunakan sebagai data masukan untuk diproses. Gambar 4.12 adalah tampilan antar muka pengambilan *input* video. Penjabaran tentang proses pemilihan video adalah sebagai berikut.

Fungsi : *Input* video bertipe *.avi pada form simulasi.
Input : Video bertipe *.avi
Output : Video bertipe *.avi
Deskripsi : Mengambil video bertipe *.avi yang telah disimpan pada komputer.

Kode program untuk *input* video adalah sebagai berikut :

```
[filename, pathname] = uigetfile('*.avi','Open File');  
if ~isequal(filename,0)  
    fullname = fullfile(pathname, filename);  
end  
vid = vision.VideoFileReader(fullname);  
i = 1;  
while ~isDone(vid)  
    frame = step(vid);  
    axes(handles.ImageAxes);  
    imshow(frame);  
    pause(0.0000000000000001)  
    i = i + 1;  
end
```



Gambar 4.12 Halaman antar muka *input* video.

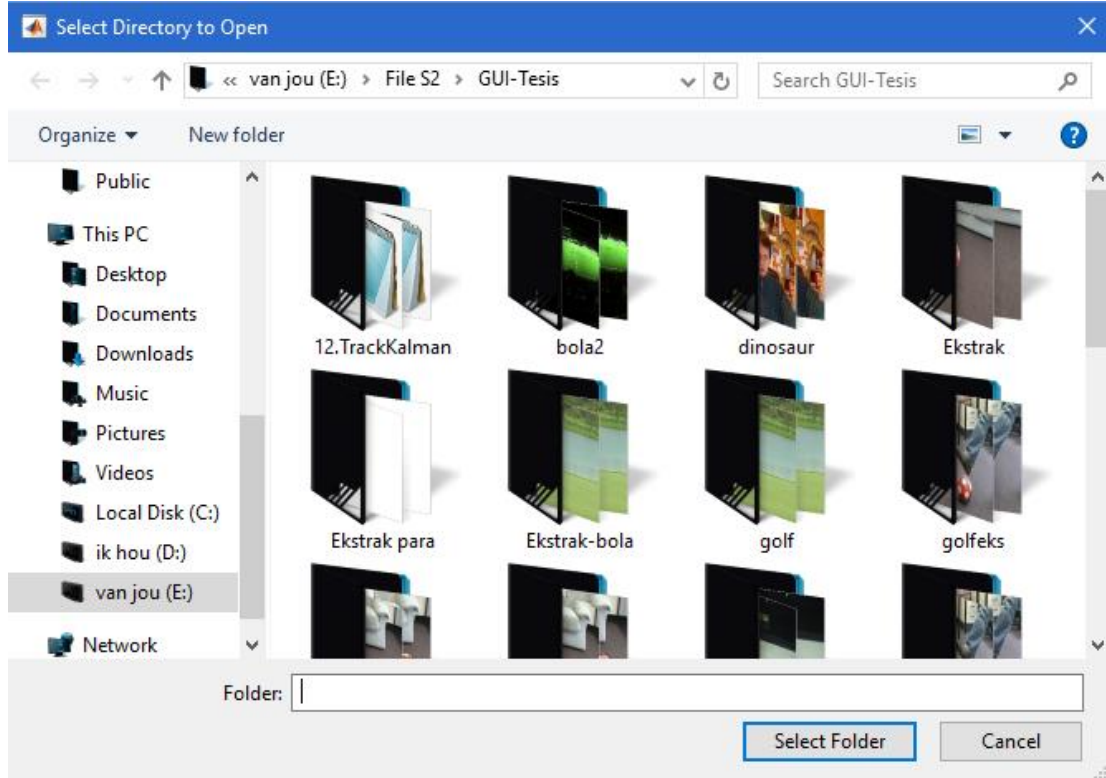
4.4.2 Proses Konversi Video ke Urutan *Frame*

Video yang digunakan sebagai *input* selanjutnya akan dirubah menjadi urutan *frame*. Gambar 4.13 adalah tampilan antar muka pengambilan *input* urutan *frame*. Penjabaran tentang proses konversi video kedalam urutan *frame* adalah sebagai berikut.

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

Kode program untuk *input* urutan *frame* adalah sebagai berikut :

```
vidi = vision.VideoFileReader(fullname);
name = 'Directory untuk menyimpan hasil konversi frame';
E = mkdir(name);
selection = questdlg(['Apakah anda ingin mengubah video ke
frame by frame?'],...
['Ubah ke frame'],...
'Ya','Tidak','Ya');
if strcmp(selection, 'Ya')
    a = vidi;
    i = 1;
    while ~isDone(a)
        frame = step(vidi);
        file = strcat(num2str(i),'.jpg');
        workingDir = name;
        fulln = fullfile(workingDir,file);
        imwrite(frame,fulln);
        i = i + 1;
    end
    msgbox('Selesai, merubah video ke frame sukses.')
    return;
else
    return;
end
```



Gambar 4.13 Halaman antar muka konversi video ke urutan *frame*.

4.4.3 Proses Pemilihan ROI Objek

Region of Interest (ROI) adalah daerah atau bagian dari *frame* atau citra yang didefinisikan sebagai objek yang akan dilacak. Pembentukan ROI objek diperoleh dengan membentuk *rectangle*. Gambar 4.14 adalah tampilan antar muka pemilihan area ROI objek. Penjabaran tentang proses pemilihan ROI objek adalah sebagai berikut.

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.

Kode program untuk ROI objek adalah sebagai berikut :

```
h = imrect;  
posisi = wait(h);  
roi = getPosition(h)  
roi2 = roi;  
target_img = imcrop(c, roi);  
axes(handles.axes4);  
imshow(target_img);
```



Gambar 4.14 Halaman antar muka pemilihan ROI objek.

4.4.4 Proses Pemotongan dan Super-resolusi *Single Frame*

Pemotongan *frame* bertujuan untuk mengecilkan area pencarian objek yang sedang dilacak. Super-resolusi dilakukan untuk membuat *frame* yang telah dipotong menjadi lebih besar, sehingga objek akan tampak terlihat lebih jelas. *Input* dan *output* pada proses ini adalah urutan *frame* dari proses sebelumnya. Gambar 4.15 adalah

tampilan antar muka yang telah dibuat. Penjabaran tentang proses pemotongan dan super-resolusi *frame* adalah sebagai berikut.

Fungsi : *Input* urutan *frame* bertipe *.jpeg pada form simulasi.
Input : Urutan *frame* bertipe *.jpeg dan koordinat ROI
Output : Urutan *frame* super-resolusi bertipe *.jpeg
Deskripsi : Pemotongan dan super-resolusi $2 \times$ urutan *frame*

Kode program untuk pemotongan adalah sebagai berikut :

```
for frame = 1 : jumlah_frame
filename = sprintf('%1.1i.jpg', frame);
R = imread([imgDir '\' filename]);
roip = [roi2(1) roi2(2) roi2(3) roi2(4)];
I = imcrop(R,roip); %proses pemotongan
```

Kode program untuk super-resolusi adalah sebagai berikut :

Inisialisasi

```
Isr = cast(zeros(jml_piksel-1,jml_piksel2-1,3), 'double');
```

Interpolasi langkah pertama

```
mulai = jml_piksel-((2*(jml_piksel-2))+1);
mulai2 = jml_piksel2-((2*(jml_piksel2-2))+1);
```

Menghitung gradient arah 45° dan 135°

```
for m=mulai:2:jml_piksel
    for n = mulai2:2:jml_piksel2
        G1(index) = abs(Isr(i+m, j-n) - Isr(i+m-2, j-n+2));
        G2(index) = abs(Isr(i+m, j+n) - Isr(i+m-2, j+n-2));
        index = index+1;
    end
end
```

Menghitung gradient arah horizontal dan vertikal

```
for m=mulai:2:-mulai+1
    for n = 0:2:jml_piksel2-1
        G11(index) = abs(Isr(i+m, j-n) - Isr(i+m-2, j-n+2));
        for mm=-2:2:jml_piksel-1
            G11(index) = G11(index) + abs(Isr(i+mm, j-1) -
Isr(i+mm, j+1));
        end
        G21(index) = abs(Isr(i-m, j+n) - Isr(i-m-2, j+n-2));
        for nn=-2:2:jml_piksel-1
            G21(index) = G21(index) + abs(Isr(i-1, j+nn) -
Isr(i+1, j+nn));
        end
    end
end
```

```

        index = index+1;
    end
end

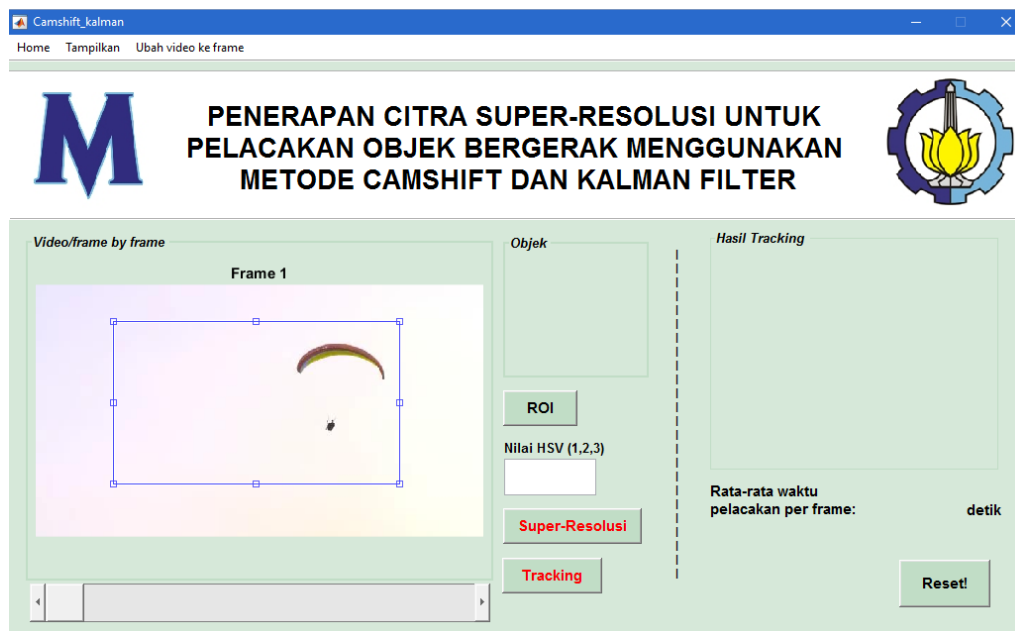
```

Menghitung piksel yang kosong

```

for k = 1 : jml_piksel
    for l = 1 : jml_piksel2-1
        if ((1 + sum(G1l))/(1 + sum(G2l))) > T
            p1 = Ib(2*k-1,2*l,1);
            p2 = Ib(2*k-1,2*l,2);
            p3 = Ib(2*k-1,2*l,3);
        elseif ((1 + sum(G1l))/(1 + sum(G2l))) < T
            p1 = Ib(2*k-1,2*l,1);
            p2 = Ib(2*k-1,2*l,2);
            p3 = Ib(2*k-1,2*l,3);
        else
            w1 = 1/(1+sum(G1l));
            w2 = 1/(1+sum(G2l));
            p1 = (w1*Ib(2*k-1,2*l,1)+w2*Ib(2*k-1,2*l,1))/(w1+w2);
            p2 = (w1*Ib(2*k-1,2*l,2)+w2*Ib(2*k-1,2*l,2))/(w1+w2);
            p3 = (w1*Ib(2*k-1,2*l,3)+w2*Ib(2*k-1,2*l,3))/(w1+w2);
        end
        Isr(2*k-1,2*l,1) = p1;
        Isr(2*k-1,2*l,2) = p2;
        Isr(2*k-1,2*l,3) = p3;
        l=l+1;
    end
    k=k+1;
end

```



Gambar 4.15 Halaman antar muka pemotongan dan super-resolusi.

4.4.5 Proses Pelacakan Objek dengan Metode Kombinasi Camshift dan Kalman Filter

Pada proses ini akan diproses pelacakan objek dengan metode Camshift dan Kalman Filter. Gambar 4.17 menunjukkan halaman pelacakan objek dengan metode kombinasi. Penjabaran tentang proses metode kombinasi Camshift dan Kalman Filter adalah sebagai berikut.

Fungsi : Memproses setiap *frame* untuk dilakukan pelacakan objek yang telah ditentukan.

Input : Urutan *frame* bertipe *.jpeg dan koordinat ROI

Output : Posisi baru dari objek yang dilacak

Deskripsi : Memproses setiap urutan *frame* untuk selanjutnya dilakukan pelacakan objek.

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

1. Menentukan ukuran dan lokasi awal serta daerah kalkulasi *search window*.

Langkah pertama dari metode Camshift ini diperoleh dari *input* berupa ROI. Dalam pemilihan ROI objek diperoleh koordinat x , y , perpanjangan x , dan perpanjangan y . Keempat *input* tersebut digunakan untuk menentukan daerah kalkulasi dengan ukuran lebih besar dari *search window*.

2. Konversi citra dari RGB ke HSV.

HSV (hue, saturation, value) adalah model warna yang lebih banyak menggambarkan fitur warna daripada model RGB (red, green, blue). Kelebihan dari model warna HSV adalah tidak sensitif terhadap cahaya. Berikut ini kode program untuk merubah warna RGB ke HSV.

```
filename = sprintf('%1.1i.jpg', frame);
R = imread([imgDir '\' filename]);
I = rgb2hsv(R);
I = I(:, :, 1/2/3);
```

3. Menghitung *centroid* dari objek yang dilacak.

Penghitungan *centroid* dilakukan dengan metode Camshift seperti yang telah dijelaskan pada Bab 2. Pada langkah ini pengaturan ROI baru tidak dilakukan untuk menghindari hasil *Not a Number* (NaN). Sehingga ukuran ROI dari awal pelacakan hingga berakhirnya *frame* video akan berukuran sama. Berikut ini kode program untuk penghitungan *centroid* dengan metode Camshift.

```
MeanConverging = 1;
while MeanConverging,
    M00 = 0.0;
    M10 = 0.0;
    M01 = 0.0;
    for i = L(1)-P : (L(1)+W(1)+P),
        for j = L(2)-P : (L(2)+W(2)+P),
            if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                continue;
            end
            M00 = M00 + double(I(round(i),round(j)));
            M10 = M10 + i * double(I(round(i),round(j)));
            M01 = M01 + j * double(I(round(i),round(j)));
        end
    end
    xc = round(M10 / M00)
    yc = round(M01 / M00)

    oldL = L
    L = [floor(xc - (W(1)/2)) floor(yc - (W(2)/2))]
    % Cek threshold
    if abs(oldL(1)-L(1)) < T || abs(oldL(2)-L(2)) < T
        MeanConverging = 0;
    end
end
```

4. Prediksi dan koreksi gerakan dari objek dengan Kalman Filter.

Prediksi pergerakan objek memanfaatkan *output* dari metode Camshift berupa koordinat *centroid* dan koordinat *rectangle*, selanjutnya digunakan sebagai *input* untuk memprediksi dengan metode Kalman Filter. Berikut ini kode program untuk prediksi dan koreksi gerakan dari objek dengan Kalman Filter.

```
Inisialisasi
RR=[[1,0]', [0,1]'];
H=[[1,0]', [0,1]', [0,0]', [0,0]'];
Q=1*eye(4);
PP = 100*eye(4);
dt=1;
```

```

A=[ [1,0,0,0]', [0,1,0,0]', [dt,0,1,0]', [0,dt,0,1]'];
kfinit = 0;
th = 50;

xcorner = L(2);
ycorner = L(1);
xwidth = W(2);
ywidth = W(1);
centroidx(frame)= xc;
centroidy(frame)= yc;

kalmanx = centroidx(frame)- xcorner
kalmany = centroidy(frame)- ycorner

Persamaan prediksi 1
    if kfinit == 0
        predicted =[centroidx(frame),centroidy(frame),0,0]' ;
    else
        predicted = A*actual(frame-1,:)' %persamaan prediksi 1
    end

    kfinit = 1;
Persamaan prediksi 2
    Ppre = A*PP*A' + Q %persamaan prediksi 2

Persamaan Kalman gain
    K = Ppre*H'/(H*Ppre*H'+RR) %persamaan kalman-gain

Update persamaan prediksi 1
    actual(frame,:) = (predicted +
        K*([centroidx(frame),centroidy(frame)]' -
        H*predicted))' %update persamaan 1

Update persamaan prediksi 2
    PP = (eye(4)-K*H)*Ppre %update persamaan 2

```



Gambar 4.16 Halaman pelacakan objek dengan metode Camshift dan Kalman Filter

BAB V





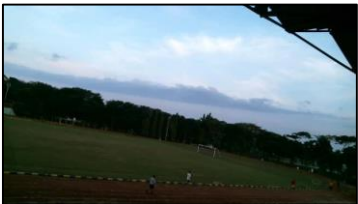

UJI COBA DAN ANALISA HASIL

Bab ini menjelaskan uji coba perancangan proses pelacakan pada bab sebelumnya yang telah dibuat dan kemudian membahas hasil dari setiap uji coba video. Uji coba yang pertama adalah pengujian pelacakan objek tanpa super-resolusi dengan menggunakan metode Camshift dan metode kombinasi Camshift dan Kalman Filter. Tujuan dari uji coba ini adalah untuk melihat tingkat presisi pelacakan objek tanpa proses super-resolusi. Tahap kedua adalah pengujian pelacakan objek dengan super-resolusi dengan menggunakan metode Camshift dan metode kombinasi Camshift dan Kalman Filter. Tujuan dari uji coba ini adalah untuk melihat tingkat presisi pelacakan objek dengan dikenai proses super-resolusi terlebih dahulu. Tahap terakhir dari ujicoba adalah membandingkan akibat dari penambahan proses super-resolusi dan metode Kalman Filter pada proses pelacakan. Uji coba terakhir bertujuan untuk melihat efisiensi waktu dan tingkat akurasi pelacakan objek.

5.1 Data Uji Coba

Uji coba pada program dilakukan pada video berekstensi .avi. Video uji coba telah tersimpan dalam penyimpanan komputer dan diperoleh dari akusisi di lingkungan kampus ITS, Youtube *dataset*, dan VIRAT *dataset* [11]. Video yang digunakan sebanyak 6 video. Terdapat 1 video dengan kasus khusus, yaitu video ke-6, dengan karakteristik video berupa objek dan kamera bergerak secara bersamaan. Daftar *input* video uji coba tersebut antara lain disajikan dalam Tabel 5.1.


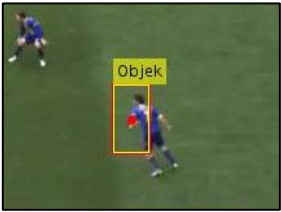
Tabel 5.1 *Dataset* Video Uji Coba Pelacakan

No	Nama	Jumlah Frame	Screenshot Video
1	paralayang.avi	167	
2	soccer.avi	76	
3	ball.avi	54	
4	surveillance.avi	111	
5	jogging.avi	237	
6	bicycle.avi (kasus khusus)	151	


Screenshot video pada Tabel 5.1 memiliki ukuran yang berbeda-beda. Video *paralayang.avi* dan *jogging.avi* memiliki ukuran 640×360 piksel. *Soccer.avi* dan *ball.avi* berukuran 856×480 piksel. Video *surveillance.avi* berukuran 1280×720 piksel. Video terakhir uji coba adalah *bicycle.avi* memiliki ukuran 320×240 piksel. Setiap video memiliki karakteristik yang berbeda-beda., yaitu perbedaan objek dengan *background*, pergerakan objek yang tidak linear, pergerakan kamera, jumlah objek yang bergerak dalam satu *frame*, dan *zoom in* objek.

Hasil pelacakan objek berupa urutan *frame* dimana setiap *frame* terdapat objek yang ditandai dengan *rectangle*. Pada penelitian ini diberikan 3 tingkat ketepatan pelacakan objek. Definisi tingkat ketepatan pelacakan diberikan pada Tabel 5.2. Sedangkan Tabel 5.3 menyajikan cara menghitung tingkat ketepatan pelacakan berdasarkan definisi pada Tabel 5.2.

Tabel 5.2 Definisi Tingkat Ketepatan Pelacakan Objek

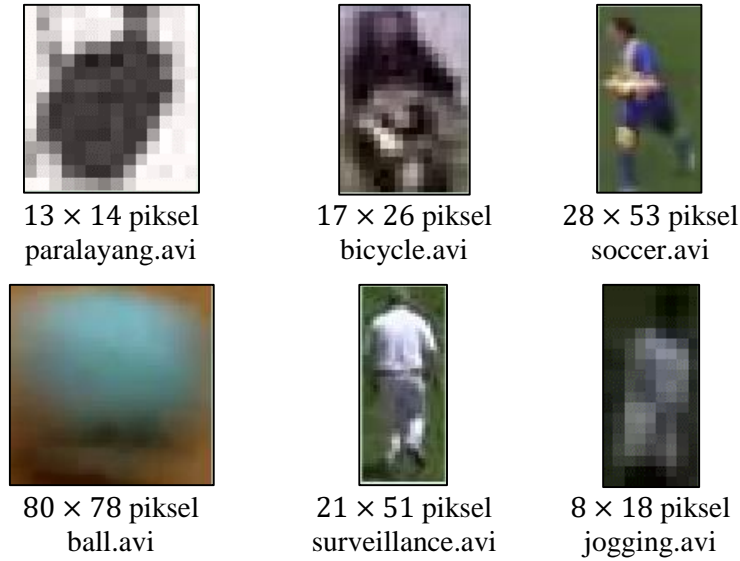
No	Nama	<i>Screenshot</i>
1	Presisi	
2	Kurang Presisi	
3	Tidak Presisi	

Tabel 5.3 Cara Menghitung Tingkat Ketepatan Pelacakan Objek

<i>Frame</i>	<i>Screenshot</i>	Presisi	Kurang presisi	Tidak presisi
1		1	-	-
2		1	-	-
...
9		-	1	-
...
151		-	-	1
Jumlah	

5.2 Uji Coba Pelacakan Objek tanpa Proses Super-resolusi

Langkah pertama untuk melakukan uji coba tanpa proses super-resolusi adalah pemilihan ROI. Pemilihan ROI bertujuan untuk mengetahui apakah perangkat lunak dapat memperoleh citra objek yang akan dilacak atau tidak. Gambar dan ukuran dari setiap citra objek yang akan dilacak tanpa proses super-resolusi dapat dilihat pada Gambar 5.1.



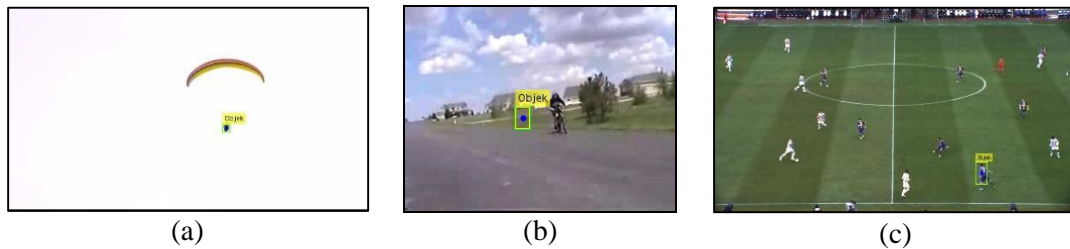
Gambar 5.1 Citra objek yang dilacak.

Pelacakan objek dari video diatas dilakukan tanpa proses super-resolusi. Tujuan dari uji coba ini adalah untuk melihat tingkat presisi pelacakan objek tanpa proses super-resolusi. Penghitungan *frame* hasil pelacakan berdasarkan Tabel 5.2 dan 5.3. Tabel 5.4 menyajikan hasil pelacakan objek tanpa proses super-resolusi dengan metode Camshift. Tabel 5.5 menyajikan hasil pelacakan objek tanpa proses super-resolusi dengan metode kombinasi Camshift dan Kalman Filter.

Tabel 5.4 Hasil Pelacakan Objek dengan Metode Camshift

No	Nama	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)			Total Waktu Pelacakan (detik)
			Presisi	Kurang Presisi	Tidak Presisi	
1	paralayang.avi	167	51	2	114	32,7841
2	soccer.avi	76	56	20	-	20,6456
3	ball.avi	54	54	-	-	18,0828
4	surveillance.avi	111	76	2	33	29,8863
5	jogging.avi	237	195	42	-	31,0074
Kasus khusus						
1	bicycle.avi	151	51	4	96	11,2887

Pelacakan objek menghasilkan nilai persentase 100% dengan metode Camshift pada video ball.avi. Pada video tersebut objek adalah bola tunggal yang digelindingkan dengan *background* polos. Sehingga, metode Camshift dapat dengan tepat melakukan pelacakan pada video ball.avi. Gambar 5.2 menyajikan beberapa contoh hasil pelacakan objek dengan metode Camshift tanpa proses super-resolusi.



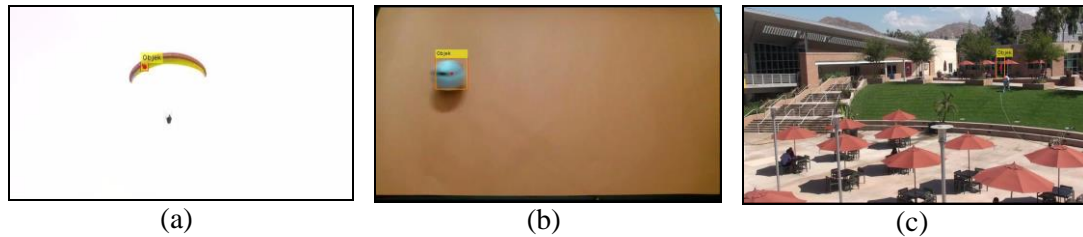
Gambar 5.2 (a) Citra hasil pelacakan objek *frame* 33 pada video paralayang.avi, (b) Citra hasil pelacakan objek *frame* 18 pada video bicycle.avi, dan (b) Citra hasil pelacakan objek *frame* 75 pada video soccer.avi.

Tabel 5.5 Hasil Pelacakan Objek dengan Metode Kombinasi

No	Nama	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)			Total Waktu Pelacakan (detik)
			Presisi	Kurang Presisi	Tidak Presisi	
1	paralayang.avi	167	53	0	114	29,1163
2	soccer.avi	76	58	18	-	17,3256
3	ball.avi	54	54	-	-	16,1417
4	surveillance.avi	111	76	3	32	28,0908
5	jogging.avi	237	206	31	-	30,3146
Kasus khusus						
1	bicycle.avi	151	65	12	74	9,9326

Pelacakan objek juga menghasilkan nilai persentase 100% dengan metode kombinasi pada video ball.avi. Terdapat peningkatan tingkat presisi pelacakan dengan metode kombinasi pada video paralayang.avi dan soccer.avi sebesar 2 *frame* dan video bicycle.avi sebesar 14 *frame*. Gambar 5.3 menyajikan beberapa contoh hasil pelacakan dengan metode kombinasi Camshift dan Kalman Filter.

Pelacakan video bicycle.avi mengalami peningkatan tingkat presisi pelacakan dengan menggunakan metode kombinasi. Tingkat presisi pelacakan naik 14 *frame* pada video yang merupakan kasus khusus pada uji coba pelacakan objek tanpa proses super-resolusi.

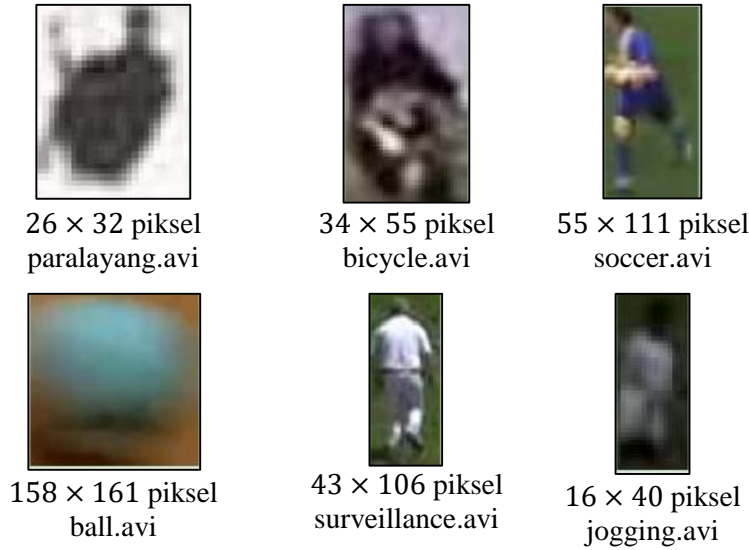


Gambar 5.3 (a) Citra hasil pelacakan objek *frame* 57 pada video paralayang.avi, (b) Citra hasil pelacakan objek *frame* 45 pada video ball.avi, dan (b) Citra hasil pelacakan objek *frame* 108 pada video surveillance.avi.

5.3 Uji Coba Pelacakan Objek dengan Proses Super-resolusi

Langkah pertama untuk melakukan uji coba sama seperti sub-bab sebelumnya, yaitu pemilihan ROI oleh pengguna. Pemilihan ROI bertujuan untuk mengetahui apakah perangkat lunak berhasil memperoleh citra objek yang akan dilacak atau tidak. Gambar dan ukuran dari setiap citra objek yang akan dilacak dengan proses super-resolusi *single frame* dapat dilihat pada Gambar 5.4.

Proses pelacakan objek dilakukan dengan proses super-resolusi terlebih dahulu. Tujuan dari uji coba ini adalah untuk melihat tingkat presisi pelacakan objek berbasis citra super-resolusi. Metode untuk proses super-resolusi adalah interpolasi *Directional Bicubic*. Semua video telah dipotong dengan ukuran tertentu dan disuper-resolusi. Tujuannya adalah untuk mendapatkan citra objek yang lebih tajam, sehingga pelacakan objek akan lebih presisi. Penghitungan *frame* hasil pelacakan berdasarkan Tabel 5.2 dan 5.3. Tabel 5.6 menyajikan hasil pelacakan objek dengan metode Camshift berbasis citra super-resolusi. Tabel 5.7 menyajikan hasil pelacakan objek dengan metode kombinasi Camshift dan Kalman Filter berbasis citra super-resolusi.

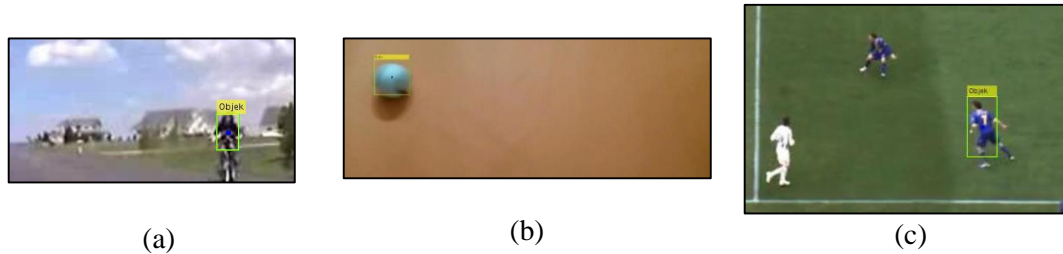


Gambar 5.4 Citra objek super-resolusi yang dilacak.

Tabel 5.6 Hasil Pelacakan Video Super-Resolusi dengan Metode Camshift

No	Nama	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)			Total Waktu Pelacakan (detik)
			Presisi	Kurang Presisi	Tidak Presisi	
1	paralayang.avi	167	163	4	-	50,8823
2	soccer.avi	76	67	9	-	20,8424
3	ball.avi	54	54	-	-	54,7917
4	surveillance.avi	111	80	31	-	17,7734
5	jogging.avi	237	237	-	-	16,8457
Kasus khusus						
1	bicycle.avi	151	147	4	-	17,8996

Pelacakan video super-resolusi menghasilkan nilai persentase 100% dengan metode Camshift pada video ball.avi. Video yang telah dikenai super-resolusi objek akan menjadi $\pm 2 \times$ lebih besar daripada objek pada video biasa. Gambar 5.5 menyajikan contoh hasil pelacakan citra super-resolusi dengan metode Camshift.



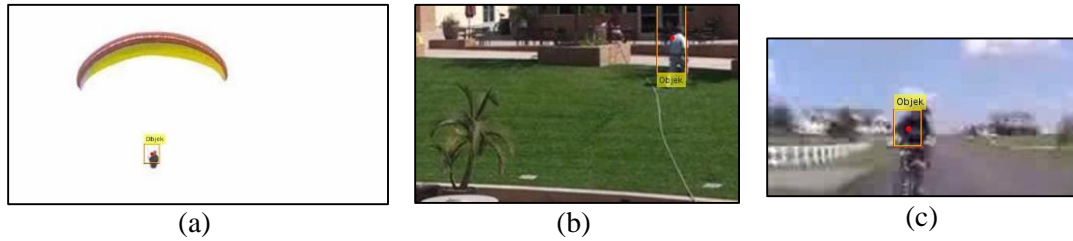
Gambar 5.5 (a) Citra hasil pelacakan objek *frame* 41 pada video *bicycle.avi*, (b) Citra hasil pelacakan objek *frame* 51 pada video *ball.avi*, dan (b) Citra hasil pelacakan objek *frame* 73 pada video *soccer.avi*.

Tabel 5.7 Hasil Pelacakan Video Super-Resolusi dengan Metode Kombinasi

No	Nama	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)			Total Waktu Pelacakan (detik)
			Presisi	Kurang Presisi	Tidak Presisi	
1	<i>paralayang.avi</i>	167	166	1	-	45,6304
2	<i>soccer.avi</i>	76	69	7	-	19,5092
3	<i>ball.avi</i>	54	54	-	-	49,8766
4	<i>surveillance.avi</i>	111	82	29	-	15,5886
5	<i>jogging.avi</i>	237	237	-	-	15,1903
Kasus khusus						
1	<i>bicycle.avi</i>	151	140	5	6	17,6963

Pelacakan objek super-resolusi menghasilkan nilai persentase 100% dengan metode kombinasi pada video *ball.avi*. Terdapat peningkatan tingkat presisi pelacakan sebesar 2 *frame* dengan metode kombinasi pada video *surveillance.avi* dan *soccer.avi*. Peningkatan presisi pelacakan sebesar 3 *frame* pada video *paralayang.avi*.

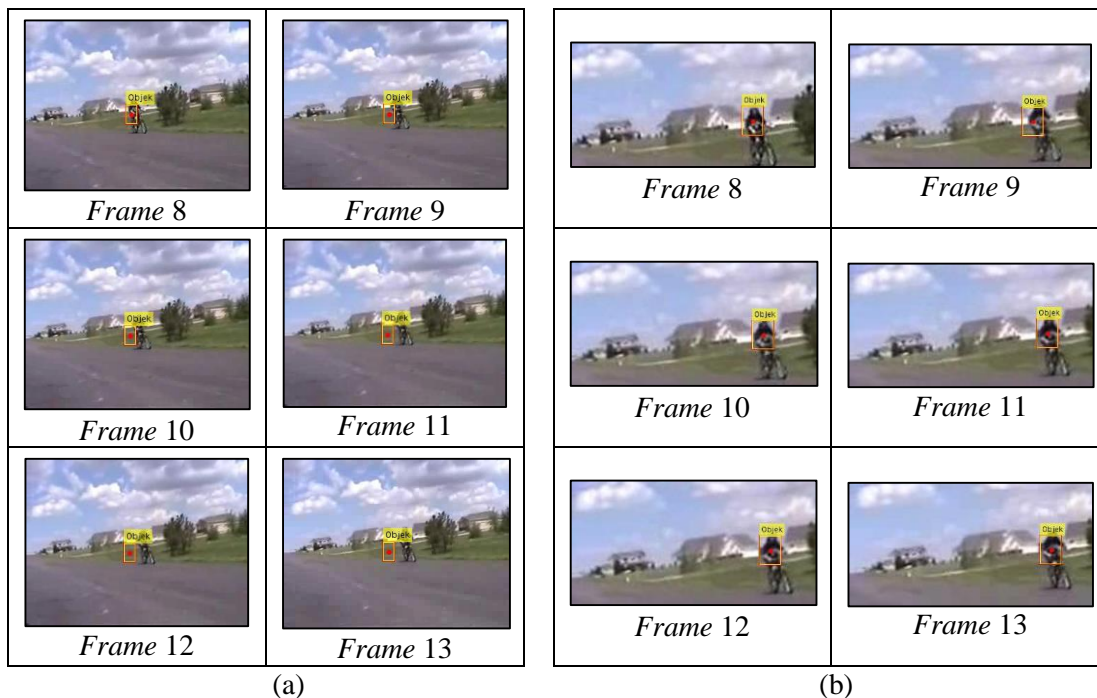
Pelacakan dengan super-resolusi tidak meningkatkan tingkat presisi pada kasus khusus dengan metode kombinasi. Hal tersebut terjadi karena faktor pergerakan kamera dan perubahan bentuk objek. Sehingga, pada proses penghitungan objek terkadang terletak pada koordinat yang sama saat *frame* i sampai $i + 1$. Gambar 5.6 menyajikan beberapa contoh hasil pelacakan dengan proses super-resolusi dengan kombinasi metode Camshift dan Kalman Filter.



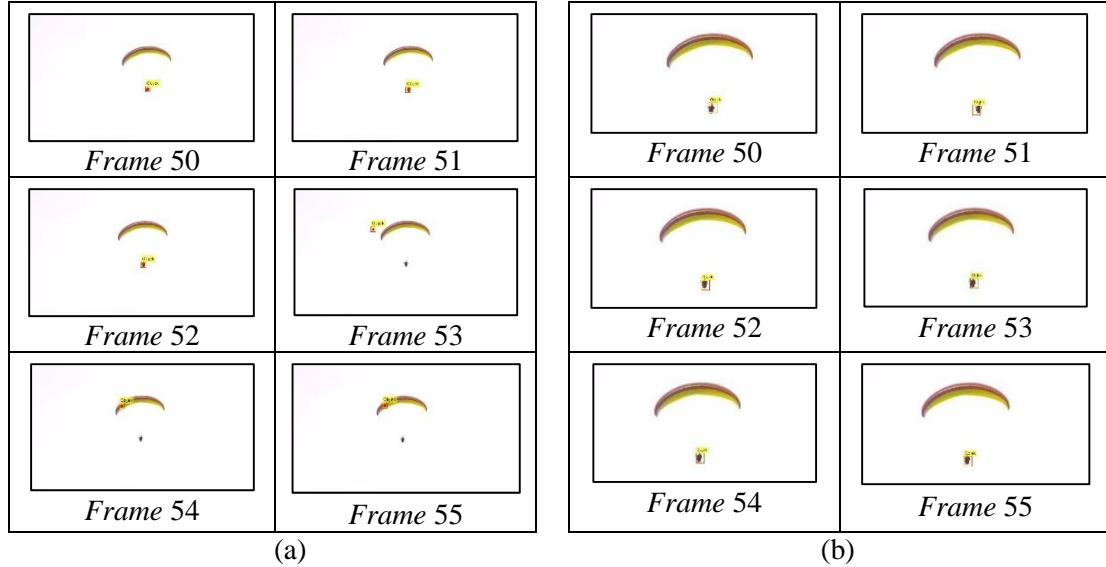
Gambar 5.6 (a) Citra hasil pelacakan objek *frame* 63 pada video *paralayang.avi*, (b) Citra hasil pelacakan objek *frame* 110 pada video *surveillance.avi*, dan (b) Citra hasil pelacakan objek *frame* 148 pada video *bicycle.avi*.

5.4 Pembahasan Hasil Pengujian

Pembahasan pada sub-bab ini difokuskan pada tingkat akurasi pelacakan dengan atau tanpa proses super-resolusi dan akibat penambahan metode Kalman Filter pada proses prediksi objek. Gambar 5.7 dan Gambar 5.8 menyajikan beberapa urutan *frame* hasil pelacakan sebelum dan sesudah super-resolusi dengan metode kombinasi.



Gambar 5.7 Potongan *frame* hasil pelacakan *bicycle.avi* (a) tanpa SR dan (b) dengan SR.



Gambar 5.8 Potongan *frame* hasil pelacakan paralayang.avi (a) tanpa SR dan (b) dengan SR.

Urutan *frame* (Gambar 5.7 dan 5.8) menunjukkan penambahan proses super-resolusi sebelum pelacakan objek akan membuat hasil pelacakan lebih presisi. Pembahasan hasil pengujian digunakan untuk mengetahui tingkat akurasi metode Camshift dan metode kombinasi Camshift dan Kalman Filter dalam melakukan pelacakan objek. Pada subbab ini diberikan cara menghitung persentase akurasi perangkat lunak.

$$PA = \frac{JR}{JS} \times 100\%$$

dengan *PA* adalah persentase akurasi, *JR* adalah jumlah *frame* ketika objek berhasil ditandai secara presisi dengan *rectangle*, dan *JS* adalah jumlah seluruh *frame* video.

Tabel 5.8 sampai Tabel 5.12 menyajikan persentase akurasi secara menyeluruh tanpa video kasus khusus dari kedua metode pelacakan, baik tanpa proses super-resolusi maupun dengan proses super-resolusi.

Tabel 5.8 Hasil Pelacakan Objek Video paralayang.avi

No	Metode	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)	Persentase Akurasi	Rata-rata Waktu Pelacakan per <i>frame</i> (detik)
Tanpa super-resolusi					
1	Camshift	167	51	31%	0,1963
2	Camshift + Kalman Filter		53	32%	0,1743
Dengan super-resolusi					
1	Camshift	167	163	98%	0,3047
2	Camshift + Kalman Filter		166	99%	0,2732

Tabel 5.9 Hasil Pelacakan Objek Video soccer.avi

No	Metode	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)	Persentase Akurasi	Rata-rata Waktu Pelacakan per <i>frame</i> (detik)
Tanpa super-resolusi					
1	Camshift	76	56	74%	0,2717
2	Camshift + Kalman Filter		58	76%	0,2280
Dengan super-resolusi					
1	Camshift	76	67	88%	0,2742
2	Camshift + Kalman Filter		69	91%	0,2567

Tabel 5.10 Hasil Pelacakan Objek Video ball.avi

No	Metode	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)	Persentase Akurasi	Rata-rata Waktu Pelacakan per <i>frame</i> (detik)
Tanpa super-resolusi					
1	Camshift	54	54	100%	0,3349
2	Camshift + Kalman Filter		54	100%	0,2989
Dengan super-resolusi					
1	Camshift	54	54	100%	1,0147
2	Camshift + Kalman Filter		54	100%	0,9236

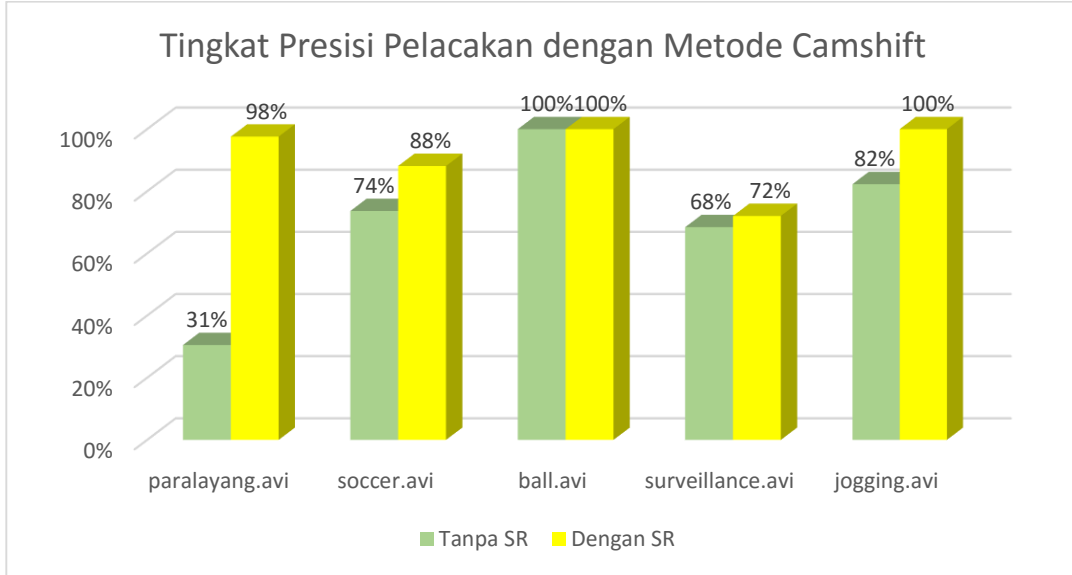
Tabel 5.11 Hasil Pelacakan Objek Video surveillance.avi

No	Metode	Jumlah Frame	Hasil Pelacakan (frame)	Persentase Akurasi	Rata-rata Waktu Pelacakan per frame (detik)
Tanpa super-resolusi					
1	Camshift	111	76	68%	0,2692
2	Camshift + Kalman Filter		76	68%	0,2531
Dengan super-resolusi					
1	Camshift	111	80	72%	0,1601
2	Camshift + Kalman Filter		82	74%	0,1404

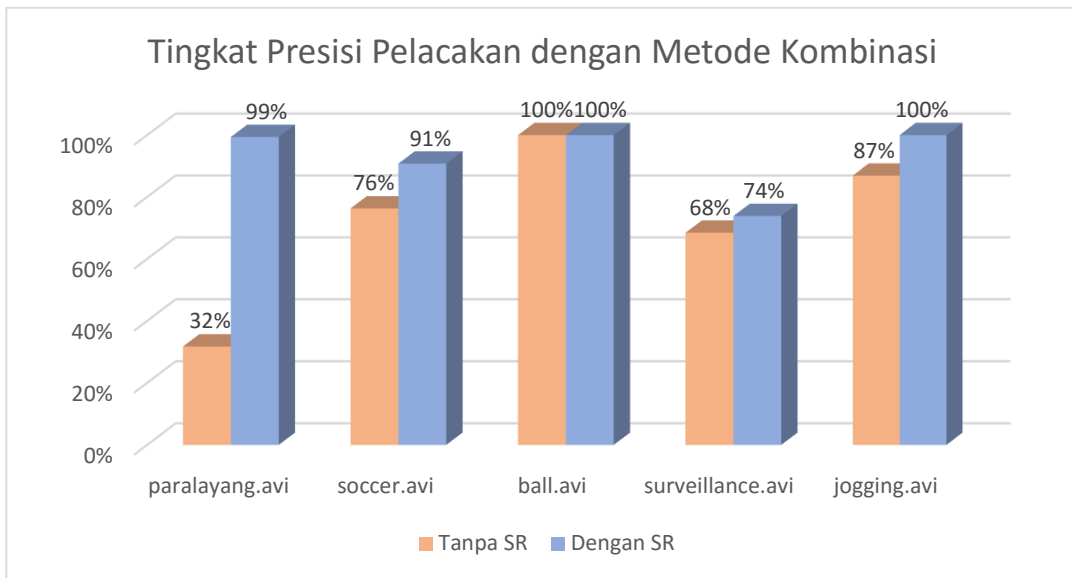
Tabel 5.12 Hasil Pelacakan Objek Video jogging.avi

No	Metode	Jumlah Frame	Hasil Pelacakan (frame)	Persentase Akurasi	Rata-rata Waktu Pelacakan per frame (detik)
Tanpa super-resolusi					
1	Camshift	237	195	82%	0,1308
2	Camshift + Kalman Filter		206	85%	0,1279
Dengan super-resolusi					
1	Camshift	237	237	100%	0,0711
2	Camshift + Kalman Filter		237	100%	0,0641

Hasil pelacakan dengan proses super-resolusi memberikan hasil pelacakan presisi sama atau lebih bagus pada setiap metode. Gambar 5.9 dan Gambar 5.10 menunjukkan peningkatan tingkat presisi pelacakan dengan metode Camshift dan metode kombinasi.



Gambar 5.9 Tingkat presisi pelacakan dengan metode Camshift.



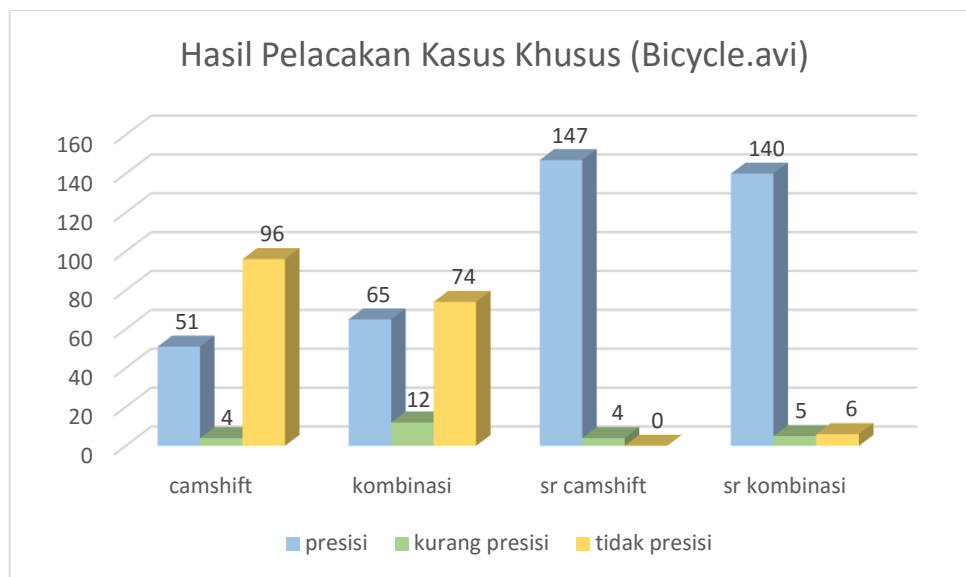
Gambar 5.10 Tingkat presisi pelacakan dengan metode kombinasi.

Tabel 5.13 menyajikan persentase akurasi secara menyeluruh video khusus dari kedua metode pelacakan, baik tanpa proses super-resolusi maupun dengan proses super-resolusi.

Tabel 5.13 Hasil Pelacakan Objek Video bicycle.avi

No	Metode	Jumlah Frame	Hasil Pelacakan (frame)	Persentase Akurasi	Rata-rata Waktu Pelacakan per frame (detik)
Tanpa super-resolusi					
1	Camshift	151	51	34%	0,0748
2	Camshift + Kalman Filter		65	43%	0,0658
Dengan super-resolusi					
1	Camshift	151	147	97%	0,1185
2	Camshift + Kalman Filter		140	93%	0,1172

Pada video khusus hasil pelacakan dengan proses super-resolusi memberikan hasil pelacakan presisi sama atau lebih bagus pada setiap metode. Akan tetapi, pelacakan super-resolusi menghasilkan penurunan presisi pelacakan sebesar 7 frame dengan metode kombinasi Camshift dan Kalman Filter. Gambar 5.11 menunjukkan peningkatan dan penurunan tingkat presisi pelacakan pada video bicycle.avi dengan metode Camshift dan metode kombinasi.

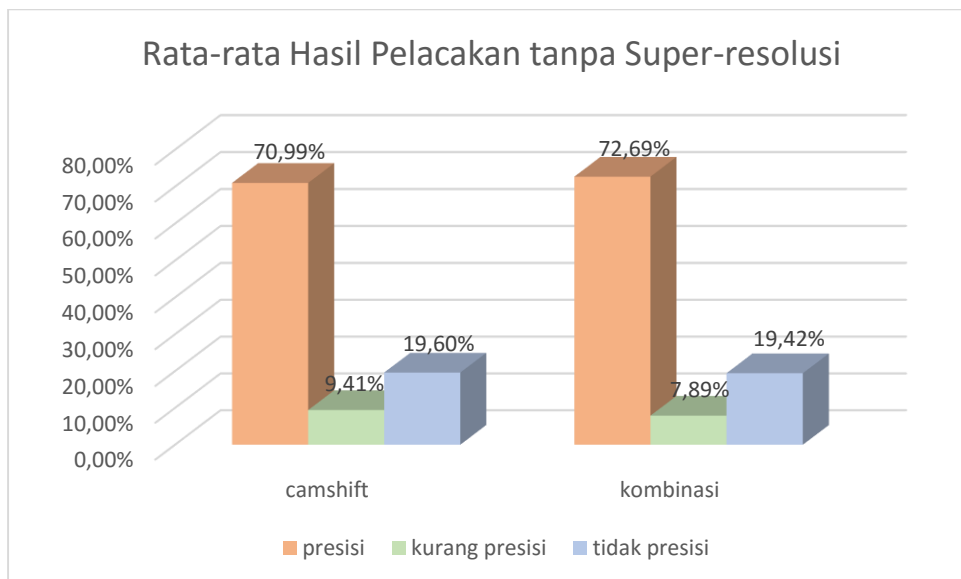


Gambar 5.11 Tingkat presisi pelacakan objek pada kasus khusus.

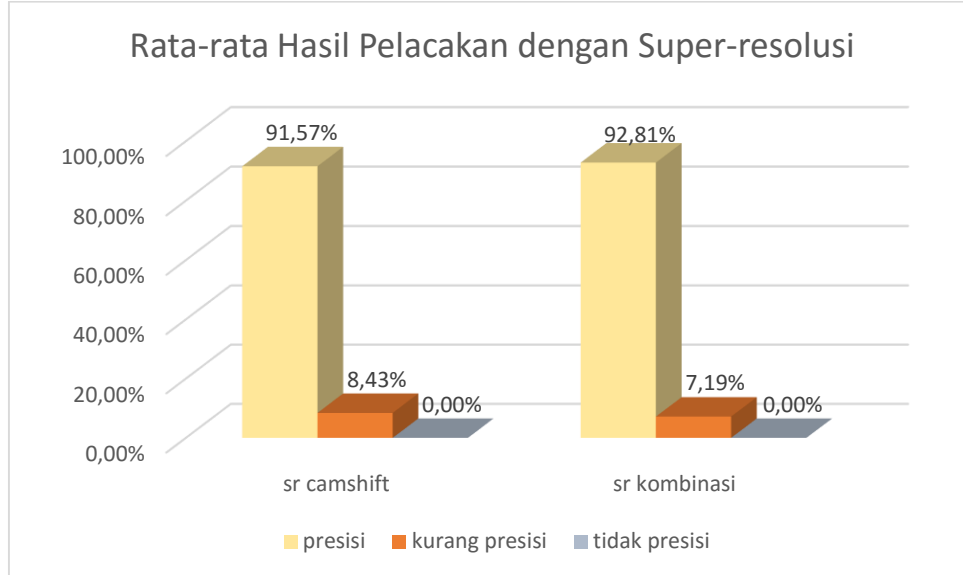
Gambar 5.11 menunjukkan pelacakan video khusus dengan metode Camshift lebih unggul 7 frame daripada metode kombinasi saat pelacakan dengan proses super-

resolusi. Video bicycle.avi memiliki karakteristik kamera bergerak dan perubahan bentuk objek. Sehingga kemungkinan prediksi yang salah semakin besar, karena metode Kalman Filter akan menghasilkan pelacakan optimal saat kamera tidak bergerak, objek tunggal, dan warna objek yang tidak menyerupai *background*.

Gambar 5.12 sampai Gambar 5.15 menunjukkan rata-rata persentase hasil pelacakan objek dari masing-masing metode dan dengan atau tanpa proses super-resolusi pada seluruh video tanpa video kasus khusus. Gambar 5.12 menunjukkan rata-rata persentase hasil pelacakan objek dengan metode kombinasi lebih unggul $\pm 1\%$ saat pelacakan tanpa proses super-resolusi. Gambar 5.13 menunjukkan rata-rata persentase hasil pelacakan objek dengan metode kombinasi lebih unggul $\pm 2\%$ saat pelacakan dengan proses super-resolusi. Hal tersebut menunjukkan dengan penambahan metode Kalman Filter membuat hasil pelacakan semakin presisi.



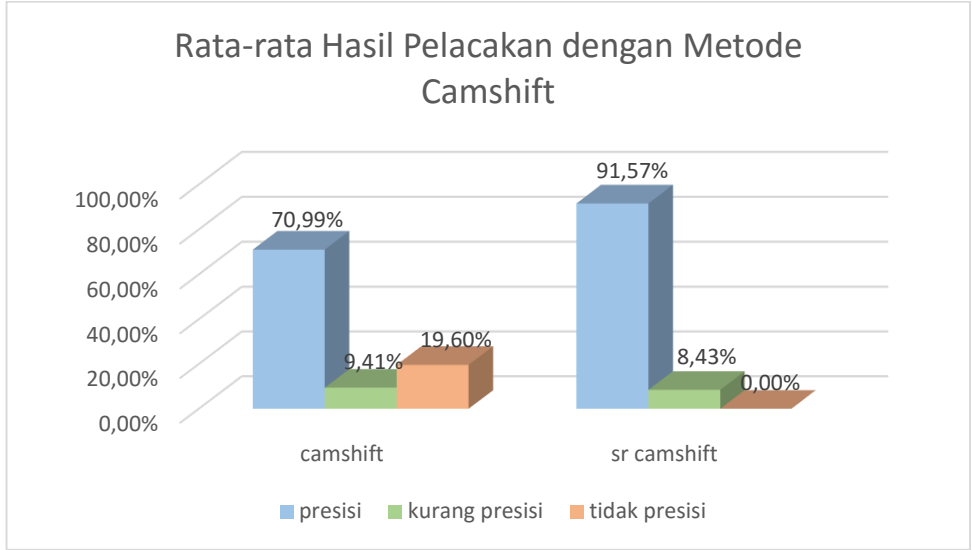
Gambar 5.12 Persentase rata-rata hasil pelacakan objek tanpa proses SR.



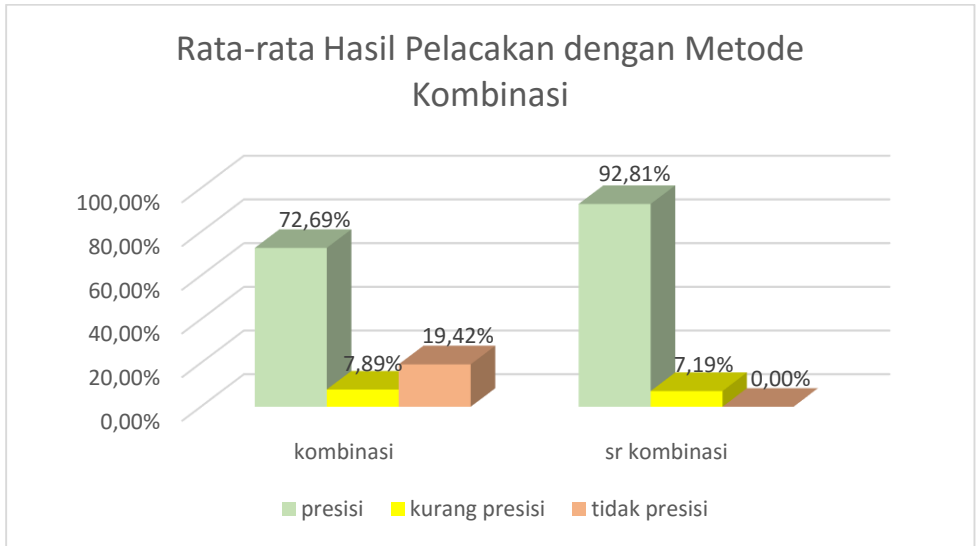
Gambar 5.13 Persentase rata-rata hasil pelacakan objek dengan proses SR.

Gambar 5.14 menunjukkan rata-rata persentase hasil pelacakan objek dengan metode Camshift yang didahului dengan proses super-resolusi memberikan hasil pelacakan $\pm 21\%$ lebih presisi. Sedangkan Gambar 5.15 menunjukkan rata-rata persentase hasil pelacakan objek dengan metode kombinasi yang didahului dengan proses super-resolusi memberikan hasil pelacakan $\pm 22\%$ lebih presisi. Hal tersebut menunjukkan dengan penambahan proses super-resolusi sebelum pelacakan objek membuat hasil pelacakan semakin presisi.

Pelacakan menggunakan metode kombinasi Camshift dan Kalman Filter secara logika akan membutuhkan waktu pelacakan yang cukup lama, karena penambahan langkah prediksi dengan metode Kalman Filter. Namun dalam penelitian yang telah dilakukan, hal tersebut yang terjadi sebaliknya, karena dengan adanya langkah prediksi dengan Kalman Filter membuat proses *looping* penghitungan *centroid* pada metode Camshift semakin singkat.



Gambar 5.14 Persentase rata-rata hasil pelacakan objek dengan metode Camshift.



Gambar 5.15 Persentase rata-rata hasil pelacakan objek dengan metode kombinasi.

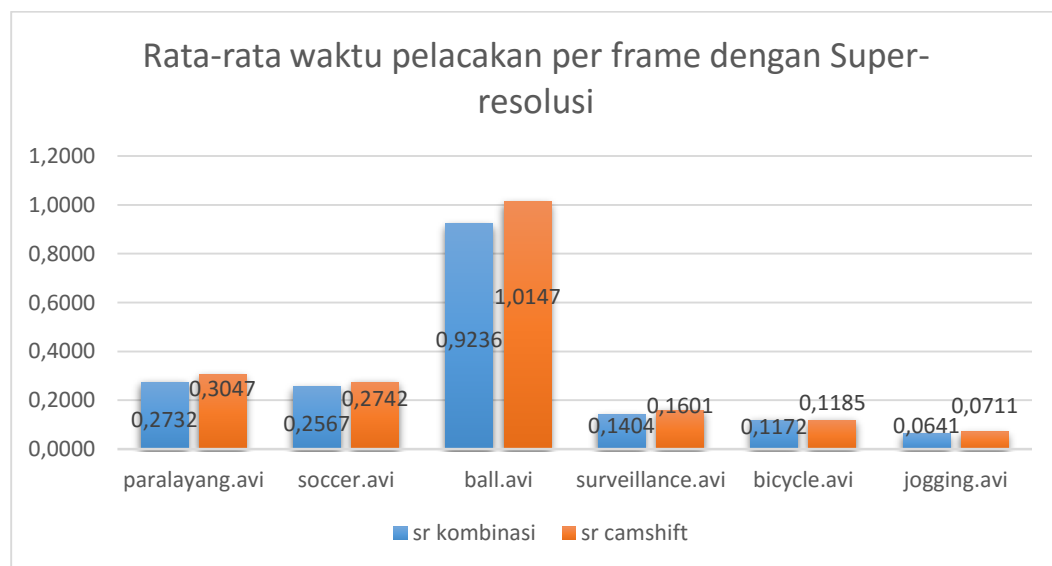
Penghitungan *centroid* metode Camshift akan berhenti saat selisih *search window frame i* dan *frame i - 1* konvergen. Pada penelitian ini, konvergen yang dimaksud adalah nilai selisih tersebut kurang dari ambang batas/ *threshold* yang telah ditentukan yaitu sebesar 1. *Threshold* bernilai 1 didapatkan dari hasil percobaan yang telah dilakukan dari ujicoba beberapa kali oleh peneliti. Tabel 5.14 menyajikan data

berapa kali proses *looping* untuk menentukan *centroid* dalam proses pelacakan objek. Contoh hasil penghitungan *centroid* pelacakan dapat dilihat pada Lampiran 4 dan 5.

Tabel 5.14 Proses *Looping* Menentukan *Centroid* saat Proses Pelacakan Objek

No.	Nama	<i>Looping centroid</i>			
		Camshift	Kombinasi	SR Camshift	SR Kombinasi
1	paralayang.avi	211	188	318	281
2	soccer.avi	134	106	198	181
3	ball.avi	109	101	131	121
4	surveillance.avi	148	133	188	153
Kasus khusus					
1	bicycle.avi	237	203	375	306

Dari tabel 5.4 sampai tabel 5.7 dapat dilihat jika rata-rata waktu pelacakan per *frame* di setiap video dengan metode kombinasi Camshift dan Kalman Filter lebih cepat daripada pelacakan dengan metode Camshift. Penambahan metode Kalman Filter membuat proses pelacakan dengan metode Camshift menjadi lebih efisien. Gambar 5.16 menunjukkan rata-rata waktu pelacakan per *frame* di setiap video dengan proses super-resolusi.

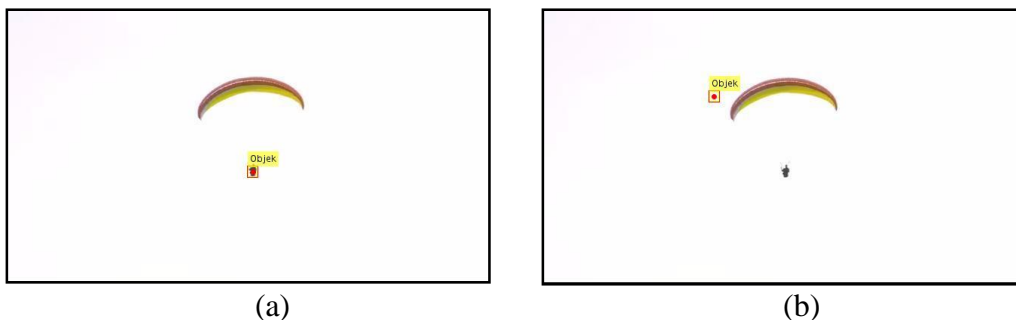


Gambar 5.16 Rata-rata waktu pelacakan objek per *frame* dengan proses super-resolusi (dalam detik).

Terdapat beberapa faktor yang dapat mempengaruhi kurang presisi tingkat pelacakan objek diantaranya sebagai berikut:

1. Pergerakan objek

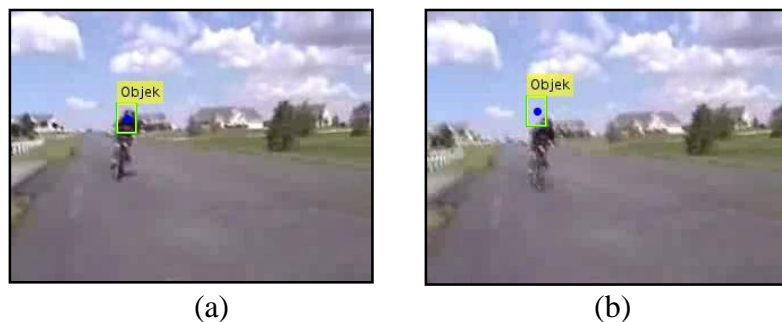
Kecepatan pergerakan objek terkadang membuat hasil penghitungan *centroid* pada *frame i* untuk *input frame i + 1* menjadi kurang atau lebih. Hasil pelacakan yang kurang tepat disajikan pada Gambar 5.17.



Gambar 5.17 Faktor kecepatan pergerakan objek dari (a) *frame 52* ke (b) *frame 53*.

2. Pergerakan kamera

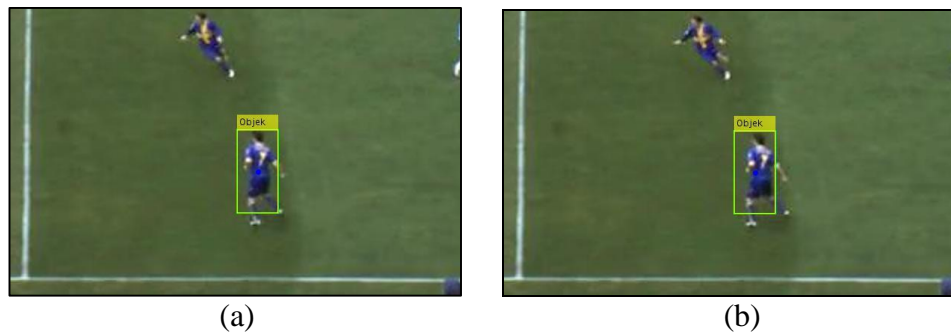
Pergerakan kamera membuat objek terkadang terletak pada koordinat yang sama saat *frame i* sampai *i + 1*. Faktor pergerakan kamera juga dapat merubah koordinat prediksi letak objek pada *frame* selanjutnya. Kejadian tersebut membuat penghitungan pada metode menjadi kurang tepat. Hasil pelacakan yang kurang tepat disajikan pada Gambar 5.18.



Gambar 5.18 Faktor pergerakan kamera dari (a) *frame 116* ke (b) *frame 117*.

3. Perubahan bentuk objek

Perubahan bentuk objek yang dilacak dapat membuat peletakan *rectangle* tidak presisi. Hal ini terjadi karena area penghitungan saat *frame i* yang digunakan sebagai acuan untuk *frame i + 1* berbeda bentuk objeknya. Hasil pelacakan yang kurang presisi disajikan pada Gambar 5.19.



Gambar 5.19 Faktor perubahan bentuk objek dari (a) *frame 58* ke (b) *frame 59*.

4. Kondisi lingkungan

Kondisi lingkungan video berupa pantulan cahaya atau waktu (pagi, siang, malam) sangat berpengaruh pada proses pelacakan objek. Misalkan warna objek yang akan dilacak menyerupai warna *background* dari video. Hasil pelacakan yang kurang presisi karea faktor kondisi lingkungan disajikan pada Gambar 5.20.





Gambar 5.20 Faktor kondisi lingkungan dari (a) *frame 76* ke (b) *frame 77*.

5.5 Uji Coba Pelacakang Berdurasi Panjang

Proses uji coba pelacakan video berdurasi panjang dengan cara memasukkan proses super-resolusi ke proses pelacakan. Kedua proses tersebut dikerjakan berurutan pada setiap *frame* di suatu video. Sehingga *frame* akan disuper-resolusi *single frame* terlebih dahulu, selanjutnya citra hasil super-resolusi akan dilacak dengan metode kombinasi Camshift dan Kalman Filter.

Uji coba ini bertujuan untuk mengetahui total waktu pelacakan apabila proses super-resolusi dilakukan sebelum proses pelacakan secara berurutan pada setiap *frame* dari suatu video. Video yang digunakan untuk uji coba terdapat 2 video hasil akuisisi di lingkungan kampus ITS. Data selengkapnya dapat dilihat pada Tabel 5.15.

Tabel 5.15 *Dataset* Video Uji Coba Berdurasi Panjang

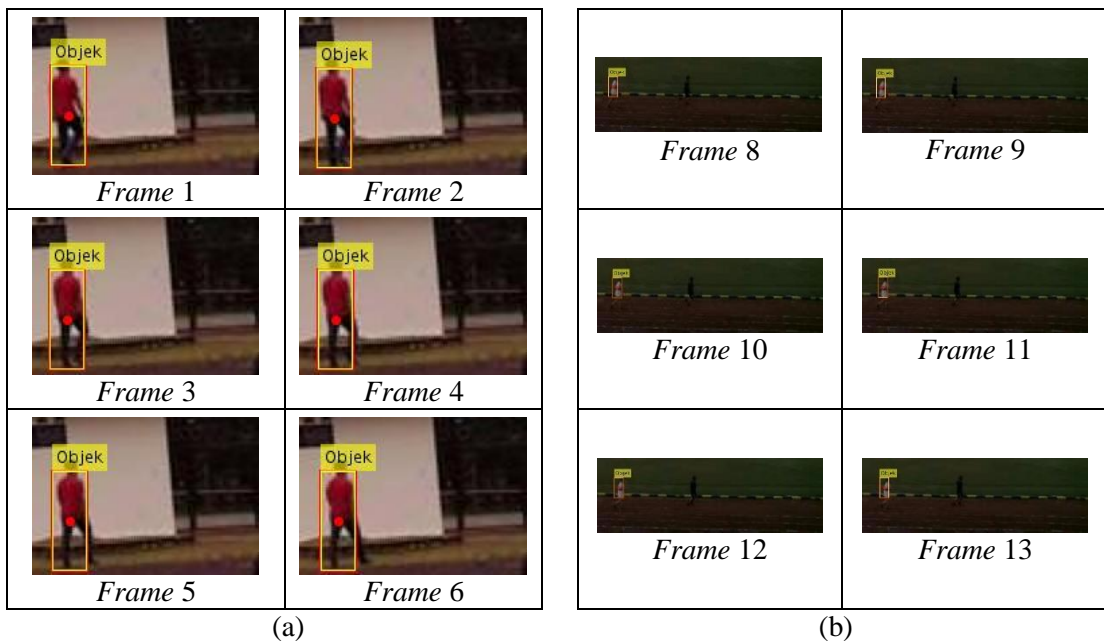
No	Nama	Jumlah Frame	Screenshot Video
1	orang.avi	210	
2	lari.avi	180	

Tabel 5.15 menyajikan *dataset* video uji coba berdurasi panjang dengan ukuran video 480×270 piksel untuk video orang.avi dan 640×360 piksel untuk video lari.avi. Dengan *fps* sebesar 24, durasi video orang.avi sebesar 00:00:08 detik dan durasi video lari.avi sebesar 00:00:08 detik.

Tabel 5.16 Hasil Pelacakan Video Berdurasi Panjang dengan Metode Kombinasi

No	Nama	Jumlah <i>Frame</i>	Hasil Pelacakan (<i>frame</i>)			Total Waktu Pelacakan (detik)
			Presisi	Kurang Presisi	Tidak Presisi	
1	orang.avi	210	210	-	-	108,72
2	lari.avi	180	180	-	-	1046,45

Tabel 5.16 menyajikan total waktu pelacakan setiap video semakin lama dari waktu asli video karena proses super-resolusi citra membutuhkan waktu komputasi yang cukup lama. Rata-rata waktu pelacakan per *frame* video orang.avi sebesar 0,5436 detik. Sedangkan video lari.avi sebesar 5,81361 detik. Hal tersebut menandakan apabila nantinya proses pelacakan akan dilakukan secara *real time*, maka akan terjadi *delay* atau penundaan waktu. Gambar 5.21 menyajikan beberapa urutan *frame* hasil pelacakan secara *real time* dengan metode kombinasi.



Gambar 5.21 Potongan *frame* hasil pelacakan video berdurasi panjang (a) orang.avi dan (b) lari.avi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang diperoleh berdasarkan penelitian yang telah dilaksanakan. Disamping itu, pada bab ini juga diberikan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan eksperimen dan pembahasan terhadap hasil pengujian yang telah dilakukan terhadap perangkat lunak untuk pelacakan objek berbasis citra super-resolusi menggunakan metode Camshift dan metode kombinasi Camshift dan Kalman Filter, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil menerapkan metode Camshift dan Kalman Filter untuk melakukan pelacakan objek pada beberapa video berbasis citra non super-resolusi maupun berbasis citra super-resolusi. Pelacakan dilakukan dengan beberapa tahapan: super-resolusi citra, pemilihan ROI, pelacakan Camshift, dan prediksi objek dengan Kalman Filter.
2. Penambahan proses super-resolusi dapat membuat hasil pelacakan dengan metode Camshift lebih presisi dengan kenaikan persentase tertinggi sebesar 67% pada video *paralayang.avi*. Sedangkan hasil pelacakan dengan metode kombinasi kenaikan persentase tertinggi juga sebesar 67% pada video *paralayang.avi*. Sedangkan penambahan proses super-resolusi tidak berarti pada video *ball.avi*. Video tersebut memiliki ciri-ciri warna *background* dan objek berbeda dan kamera tidak bergerak.
3. Penelitian ini berhasil mengkombinasikan metode Camshift dan Kalman Filter untuk pelacakan objek. Pelacakan dengan metode Camshift menghasilkan rata-rata pelacakan presisi keseluruhan video sebesar 71% tanpa proses super-resolusi dan 92% dengan super-resolusi. Pelacakan dengan metode kombinasi Camshift dan Kalman Filter menghasilkan rata-rata pelacakan presisi keseluruhan video sebesar

73% tanpa proses super-resolusi dan 93% dengan super-resolusi. Sehingga penambahan proses super-resolusi dan Kalman Filter membuat hasil pelacakan objek menjadi semakin presisi.

4. Penambahan metode Kalman Filter pada proses pelacakan membuat waktu pelacakan pada setiap frame dengan metode kombinasi lebih cepat daripada pelacakan dengan metode Camshift. Kenaikan rata-rata waktu pelacakan per frame tertinggi sebesar 0,09102 detik dengan proses super-resolusi pada video ball.avi.
5. Uji coba pelacakan video berdurasi panjang menghasilkan *delay* atau penundaan waktu hasil pelacakan. Rata-rata waktu pelacakan per frame video orang.avi sebesar 0,5436 detik. Sedangkan video lari.avi sebesar 5,81361 detik. Proses uji coba tersebut dilakukan dengan cara memasukkan proses super-resolusi ke proses pelacakan.

6.2 Saran

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang disarankan untuk perbaikan kinerja dan pengembangan selanjutnya yaitu:

1. Program belum dapat melakukan pelacakan terhadap benda atau objek yang memiliki warna hampir sama dengan warna *background*. Penelitian berikutnya diharapkan untuk mengkhususkan pada objek yang memiliki karakteristik tersebut.
2. Objek penelitian berupa objek tunggal yang tidak oklusi dan ukuran video yang kurang dari 1000×1000 piksel. Pada penelitian selanjutnya diharapkan objek dapat berupa *multiple* objek dan dapat ditambahkan kasus oklusi, serta dengan ukuran video yang lebih besar.
3. Pada penelitian berikutnya, proses pelacakan dapat diterapkan pada sistem *real time* dengan menggunakan metode super-resolusi yang lebih cepat waktu komputasinya. Syarat lainnya adalah metode pelacakan harus melacak dari frame ke-1. Sehingga dari segi proses super-resolusi dan proses pelacakan, masing-masing proses tidak akan terdapat *delay* saat *running* secara *real time*.

4. Penghitungan pelacakan objek masih dilakukan secara manual. Pada penelitian selanjutnya bisa dilakukan dengan membandingkan nilai sebenarnya yang dapat diperoleh dari penyedia *dataset* video.
5. Program belum terhubung dengan *database*. Sehingga data penghitungan hanya disimpan sementara oleh memori CPU. Pada penelitian berikutnya program dapat dihubungkan dengan *database*. Sehingga data dapat disimpan dan diolah untuk kepentingan lebih lanjut.

DAFTAR PUSTAKA

- [1] Chen, X., Wu, H., Li, X., Luo, X., dan Qiu, T., (2012), “Real-time Visual Object Tracking via Camshift-Based Robust Framework”, *International Journal of Fuzzy Systems*, Vol. 14, No. 2.
- [2] Collins, R.T, Zhou, X., dan Teh, S.K., (2005), “An Open Source Tracking Testbed and Evaluation Web Site”, *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*.
- [3] Daviesy, D., Palmery, P. dan Mirmehdiz, M., (1998), “Detection and Tracking of Very Small Low Contrast Objects”, *Proceedings of the British Machine Vision Conference*, Eds: Nixon, M. et al., BMVA Press, hal. 60.1-60.10.
- [4] Deori, B. dan Thounaojam, D. M., (2014), “A Survey on Moving Object Tracking in Video”, *International Journal on Information Theory (IJIT)*, Vol. 3, No. 3, hal 31-46.
- [5] Fisher, R.B., (2004), “Surveillance Ground Truth Data Set” , *Proceedings Sixth IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, hal 1-5.
- [6] Hsia, K., Lien, S., dan Su, J., (2013), “Moving Target Tracking Based on Camshift Approach and Kalman Filter”, *An International Journal Applied Mathematics & Information Sciences*, Vol. 7, No. 1, hal. 193-200.
- [7] Huang, S. dan Hong, J., (2011), “Moving Object Tracking System Based on Camshift and Kalman Filter”, *IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet)*, hal 1423-1426.

- [8] Jing, L., Zongliang, G. dan Xiuchang, Z., (2013), "Directional Bicubic Interpolation-A New Method of Image Super-Resolution", *Proceedings of International Conference on Multimedia Technology*, Atlantis Press, hal. 470-477.
- [9] Lefevre, S., Foissotte, T. dan Vincent, N., (2003), "Detection And Tracking of Small Objects by Details Removal", *Proceedings of 4th European Workshop on Image Analysis for Multimedia Interactive Services*, University of London, Queen Mary, hal 137-140.
- [10] Mise, O. dan Breckonb, T.P., (2013), "Super-Resolution Imaging Applied to Moving Targets in High Dynamic Scenes", *International Society for Optics and Photonics (SPIE Security+ Defense)*.
- [11] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C., Lee, J.T., Mukherjee, S., Aggarwal, J.K., Lee, H., Davis, L, Swears, E., Wang, X., Ji, Q., Reddy, K., Shah, M, Vondrick, C., Pirsiavash, H., Ramanan, D., Yuen, J., Torralba, A., Song, B., Fong, A., Roy-Chowdhury, A., dan Desai, M., (2011), "A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video", *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, hal 3153-3160.
- [12] Patel, H.A. dan Thakore, D.G., (2013), "Moving Object Tracking Using Kalman Filter", *International Journal of Computer Science and Mobile Computing*, Vol. 2, No. 4, hal. 326-332.
- [13] Raval, H. dan Kothari, A., (2014), "Analysis of Single Frame Super Resolution Methods", *International Journal Of Engineering And Computer Science*, Vol. 3, No. 11, hal 9073-9076.

- [14] Yilmaz, A., Javed, O., dan Shah, M., (2006), “Object tracking: a survey”, *ACM Comput. Surv.* 38, (4), Article 13.

BIODATA PENULIS



Penulis yang bernama lengkap Galandaru Swalaganata dilahirkan di Tulungagung pada tanggal 28 Oktober 1991 bertepatan dengan hari sumpah pemuda. Anak pertama dari dua bersaudara dari orangtua Agus Sunaryo dan Insiyatun. Riwayat pendidikan dimulai dari SD Negeri 1 Bendiljati Wetan dan tamat tahun 2004. Selanjutnya melanjutkan ke SMP Negeri 1 Ngunut tamat tahun 2007 dan masuk SMA Negeri 1 Boyolangu tamat tahun 2010. Setelah itu penulis melanjutkan ke Perguruan Tinggi Negeri jurusan Matematika angkatan 2010 di Universitas Negeri Malang melalui jalur PMDK. Selama kuliah di UM pernah menjadi asisten dosen pada matakuliah Desain Web dan Perencanaan Sistem Informasi. Penulis juga pernah aktif dalam organisasi Himpunan Mahasiswa Jurusan Matematika. Penulis lulus sarjana tepat delapan semester dengan mendapat gelar Sarjana Sains. Penulis melanjutkan studi S2 di Jurusan Matematika Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2014 dan mengambil Bidang Minat Ilmu Komputer. Selama kuliah di ITS penulis mengikuti penelitian yang dilakukan oleh dosen untuk menambah pengalaman. Untuk membentuk jaringan atau membutuhkan informasi yang berhubungan dengan tesis ini atau erat kaitannya dengan matematika, penulis dapat dihubungi melalui *e-mail* galandaru.swalaganata@gmail.com.

LAMPIRAN

Lampiran 1. Kode program super-resolusi interpolasi Directional Bicubic

```
%langkah 1
Isr = cast(zeros(jml_piksel-1,jml_piksel2-1,d), 'double');
Ib = imresize(I,2,'bicubic'); %langkah 2
Ib1 = histeq(Ib(:,:,1)); %%%ekualisasi histogram
Ib2 = histeq(Ib(:,:,2)); %%%ekualisasi histogram
Ib3 = histeq(Ib(:,:,3)); %%%ekualisasi histogram
for i=1:jml_piksel
    for j=1:jml_piksel2
        Isr(zoom*i-1, zoom*j-1, 1) = I(i, j, 1);
        Isr(zoom*i-1, zoom*j-1, 2) = I(i, j, 2);
        Isr(zoom*i-1, zoom*j-1, 3) = I(i, j, 3);
    end
end
Ihr = Isr;

%langkah 3
index = 1;
i = jml_piksel;
j = jml_piksel2;
T = 3;
mulai = jml_piksel - ((2*(jml_piksel-2))+1);
mulai2 = jml_piksel2 - ((2*(jml_piksel2-2))+1);

for m=mulai:2:jml_piksel
    for n = mulai2:2:jml_piksel2
        G1(index) = abs(Isr(i+m, j-n) - Isr(i+m-2, j-n+2));
        G2(index) = abs(Isr(i+m, j+n) - Isr(i+m-2, j+n-2));
        index = index+1;
    end
end

(1 + max(G1))/(1 + max(G2));
for k = 1 : jml_piksel-1
    for l = 1 : jml_piksel2-1
        if ((1 + max(G1))/(1 + max(G2))) > T
            p1 = Ib(2*k, 2*l, 1);
            p2 = Ib(2*k, 2*l, 2);
            p3 = Ib(2*k, 2*l, 3);
        elseif ((1 + max(G1))/(1 + max(G2))) < T
            p1 = Ib(2*k, 2*l, 1);
            p2 = Ib(2*k, 2*l, 2);
            p3 = Ib(2*k, 2*l, 3);
        else
            w1 = 1/(1+max(G1));
            w2 = 1/(1+max(G2));
            p1 = (w1*Ib(2*k, 2*l, 1)+w2*Ib(2*k, 2*l, 1))/(w1+w2);
        end
    end
end
```

```

        p2 = (w1*Ib(2*k,2*1,2)+w2*Ib(2*k,2*1,2))/(w1+w2);
        p3 = (w1*Ib(2*k,2*1,3)+w2*Ib(2*k,2*1,3))/(w1+w2);
    end
    Isr(2*k,2*1,1) = p1;
    Isr(2*k,2*1,2) = p2;
    Isr(2*k,2*1,3) = p3;
    l=l+1;
end
        k=k+1;
end

%langkah 4
index = 1;
i = jml_piksel;
j = jml_piksel2;
for m=mulai:2:-mulai+1
    for n = 0:2:jml_piksel2-1
        G11(index) = abs(Isr(i+m, j-n) - Isr(i+m-2, j-n+2));
        for mm=-2:2:jml_piksel-1
            G11(index) = G11(index) + abs(Isr(i+mm, j-1) -
Isr(i+mm, j+1));
        end
        G21(index) = abs(Isr(i-m, j+n) - Isr(i-m-2, j+n-2));
        for nn=-2:2:jml_piksel-1
            G21(index) = G21(index) + abs(Isr(i-1, j+nn) -
Isr(i+1, j+nn));
        end
        index = index+1;
    end
end

(1 + max(G11))/(1 + max(G21));
for k = 1 : jml_piksel
    for l = 1 : jml_piksel2-1
        if ((1 + max(G11))/(1 + max(G21))) > T
            p1 = Ib(2*k-1,2*1,1);
            p2 = Ib(2*k-1,2*1,2);
            p3 = Ib(2*k-1,2*1,3);
        elseif ((1 + max(G11))/(1 + max(G21))) < T
            p1 = Ib(2*k-1,2*1,1);
            p2 = Ib(2*k-1,2*1,2);
            p3 = Ib(2*k-1,2*1,3);
        else
            w1 = 1/(1+max(G11));
            w2 = 1/(1+max(G21));
            p1 = (w1*Ib(2*k-1,2*1,1)+w2*Ib(2*k-
1,2*1,1))/(w1+w2);
            p2 = (w1*Ib(2*k-1,2*1,2)+w2*Ib(2*k-
1,2*1,2))/(w1+w2);
            p3 = (w1*Ib(2*k-1,2*1,3)+w2*Ib(2*k-
1,2*1,3))/(w1+w2);
        end
        Isr(2*k-1,2*1,1) = p1;

```

```

        Isr(2*k-1,2*1,2) = p2;
        Isr(2*k-1,2*1,3) = p3;
        l=l+1;
    end
        k=k+1;
end
index = 1;
i = jml_piksel;
j = jml_piksel2;
for m=mulai:2:-mulai+1
    for n = 0:2:jml_piksel2-1
        G12(index) = abs(Isr(i+m, j-n) - Isr(i+m-2, j-n+2));
        for mm=-2:2:2
            G12(index) = G12(index) + abs(Isr(i+mm, j-1) -
Isr(i+mm, j+1));
        end
        G22(index) = abs(Isr(i-m, j+n) - Isr(i-m-2, j+n-2));
        for nn=-2:2:2
            G22(index) = G22(index) + abs(Isr(i-1, j+nn) -
Isr(i+1, j+nn));
        end
        index = index+1;
    end
end

(1 + max(G12))/(1 + max(G22));
for k = 1 : jml_piksel-1
    for l = 1 : jml_piksel2
        if ((1 + max(G12))/(1 + max(G22))) > T
            p1 = Ib(2*k,2*1-1,1);
            p2 = Ib(2*k,2*1-1,2);
            p3 = Ib(2*k,2*1-1,3);
        elseif ((1 + max(G12))/(1 + max(G22))) < T
            p1 = Ib(2*k,2*1-1,1);
            p2 = Ib(2*k,2*1-1,2);
            p3 = Ib(2*k,2*1-1,3);
        else
            w1 = 1/(1+max(G12));
            w2 = 1/(1+max(G22));
            p1 = (w1*Ib(2*k,2*1-1,1)+w2*Ib(2*k,2*1-
1,1))/(w1+w2);
            p2 = (w1*Ib(2*k,2*1-1,2)+w2*Ib(2*k,2*1-
1,2))/(w1+w2);
            p3 = (w1*Ib(2*k,2*1-1,3)+w2*Ib(2*k,2*1-
1,3))/(w1+w2);
        end
        Isr(2*k,2*1-1,1) = p1;
        Isr(2*k,2*1-1,2) = p2;
        Isr(2*k,2*1-1,3) = p3;
        l=l+1;
    end
        k=k+1;
end
end

```

```
toc;
[a b] = size(Isr);
Ie(:, :, 1) = Ib1;
Ie(:, :, 2) = Ib2;
Ie(:, :, 3) = Ib3;

Isr=cast(Ib, 'uint8'); %biasa
file=strcat('sr', num2str(frame), '.jpg');
workingDir = name;
fulln = fullfile(workingDir, file);
imwrite(Isr, fulln);

Isrh=cast(Ie, 'uint8'); %hasil ekualisasi
fileh=strcat('sr-ekual', num2str(frame), '.jpg');
fullnh = fullfile(workingDir, fileh);
imwrite(Isrh, fullnh);
```

Lampiran 2. Kode program pelacakan objek dengan metode Camshift

```
for frame = 1:nframe
    tic;
    filename = sprintf('sr%1.1i.jpg', frame);
    R = imread([imgDir '\' filename]);
    % RGB ke HSV
    I = rgb2hsv(R);
    % Ekstrak information HSV
    I = I(:,:,g);
    oldCamL = [0 0];
    MeanConverging = 1;
    oldL1 = L;

    while MeanConverging,
        % Compute centroid of search window
        M00 = 0.0;
        for i = L(1)-P : (L(1)+W(1)+P),
            for j = L(2)-P : (L(2)+W(2)+P),
                if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                    continue;
                end
                if isnan(i) == 1 || isnan(j) == 1
                    M00 = 1;
                else
                    M00 = M00 + double(I(round(i),round(j)));
                end
            end
        end
        M10 = 0.0;
        for i = L(1)-P : (L(1)+W(1)+P),
            for j = L(2)-P : (L(2)+W(2)+P),
                if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                    continue;
                end
                if isnan(i) == 1 || isnan(j) == 1
                    M10 = 10000;
                else
                    M10 = M10 + i * double(I(round(i),round(j)));
                end
            end
        end
        M01 = 0.0;
        for i = L(1)-P : (L(1)+W(1)+P),
            for j = L(2)-P : (L(2)+W(2)+P),
                if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                    continue;
                end
                if isnan(i) == 1 || isnan(j) == 1
                    M01 = 10000;
                else
```

```

                M01 = M01 + j * double(I(round(i),round(j)));
            end
        end
    end

    xc = round(M10 / M00);
    yc = round(M01 / M00);

    oldL = L;
    L = [floor(xc - (W(1)/2)) floor(yc - (W(2)/2))];
    count = count + 1;
    % Check threshold
    if xc == 10000 || yc ==10000
        L = [oldL(1)-100 oldL(2)-100];
        xc = L(1) + double((W(1))/2);
        yc = L(2) + double((W(2))/2);
        MeanConverging = 0;
    end
    if abs(oldL(1)-L(1)) < T || abs(oldL(2)-L(2)) < T
        MeanConverging = 0;
    end
end
centroidy(frame)= yc;
disp(sprintf('%3i:  %3i, %3i', frame, xc, yc));
Xpoints = [Xpoints xc];
Ypoints = [Ypoints yc];
R = insertShape(R, 'rectangle', [L(2) L(1) W(2)
W(1)], 'Color', 'green', 'lineWidth', 2);
R = insertShape(R, 'circle', [yc xc
1], 'Color', 'blue', 'lineWidth', 4);
R = insertObjectAnnotation(R, 'rectangle', [L(2) L(1) W(2)
W(1)], 'Objek');
file=strcat(num2str(frame), '.jpg');
workingDir = name;
fulln = fullfile(workingDir, file);
imwrite(R, fulln);
frame2 = imread([name num2str(frame) '.jpg']);
writeVideo(writerObj, frame2);
toc;
t1(frame)=toc;
waitbar(frame / nframe)
end
close(writerObj);
close(h)
round(centroidx)
round(centroidy)
count
time = sum(t1)
set(handles.time, 'String', time);
fullname = fullfile(name, 'tr-para-sr-cam.avi');
vid = vision.VideoFileReader(fullname);
i=1;
while ~isDone(vid)

```

```
frame = step(vid);  
axes(handles.axes3);  
imshow(frame);  
pause(0.000000000000001)  
i=i+1;  
end
```


Lampiran 3. Kode program pelacakan objek dengan metode kombinasi Camshift dan Kalman filter

```

%%%%%%%%%%%%%% inisialisasi kalman
centroidx = zeros(nframe,1);
centroidy = zeros(nframe,1);
predicted = zeros(nframe,4);
actual = zeros(nframe,4);

RR=[[1,0]', [0,1]'];
H=[[1,0]', [0,1]', [0,0]', [0,0]'];
Q=1*eye(4);
PP = 100*eye(4);
dt=1;
A=[[1,0,0,0]', [0,1,0,0]', [dt,0,1,0]', [0,dt,0,1]'];
%A=[[1,0,1,0]', [0,1,0,1]', [0,0,1,0]', [0,0,0,1]'];%sesuai referensi
tapi malah erornya tinggi (pengaruhnya nanti di prediksi)
kfinit = 0;
th = 50;
%%%%%%%%%%%%%%

name = 'E:\File S2\GUI-Tesis\Video fix\tr-lagi\';
E=mkdir(name);

h = waitbar(0, 'Ditunggu Yaaa...')
steps = 1000;
writerObj = VideoWriter('tr-surv-sr-cam');
writerObj.FrameRate = 24;
open(writerObj);
for frame = 1:nframe
    tic;
    filename = sprintf('sr%1.1i.jpg', frame);
    R = imread([imgDir '\' filename]);
    % RGB ke HSV
    I = rgb2hsv(R);
    % Extrak information HSV
    I = I(:, :, g);
    oldCamL = [0 0];
    MeanConverging = 1;
    oldL1 = L;

    while MeanConverging,
        % Compute centroid of search window
        M00 = 0.0;
        for i = L(1)-P : (L(1)+W(1)+P),
            for j = L(2)-P : (L(2)+W(2)+P),
                if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
                    continue;
                end
                if isnan(i) == 1 || isnan(j) == 1
                    M00 = 1;
                else

```

```

        M00 = M00 + double(I(round(i),round(j)));
    end
end
end

M10 = 0.0;
for i = L(1)-P : (L(1)+W(1)+P),
    for j = L(2)-P : (L(2)+W(2)+P),
        if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
            continue;
        end
        if isnan(i) == 1 || isnan(j) == 1
            M10 = 10000;
        else
            M10 = M10 + i * double(I(round(i),round(j)));
        end
    end
end

M01 = 0.0;
for i = L(1)-P : (L(1)+W(1)+P),
    for j = L(2)-P : (L(2)+W(2)+P),
        if i > size(I,1) || j > size(I,2) || i < 1 || j < 1
            continue;
        end
        if isnan(i) == 1 || isnan(j) == 1
            M01 = 10000;
        else
            M01 = M01 + j * double(I(round(i),round(j)));
        end
    end
end

xc = round(M10 / M00);
yc = round(M01 / M00);

oldL = L;
L = [floor(xc - (W(1)/2)) floor(yc - (W(2)/2))];
count = count + 1;
% Check threshold
if xc == 10000 || yc == 10000
    L = [oldL(1)-100 oldL(2)-100];
    xc = L(1) + double((W(1))/2);
    yc = L(2) + double((W(2))/2);
    MeanConverging = 0;
end
if abs(oldL(1)-L(1)) < T || abs(oldL(2)-L(2)) < T
    MeanConverging = 0;
end
end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% kalman dimulai
xcorner = L(2);
ycorner = L(1);
xwidth = W(2);
ywidth = W(1);
centroidx(frame)= xc;
centroidy(frame)= yc;
kalmanx = centroidx(frame)- xcorner;
kalmany = centroidy(frame)- ycorner;

if kfinit == 0
    predicted =[centroidx(frame),centroidy(frame),0,0]' ;
else
    predicted = A*actual(frame-1,:)' ; %persamaan prediksi 1
end

kfinit = 1;
Ppre = A*PP*A' + Q; %persamaan prediksi 2

K = Ppre*H'/(H*Ppre*H'+RR); %persamaan kalman-gain
actual(frame,:) = (predicted +
K*([centroidx(frame),centroidy(frame)]' - H*predicted))'; %update
persamaan 1
PP = (eye(4)-K*H)*Ppre; %update persamaan 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp(sprintf('%3i:  %3i, %3i', frame, xc, yc));
Xpoints = [Xpoints xc];
Ypoints = [Ypoints yc];
R = insertShape(R, 'rectangle', [L(2) L(1) W(2)
W(1)], 'Color', 'red', 'lineWidth', 2);
R = insertShape(R, 'circle', [centroidy(frame) centroidx(frame)
1], 'Color', 'red', 'lineWidth', 4);
R = insertObjectAnnotation(R, 'rectangle', [L(2) L(1) W(2)
W(1)], 'Objek');
L = [actual(frame,2)-kalmany actual(frame,1)-kalmanx];
W = [ywidth xwidth];
file=strcat(num2str(frame), '.jpg');
workingDir = name;
fulln = fullfile(workingDir,file);
imwrite(R,fulln);
frame2 = imread([name num2str(frame) '.jpg']);
writeVideo(writerObj,frame2);
toc;
t1(frame)=toc;
waitbar(frame / nframe)
end
close(writerObj);
close(h)
round(centroidx)
round(centroidy)
count

```

```
time = sum(t1)
set(handles.time, 'String', time);
fullname = fullfile(name, 'tr-para-sr-cam.avi');
vid = vision.VideoFileReader(fullname);
i=1;
while ~isDone(vid)
    frame = step(vid);
    axes(handles.axes3);
    imshow(frame);
    pause(0.000000000000001)
    i=i+1;
end
```

Lampiran 4. Centroid hasil pelacakan video soccer.avi

	x	y	perp x	perp y
		29		
Koordinat ROI :	469	4	28	53
hsv 1				
Camshift			Camshift+Kalman	
Time :	20,6456		Time :	17,3256
			Ekualisasi	
Ekualisasi :	no		:	no



M 134
Koordinat centroid

	x	y
1	321	483
2	321	483
3	322	483
4	322	482
5	322	481
6	323	479
7	324	478
8	324	478
9	325	478
10	326	477
11	328	476
12	330	475
13	331	475
14	332	475
15	333	475
16	334	475
17	335	474
18	336	474
19	337	474
20	338	474
21	339	474
22	341	474
23	342	474
24	344	475
25	346	476
26	347	476
27	351	479
28	353	480
29	355	481
30	355	482
31	355	483
32	361	486
33	365	489
34	369	492
35	370	494
36	370	495
37	370	496
38	370	498
39	370	500

M 106
Koordinat centroid

	x	y
1	321	483
2	321	483
3	322	483
4	322	482
5	322	481
6	323	480
7	324	479
8	324	478
9	325	478
10	325	477
11	326	477
12	327	476
13	329	475
14	330	474
15	331	474
16	332	474
17	333	474
18	334	474
19	335	474
20	337	474
21	339	474
22	341	474
23	342	474
24	343	475
25	344	476
26	345	476
27	347	477
28	353	480
29	354	480
30	355	481
31	356	483
32	357	484
33	358	486
34	367	492
35	367	492
36	368	494
37	368	496
38	368	498
39	368	500

Selisih

x	y
0	0
0	0
0	0
0	0
0	0
0	1
0	1
0	0
0	0
1	0
2	1
3	1
2	0
2	1
2	1
2	1
2	0
2	0
1	0
0	0
0	0
0	0
1	0
2	0
2	0
4	2
0	0
1	1
0	1
1	0
4	2
7	3
2	0
3	2
2	1
2	0
2	0
2	0

40	370	502
41	371	506
42	374	513
43	376	517
44	378	521
45	382	526
46	382	527
47	382	529
48	384	534
49	385	538
50	387	543
51	386	546
52	384	551
53	384	553
54	384	555
55	382	561
56	382	563
57	386	571
58	388	575
59	388	576
60	388	578
61	388	581
62	388	584
63	388	587
64	387	593
65	387	596
66	387	599
67	387	602
68	386	608
69	386	611
70	386	613
71	386	616
72	388	622
73	390	627
74	390	629
75	390	631
76	389	636

40	368	502
41	370	508
42	370	510
43	376	519
44	376	519
45	382	526
46	382	527
47	383	529
48	384	531
49	385	534
50	387	542
51	386	544
52	384	550
53	383	552
54	383	555
55	383	557
56	383	560
57	386	569
58	388	575
59	388	576
60	389	578
61	389	581
62	389	584
63	389	587
64	388	590
65	388	593
66	388	596
67	387	599
68	386	602
69	385	605
70	385	608
71	385	611
72	385	613
73	386	616
74	386	618
75	386	621
76	386	623

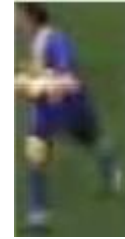
2	0
1	2
4	3
0	2
2	2
0	0
0	0
1	0
0	3
0	4
0	1
0	2
0	1
1	1
1	0
1	4
1	3
0	2
0	0
0	0
1	0
1	0
1	0
1	3
1	3
1	3
0	3
0	6
1	6
1	5
1	5
3	9
4	11
4	11
4	10
3	13

Lampiran 5. Centroid hasil pelacakan video soccer.avi dengan SR

Koordinat ROI : x y perp x perp y
 124 33 55 111

Camshift
 Time : 25,74882
 Ekualisasi : no
 20,8424
 M : 198

Camshift+Kalman
 Time : 25,85145
 Ekualisasi : no
 19,5092
 M : 181



Koordinat centroid

	x	y
1	89	151
2	89	150
3	96	145
4	92	138
5	92	137
6	93	135
7	93	134
8	95	133
9	98	131
10	100	130
11	103	129
12	110	126
13	114	124
14	116	124
15	117	124
16	119	123
17	119	123
18	120	123
19	121	123
20	125	124
21	130	126
22	133	126
23	137	128
24	145	132
25	147	133
26	151	134
27	154	138
28	158	141
29	166	144
30	166	145
31	167	148
32	178	154
33	181	159
34	185	166
35	185	168
36	185	170
37	185	173
38	185	176
39	185	179
40	186	185

Koordinat centroid

	x	y
1	89	151
2	89	150
3	96	145
4	92	138
5	92	138
6	92	137
7	92	136
8	93	135
9	94	134
10	95	133
11	98	132
12	110	126
13	112	124
14	114	124
15	115	124
16	116	124
17	117	124
18	118	124
19	120	124
20	123	124
21	130	126
22	132	126
23	134	127
24	143	132
25	146	133
26	148	134
27	149	136
28	157	142
29	159	142
30	160	144
31	161	147
32	177	155
33	181	160
34	185	166
35	185	168
36	186	171
37	186	174
38	186	177
39	186	181
40	186	184

selisih

x	y
0	0
0	0
0	0
0	0
0	1
1	2
1	2
2	2
4	3
5	3
5	3
0	0
2	0
2	0
2	0
3	1
2	1
2	1
1	1
2	0
0	0
1	0
3	1
2	0
1	0
3	0
5	2
1	1
7	2
6	1
6	1
1	1
0	1
0	0
0	0
1	1
1	1
1	1
1	2
0	1

41	189	198
42	196	214
43	199	220
44	203	227
45	210	234
46	210	236
47	210	239
48	210	243
49	216	262
50	217	267
51	212	278
52	207	286
53	207	289
54	207	293
55	207	298
56	209	311
57	215	323
58	219	332
59	219	334
60	219	338
61	217	350
62	217	354
63	217	358
64	216	368
65	216	373
66	216	378
67	214	392
68	213	401
69	213	405
70	213	409
71	221	428
72	221	430
73	225	438
74	225	441
75	223	450
76	220	459

41	187	187
42	196	210
43	199	222
44	199	224
45	210	234
46	210	235
47	211	239
48	212	244
49	216	260
50	217	270
51	212	278
52	207	286
53	208	290
54	209	293
55	209	297
56	210	302
57	215	321
58	219	332
59	219	334
60	220	338
61	220	343
62	220	348
63	218	361
64	216	366
65	216	371
66	216	376
67	214	391
68	212	396
69	212	401
70	212	405
71	213	409
72	221	431
73	225	439
74	225	441
75	223	452
76	220	459

2	11
0	4
0	2
4	3
0	0
0	1
1	0
2	1
0	2
0	3
0	0
0	0
1	1
2	0
2	1
1	9
0	2
0	0
0	0
1	0
3	7
3	6
1	3
0	2
0	2
0	2
0	1
1	5
1	4
1	4
8	19
0	1
0	1
0	0
0	2
0	0