



TESIS- TE142599

PENANGANAN OKLUSI PADA PENJEJAKAN OBJEK DENGAN MENGGUNAKAN KAMERA PTZ

SULFAN BAGUS SETYAWAN

NRP. 2213204015

DOSEN PEMBIMBING

Ir. Djoko Purwanto, M. Eng., Ph.D

Ronny Mardiyanto, ST., MT., Ph.D.

PROGRAM MAGISTER

BIDANG KEAHLIAN ELEKTRONIKA

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016



TESIS -TE142599

OCCLUSION HANDLING IN OBJECT TRACKING USING PTZ CAMERA

SULFAN BAGUS SETYAWAN
2213204015

ADVISOR

Ir. Djoko Purwanto, M. Eng., Ph.D.

Ronny Mardiyanto, ST., MT., Ph.D.

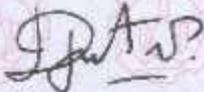
MAGISTER PROGRAM
FIELD STUDY OF ELECTRONIC ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)
di
Institut Teknologi Sepuluh Nopember

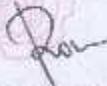
Oleh:
Sulfan Bagus Setyawan
Nrp. 2213204015

Tanggal Ujian : 7 Januari 2016
Periode Wisuda : Maret 2016

Disetujui oleh:



1. **Ir. Djoko Purwanto, M.Eng., Ph.D.** (Pembimbing 1)
NIP: 19651211 199002 1 002

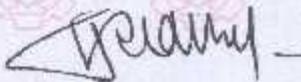


2. **Rouny Mardiyanto, ST., MT., Ph.D.** (Pembimbing 2)
NIP: 19810118 200312 1 003

3. **Dr. Muhammad Rivai, ST., MT.** (Penguji)
NIP: 19690426 199403 1 003



4. **Achmad Arifin, ST., M.Eng., Ph.D.** (Penguji)
NIP: 19710314 199702 1 001



5. **Dr. Tri Arief Sardjono, ST., MT.** (Penguji)
NIP: 19700212 199512 1 001



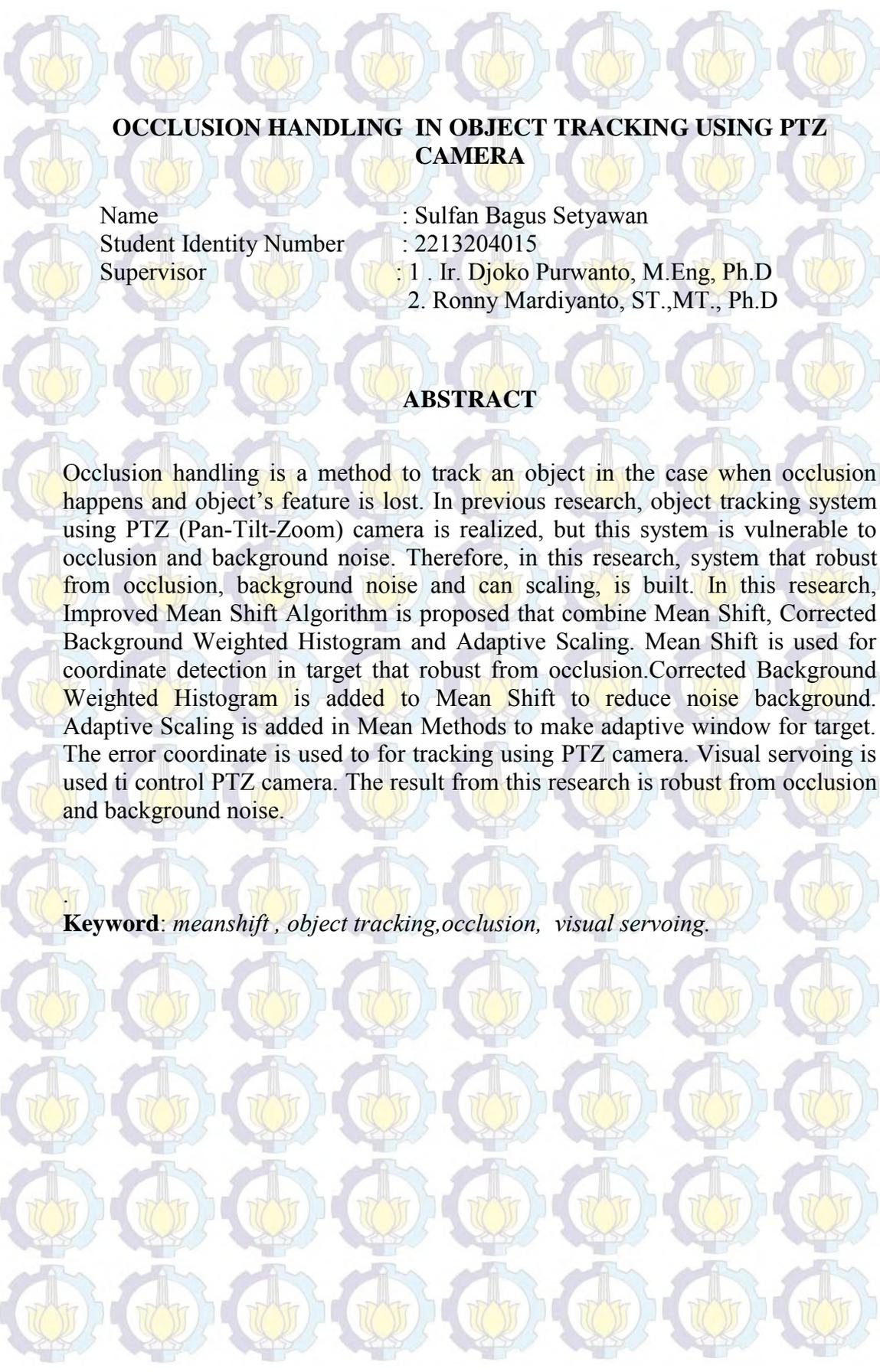
PENANGANAN OKLUSI PADA PENJEJAKAN OBJEK DENGAN MENGUNAKAN KAMERA PTZ

Nama Mahasiswa : Sulfan Bagus Setyawan
NRP : 2213204015
Dosen Pembimbing : 1. Ir. Djoko Purwanto, M.Eng, Ph.D.
2. Ronny Mardiyanto, ST.,MT., Ph.D

ABSTRAK

Penanganan oklusi adalah suatu cara agar objek yang terhalang oleh oklusi atau halangan dapat tetap dijejak ketika fitur dari objek sebagai target hilang. Pada penelitian sebelumnya, telah dilakukan sistem penjejukan objek yang menggunakan kamera PTZ (*Pan-Tilt-Zoom*) dengan memilih sembarang objek, tetapi pada sistem penjejukan objek tersebut tidak tahan terhadap oklusi dan gangguan *noise* background. Maka pada penelitian ini, dibangun sistem penjejukan objek yang tahan terhadap oklusi, background noise dan dapat melakukan penskalaan. Pada penelitian ini, dibangun Algoritma *Improved Mean Shift*, yaitu dengan menggabungkan metode *Mean Shift*, *Corrected Background Weighted Histogram* dan *Adaptive Scaling*. *Mean Shift* digunakan untuk mendeteksi koordinat pada target yang tahan terhadap gangguan oklusi. Metode *Corrected Background Weighted Histogram* ditambahkan pada metode *Mean Shift* untuk menghilangkan noise. *Adaptive scaling* ditambahkan pada metode *Mean Shift*, sehingga window tracking dapat beradaptasi dengan ukuran window. Hasil error dari koordinat target terhadap referensi akan digunakan sebagai input pada kontrol pergerakan kamera PTZ. Kontrol yang digunakan pada penjejukan objek adalah visual servoing. Hasil dari penelitian ini, target dapat tahan terhadap oklusi dan noise background.

Kata kunci: *meanshift, penjejukan objek, oklusi, visual servoing.*



OCCLUSION HANDLING IN OBJECT TRACKING USING PTZ CAMERA

Name : Sulfan Bagus Setyawan

Student Identity Number : 2213204015

Supervisor : 1. Ir. Djoko Purwanto, M.Eng, Ph.D
2. Ronny Mardiyanto, ST.,MT., Ph.D

ABSTRACT

Occlusion handling is a method to track an object in the case when occlusion happens and object's feature is lost. In previous research, object tracking system using PTZ (Pan-Tilt-Zoom) camera is realized, but this system is vulnerable to occlusion and background noise. Therefore, in this research, system that robust from occlusion, background noise and can scaling, is built. In this research, Improved Mean Shift Algorithm is proposed that combine Mean Shift, Corrected Background Weighted Histogram and Adaptive Scaling. Mean Shift is used for coordinate detection in target that robust from occlusion. Corrected Background Weighted Histogram is added to Mean Shift to reduce noise background. Adaptive Scaling is added in Mean Methods to make adaptive window for target. The error coordinate is used to for tracking using PTZ camera. Visual servoing is used to control PTZ camera. The result from this research is robust from occlusion and background noise.

Keyword: *meanshift, object tracking, occlusion, visual servoing.*

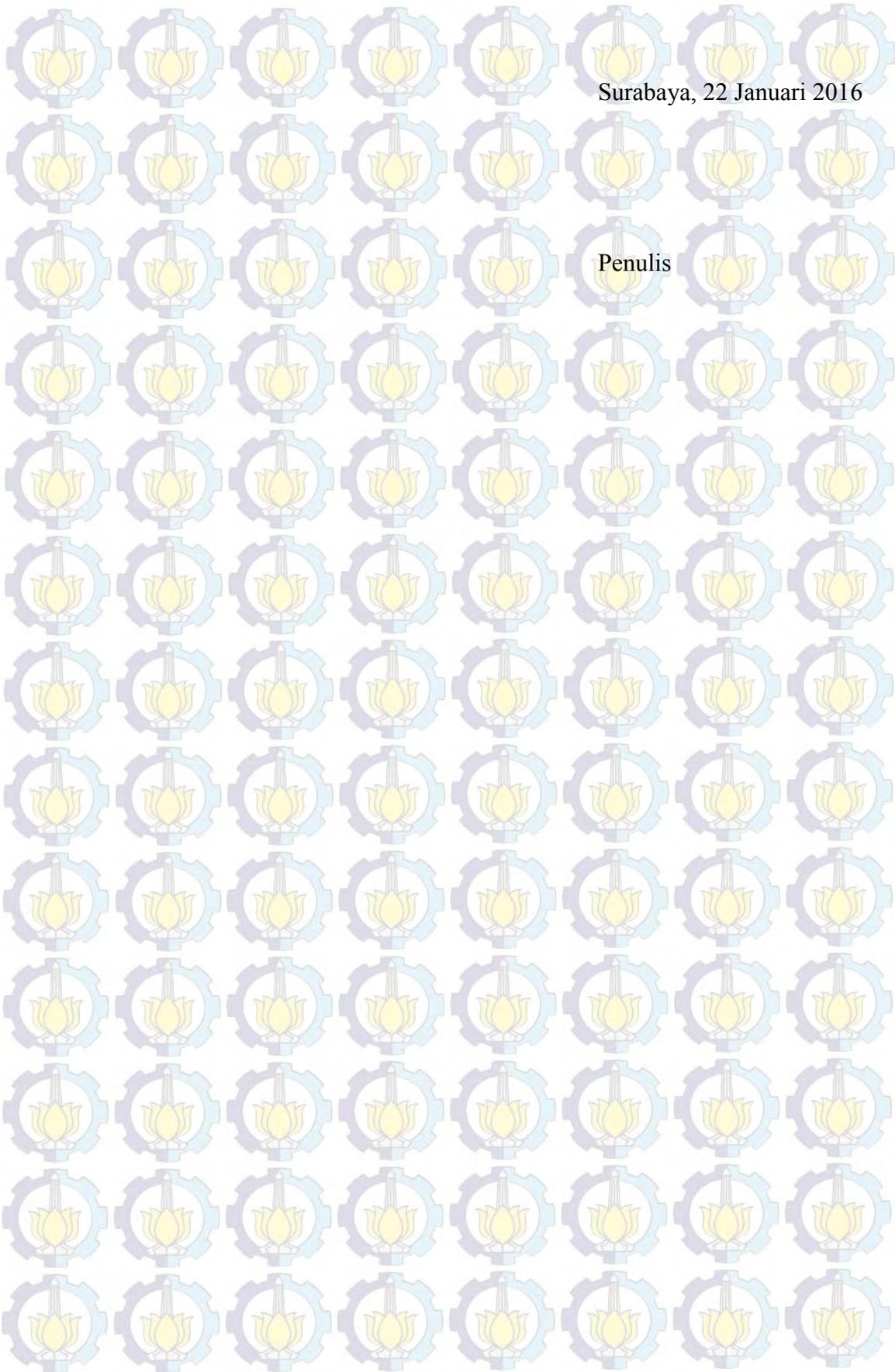
KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala nikmat-Nya lah tesis ini dapat diselesaikan. Tesis berjudul “*Penanganan Oklusi Pada Penjejakan Objek dengan menggunakan kamera PTZ*” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Magister Teknik (MT) pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis menyadari bahwa dalam penyusunan tesis ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Bapak dan Ibu, serta keluarga di Madiun atas segala dukungan dan doanya hingga sampai terselesaikannya tesis ini.
2. Bapak Ir. Djoko Purwanto, M.Eng., Ph.D selaku dosen pembimbing yang telah banyak memberikan bantuan dan saran.
3. Bapak Ronny Mardiyanto, ST, MT., Ph.D selaku dosen pembimbing yang telah banyak memberikan saran dan bantuan.
4. Bapak Achmad Arifin, ST., M.Eng., Ph.D selaku koordinator Bidang Studi Elektronika.
5. Bapak Dr. Muhammad Rivai, ST, MT selaku dosen penguji yang telah banyak memberikan saran.
6. Bapak Dr. Tri Arief Sardjono, ST., MT selaku penguji yang telah banyak memberikan saran.
7. Rekan-rekan seperjuangan angkatan 2013, teman-teman lab B202,B205,B402.

Pada akhirnya, penulis menyadari bahwa tesis ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.



Surabaya, 22 Januari 2016

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.4. Tujuan Dan Manfaat Penelitian	3
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1 Penjejakan Objek	5
2.2 Fishbone Diagram	12
2.3 Visual Servoing	14
2.4 Kalman Filter	22
2.5 Kamera PTZ	24
BAB 3 METODOLOGI PENELITIAN	31
3.1 Perancangan Sistem Penjejakan Objek	32
3.1.1 Representasi Model Target	33
3.1.2 Representasi <i>Background</i>	35
3.1.3 Corrected Background Weighed Histogram (CWBH)	37
3.1.4 Representasi Kandidat Target	38
3.1.5 <i>Mean Shift</i>	38
3.1.6 Metode Penskalaan	39
3.1.7 Algoritma Improved Mean Shift	43
3.2 Perancangan Sistem Kontrol Kamera PTZ	43
3.3 Perancangan Sistem Penjejakan Objek dengan kamera PTZ	44
3.4 Perancangan Deteksi Oklusi pada Penjejakan Objek	46

BAB 4 Hasil Penelitian DAN PEMBAHASAN	49
4.1 Pengujian Representasi Objek.....	49
4.2 Pengujian ketahanan terhadap <i>noise Background</i>	51
4.3 Pengujian Penjejakan Objek	53
4.4 Pengujian Penskalaan.....	62
4.5 Pengujian Kecepatan Metode Penjejakan Objek	66
4.6 Pengujian Ketahanan Metode Penjejakan Objek	68
4.7 Pengujian Kontrol Pan Tilt.....	70
4.8 Pengujian Penjejakan Objek Statis.....	72
4.9 Pengujian Penjejakan Objek bergerak.....	73
4.10 Pengujian Oklusi pada Penjejakan Objek	80
4.11 Pengujian Pengaruh Cahaya terhadap Obyek Target.....	85
4.12 Pengujian Kecepatan Penjejakan Objek dengan Menggunakan Kamera PTZ.....	86
4.13 Pengujian dengan Penjejakan Objek dengan menggunakan Objek yang Berbeda.	87
4.14 Pengujian Penjejakan Objek dengan menggunakan Warna Latar yang Heterogen	89
BAB 5 PENUTUP.....	91
5.1 Kesimpulan	91
5.2 Saran	91
DAFTAR PUSTAKA.....	93

DAFTAR GAMBAR

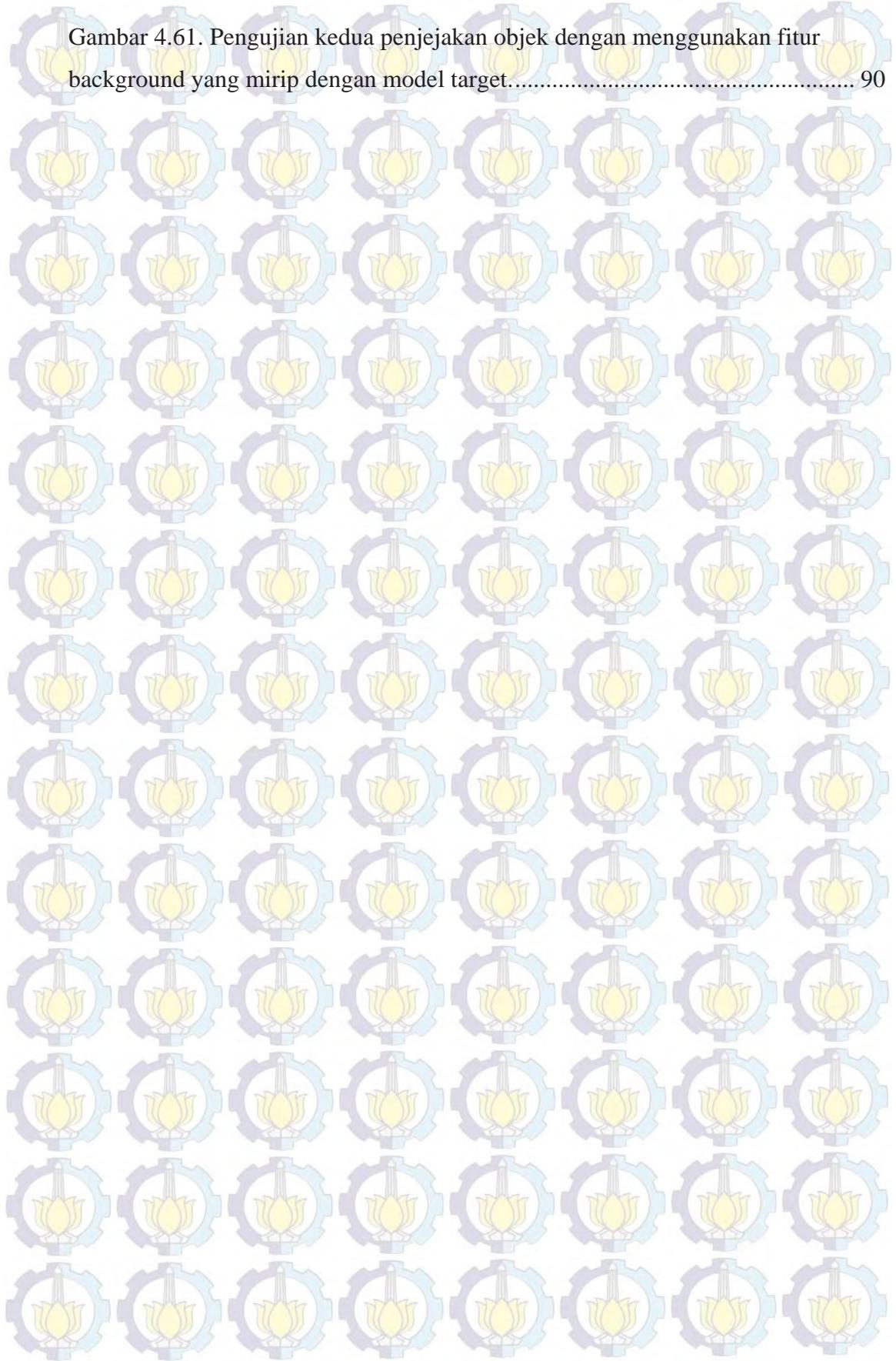
Gambar 2.1. Penjejakan objek dengan PTZ kamera (Xinghua Li dkk., 2012).....	6
Gambar 2. 2. Blok diagram penjejak benda 2-D <i>affine motion</i> (Zhang dkk., 2012).	6
Gambar 2.3. Hasil percobaan penjejak benda 2-D <i>affine motion</i> (Zhang dkk., 2012)	7
Gambar 2. 4. Blok diagram penjejakan sembarang objek (Fahrianur, 2012).	8
Gambar 2. 5. Penjejakan objek berdasarkan kombinasi dari <i>dynamic template</i> <i>matching</i> dan <i>kalman filter</i> (Zheng dkk., 2012).....	9
Gambar 2. 6. Hasil percobaan dengan menggunakan <i>adaptive colour filter</i> (Feng Su dkk., 2012).	10
Gambar 2.7. Hasil percobaan terhadap oklusi berupa kertas (Feng Su dkk., 2012).	10
Gambar 2. 8. Hasil penjejakan dengan menggunakan metode <i>meanshift</i> (Da Tang dkk., 2011).	11
Gambar 2. 9. Hasil penjejakan dengan menggunakan metode <i>mean shift</i> dan partikel filter (Da Tang dkk., 2011).	12
Gambar 2.10. Diagram <i>fishbone</i> penelitian.....	13
Gambar 2. 11. Diagram blok Image-Based Visual Servo(IBVS) kontrol.....	15
Gambar 2. 12. Diagram blok Position-Based Visual Servo(PBVS) kontrol.	15
Gambar 2.13. Ilustrasi Pergerakan Kamer	17
Gambar 2.14. Proyeksi koordinat obyek.....	17
Gambar 2.15. Perbandingan panjang segitiga.....	17
Gambar 2. 16. Arah Pergerakan Kecepatan Sudut.....	18
Gambar 2.17. Hubungan ω dan M dalam bentuk vektor.....	19
Gambar 2. 18. Kamera PTZ (Samsung, 2000).....	25
Gambar 2. 19. Panel Depan S-Video Card kamera PTZ (Samsung, 2000)	26
Gambar 2. 20. Gambar panel SW2 (Samsung, 2000).....	26
Gambar 2. 21. USB 415 PixelView	26
Gambar 3. 1. Diagram blok sistem kerja.....	31

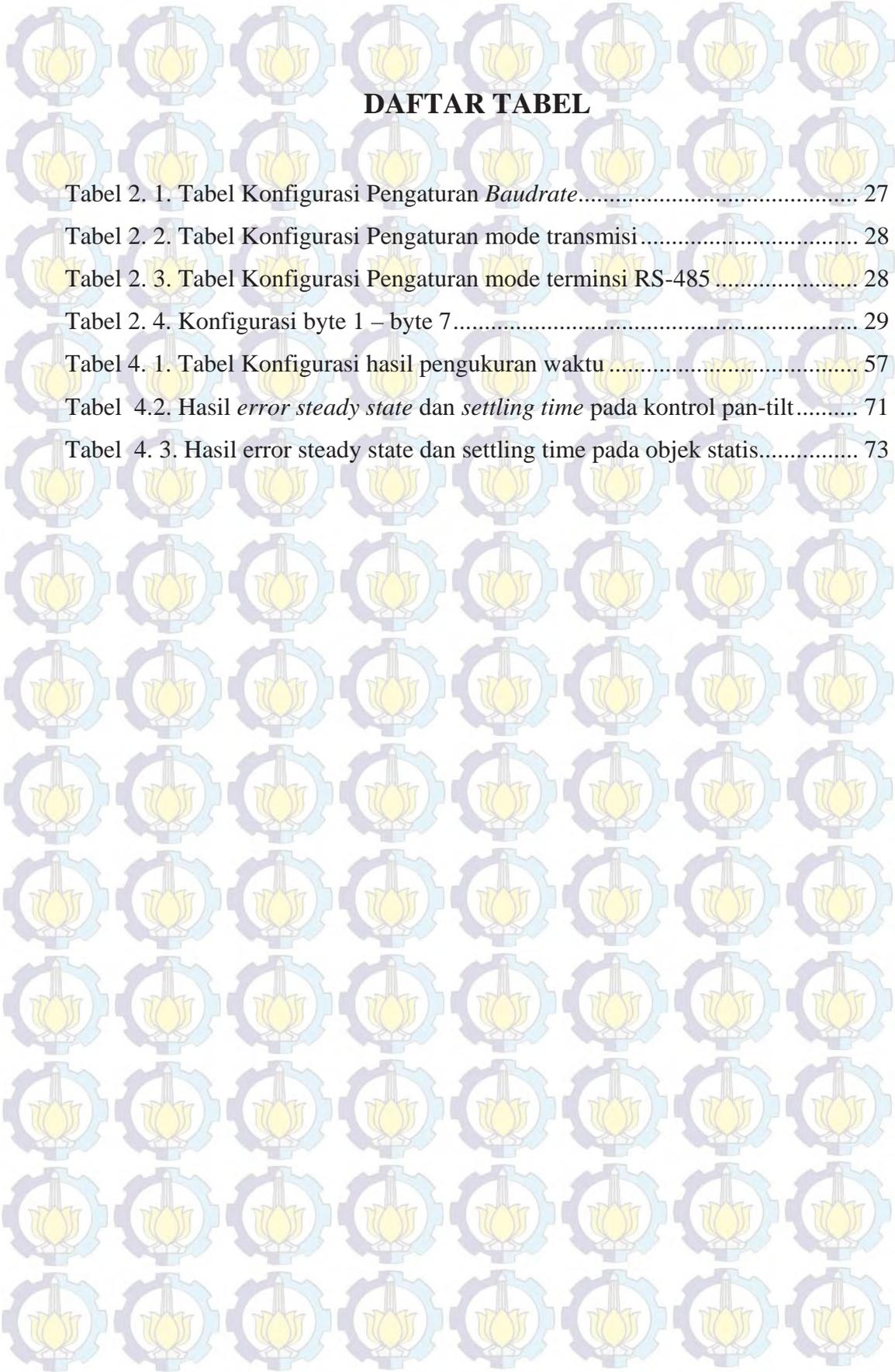
Gambar 3. 2. Tahapan perancangan metode penelitian	32
Gambar 3.3. Diagram blok sistem penjejakan objek.....	33
Gambar 3. 4 Pemilihan objek target	34
Gambar 3. 5. Ukuran <i>background</i>	36
Gambar 3. 6. Blok diagram kontrol kamera PTZ.....	44
Gambar 3. 12. Ilustrasi sistem kerja pada penjejakan objek dengan menggunakan kamera PTZ	45
Gambar 3. 13. Posisi awal objek	46
Gambar 3. 14. Posisi akhir objek.....	46
Gambar 4.1. Hasil pengujian deteksi objek, (a) target berupa wajah, (b) target berupa badan objek, (c) target berupa kaki.....	50
Gambar 4. 2. Histogram probabilitas warna pada wajah.....	50
Gambar 4. 3. Histogram probabilitas warna pada badan.....	50
Gambar 4. 4. Histogram probabilitas warna pada kaki	51
Gambar 4. 5. Objek target dan <i>background</i> yang dipilih	52
Gambar 4. 6. Histogram probabilitas warna pada objek target wajah.....	52
Gambar 4. 7. Histogram probabilitas warna pada <i>background</i>	53
Gambar 4.8. Histogram probabilitas warna hasil CBWH	53
Gambar 4. 9. Hasil pengujian penjejakan objek dengan metode <i>Mean Shift</i> , (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150.....	55
Gambar 4. 10. Hasil pengujian penjejakan objek dengan metode <i>Improved Mean Shift</i> , (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150.....	56
Gambar 4.11. Grafik error penjejakan objek wajah dengan menggunakan <i>Mean Shift</i> dan <i>Improved Mean Shift</i>	56
Gambar 4. 12. Perbandingan iterasi antara <i>Improved Mean Shift</i> dan <i>Mean Shift</i> dalam 20 iterasi.....	58
Gambar 4.13. Hasil pengujian kedua penjejakan objek dengan metode <i>Mean Shift</i> , (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150.....	59
Gambar 4.14. Hasil pengujian kedua penjejakan objek dengan <i>Improved Mean Shift</i> , (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150.....	59
Gambar 4. 15. Grafik error penjejakan objek wajah dengan menggunakan <i>Mean Shift</i> dan <i>Improved Mean Shift</i>	60

Gambar 4. 16. Histogram probabilitas distribusi warna hasil <i>Improved Mean Shift</i> pada penjejakan objek wajah	61
Gambar 4. 17. Grafik penskalaan penjejakan objek wajah dengan menggunakan <i>improved Mean Shift</i> (a) frame ke-1, (b) frame ke-80, (c) frame ke-100, (d) frame ke-150.....	62
Gambar 4. 18. Grafik penskalaan penjejakan objek wajah dengan menggunakan <i>Mean Shift</i> , (a) frame ke-1, (b) frame ke-80, (c) frame ke-100, (d) frame ke-150	63
Gambar 4.19. Grafik tinggi objek wajah dengan menggunakan <i>improved Mean Shift</i>	64
Gambar 4. 20. Grafik lebar objek wajah dengan menggunakan <i>improved Mean Shift</i>	64
Gambar 4. 21. Grafik bobot pada penjejakan objek.....	65
Gambar 4. 22. Grafik koefisien bhattacharrya koefisien antara metode <i>Improved Mean Shift</i> (berwarna biru) dan <i>Mean Shift</i> (berwarna merah).....	65
Gambar 4. 23. Pengujian kecepatan metode objek	66
Gambar 4. 24. Grafik pengujian pada kecepatan 80.81 piksel/ detik	67
Gambar 4. 25. Grafik pengujian pada kecepatan 294.44 piksel/ detik	67
Gambar 4. 26. Grafik pengujian pada kecepatan 421.22 piksel /detik	68
Gambar 4.27. Pengujian ketahanan ketika objek berotasi	69
Gambar 4.28. Grafik koordinat x dan y pada pengujian ketahanan ketika objek berotasi.....	69
Gambar 4. 29. Grafik lebar dan tinggi pengujian ketahanan ketika objek berotasi	70
Gambar 4. 30. Grafik koefisien Bhattacharyya pada pengujian ketahanan objek	70
Gambar 4.31. Grafik hasil pengujian kontrol pan-tilt.....	71
Gambar 4.32. Grafik hasil pengujian pengukuran waktu	71
Gambar 4. 33. Pengujian penjejakan objek dengan objek diam	72
Gambar 4.34. Hasil objek dengan objek diam	73
Gambar 4. 35 Pengujian penjejakan objek bergerak ke kanan	74
Gambar 4.36 Hasil grafik pengujian penjejakan objek dengan objek bergerak kekanan	74

Gambar 4. 37. Hasil grafik error posisi pada penjejakan objek bergerak kekanan	75
Gambar 4.38. Pengujian penjejakan objek bergerak ke kiri	75
Gambar 4. 39. Hasil grafik pengujian penjejakan objek dengan objek bergerak ke kiri	76
Gambar 4. 40. Hasil error posisi dengan objek bergerak ke kiri	76
Gambar 4.41. Pengujian penjejakan objek diagonal kekanan	77
Gambar 4.42. Hasil grafik pengujian penjejakan objek dengan objek bergerak diagonal ke kanan	78
Gambar 4.43. Error posisi objek bergerak diagonal ke kanan	78
Gambar 4. 44. Pengujian penjejakan objek diagonal ke kiri	79
Gambar 4.45. Grafik Pengujian penjejakan objek diagonal ke kiri	79
Gambar 4.46. Grafik Pengujian penjejakan objek diagonal ke kiri	80
Gambar 4. 47. Pengujian oklusi dengan fitur yang berbeda	81
Gambar 4. 48. Koefisien Bhattacharyya pada pengujian oklusi	81
Gambar 4. 49. NCC pada pengujian oklusi	82
Gambar 4. 50. Pengujian oklusi dengan fitur yang sama	83
Gambar 4. 51. Koefisien Bhattacharyya pada pengujian oklusi dengan fitur yang sama	83
Gambar 4. 53. Grafik Pengujian oklusi dibelakang	84
Gambar 4.54. Pengujian terhadap intensitas cahaya yang berbeda-beda (a) intensitas cahaya 5499 lux, (b) intensitas cahaya 799 lux, (c) intensitas cahaya 50 lux, (d) intensitas cahaya 0 lux	85
Gambar 4.55. Pengujian penjejakan objek dengan menggunakan kecepatan 94.574 piksel/detik	86
Gambar 4.56 Pengujian penjejakan objek dengan menggunakan kecepatan 189.574 piksel/detik	86
Gambar 4.57. Pengujian penjejakan objek dengan menggunakan kecepatan 536.974 piksel/detik	87
Gambar 4.58. Pengujian penjejakan objek yang berbeda	88
Gambar 4. 59. Hasil NCC pada penjejakan objek pada Gambar 4.58	88
Gambar 4.60. Pengujian pertama penjejakan objek dengan menggunakan fitur background yang mirip dengan model target	89

Gambar 4.61. Pengujian kedua penjejak objek dengan menggunakan fitur background yang mirip dengan model target..... 90





DAFTAR TABEL

Tabel 2. 1. Tabel Konfigurasi Pengaturan <i>Baudrate</i>	27
Tabel 2. 2. Tabel Konfigurasi Pengaturan mode transmisi.....	28
Tabel 2. 3. Tabel Konfigurasi Pengaturan mode terminasi RS-485.....	28
Tabel 2. 4. Konfigurasi byte 1 – byte 7.....	29
Tabel 4. 1. Tabel Konfigurasi hasil pengukuran waktu.....	57
Tabel 4.2. Hasil <i>error steady state</i> dan <i>settling time</i> pada kontrol pan-tilt.....	71
Tabel 4. 3. Hasil <i>error steady state</i> dan <i>settling time</i> pada objek statis.....	73

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Dengan kemajuan teknologi yang semakin pesat pada era modern ini, banyak sekali bermunculan sistem yang dapat mempermudah kehidupan dan pekerjaan manusia. Disiplin ilmu yang ada diantaranya adalah pengolahan citra. Penjeakan suatu objek benda merupakan salah satu bidang pengolahan citra yang cukup berkembang dewasa ini. Salah satu aplikasinya dalam bidang keamanan (*security system*) seperti pengawasan lokasi (*surveillance*) dimana manusia sebagai objek target. Untuk sistem pengawasan, Zhiwen Chen dkk menggunakan video rekaman (Zhiwen dkk., 2011). Hal itu dapat memberikan bukti kejadian rekaman yang terjadi, tetapi sistem tersebut tidak *real-time*. Kamera yang digunakan dalam sistem pengawasan adalah kamera CCTV, tetapi kamera CCTV memiliki kekurangan yaitu jangkauan kamera yang tidak luas dan titik buta (Jupp dkk., 2007; Nicholson, 1999). Oleh karena itu, pada tesis ini digunakan kamera PTZ yang memiliki jangkauan yang cukup luas.

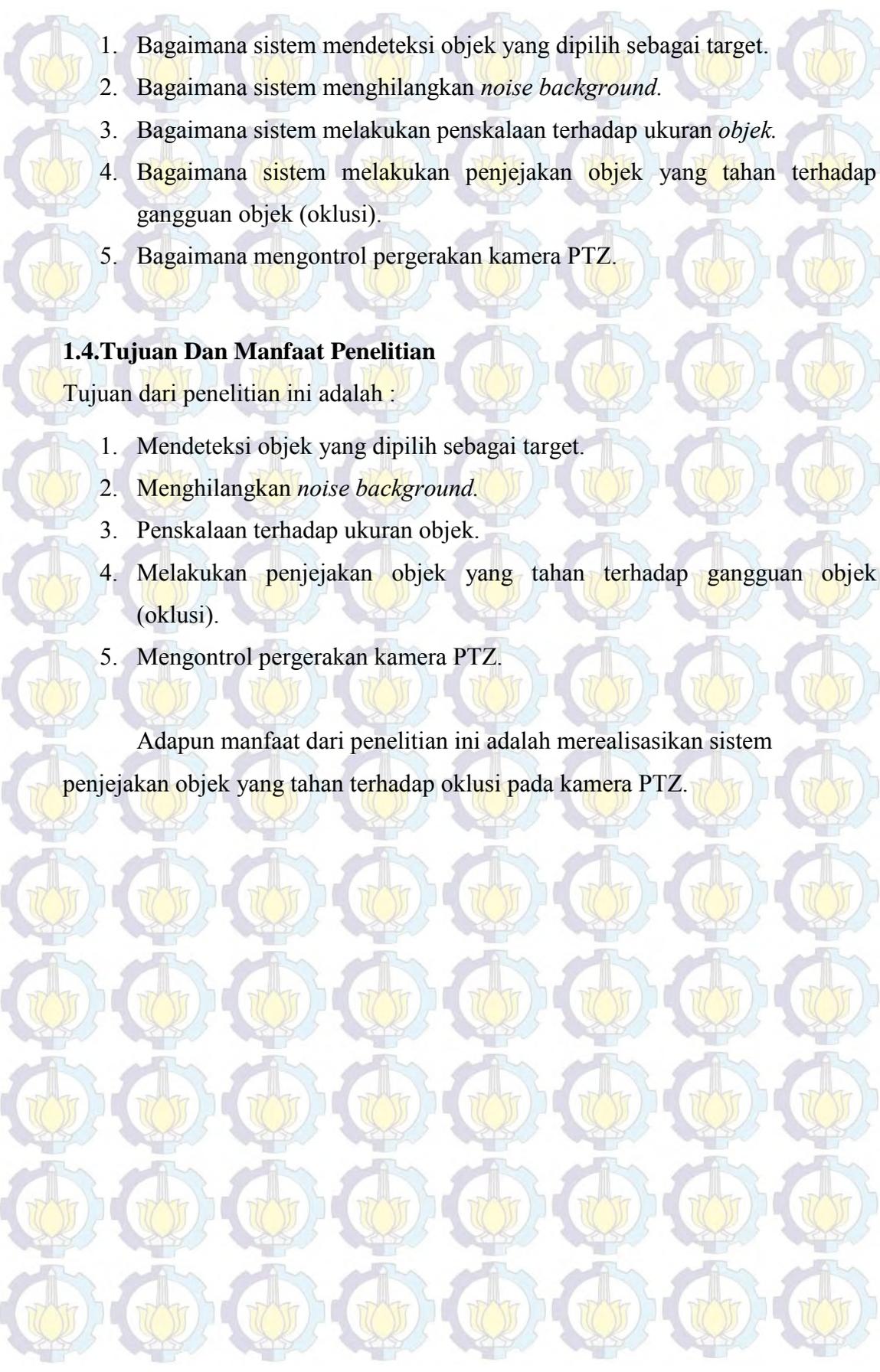
Penelitian tentang penjeakan objek telah banyak dilakukan, diantaranya dilakukan sistem penjeakan objek berbasis visual servoing dengan menggunakan kamera PTZ (Fahrianur, 2013). Pada penelitian tersebut, objek target dapat dipilih sembarang dengan menggunakan *mouse*, dan kamera PTZ akan mengikuti pergerakan dari objek target tersebut. Tetapi pada penelitian tersebut, objek yang terdeteksi tidak handal terhadap oklusi yang menyebabkan sistem tidak dapat menjejak objek target. Oklusi terjadi ketika objek yang dijejak atau atribut penting yang digunakan untuk mengenali identitas dari objek tersebut tidak tersedia pada kamera untuk menjejak ketika objek tersebut masih berada pada jangkauan kamera (Lee dkk., 2014). Salah satu pengembangan penjeakan objek lainnya yaitu kombinasi dari *dynamic template matching* dan *kalman filter* (Zheng dkk., 2012). Pendeteksian objek dilakukan dengan cara mendapatkan area dari daerah penjeak objek dengan menggunakan metode *inter frame difference* dan mengambil fitur SIFT (*Scale Invariant Feature*) pada objek tersebut. Hasil

dari sistem tersebut dapat digunakan untuk menjejak objek yang ditargetkan dan dapat tetap menjejak objek meskipun objek tersebut terhalang oleh benda. Tetapi pada sistem tersebut memerlukan waktu yang lama akibat dari *update template* pada saat hilangnya fitur dan penelitian tersebut hanya menggunakan video.

Dari penelitian yang telah ada, maka dapat dikembangkan sistem penjejakan objek yang tahan terhadap oklusi pada kamera PTZ. Pada penelitian ini, objek target dipilih dengan menggunakan mouse. Kemudian objek akan dijejak dengan menggunakan metode *mean shift*. Metode ini digunakan karena metode ini mampu digunakan untuk menangani oklusi sebagian. Metode *Mean Shift* memiliki kelemahan yaitu tidak dapat digunakan untuk penskalaan, rentan terhadap *noise background*. Untuk menangani permasalahan pada *Mean Shift*, maka digunakan metode *Improved Mean Shift*. Metode *Improved Mean Shift* digunakan untuk menghilangkan *noise background* dan *window* penjejakan dapat menyesuaikan terhadap ukuran objek. Untuk penskalaan maka digunakan menggunakan *zeroth order moment*, *second order moment* dan koefisien Bhattacharyya antara model target dan kandidat target untuk mengestimasi lebar dan tinggi objek. Untuk menghilangkan *noise background*, maka digunakan metode *Corrected Background Weighted Histogram* dan digabungkan ke metode penjejakan *Mean Shift*. Hasil dari *Improved Mean Shift* berupa koordinat x dan y titik tengah dari objek target. selisih koordinat objek target dan koordinat referensi digunakan sebagai input kontrol pada kamera PTZ. Untuk mengurangi *noise* pada penjejakan objek dengan menggunakan kamera PTZ, metode kalman filter diaplikasikan pada output dari hasil *Improved Mean Shift*. Metode kontrol yang digunakan untuk dalam kamera PTZ adalah *visual servoing*. Output dari kontrol *visual servoing* adalah kecepatan pan dan tilt yang digunakan untuk menggerakkan kamera PTZ secara vertikan dan horizontal. Untuk mendeteksi adanya oklusi atau tidak maka ditambahkan dua parameter yaitu Normalized Cross Correlation (NCC) dan koefisien Bhattacharyya.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat disusun rumusan masalah sebagai berikut:

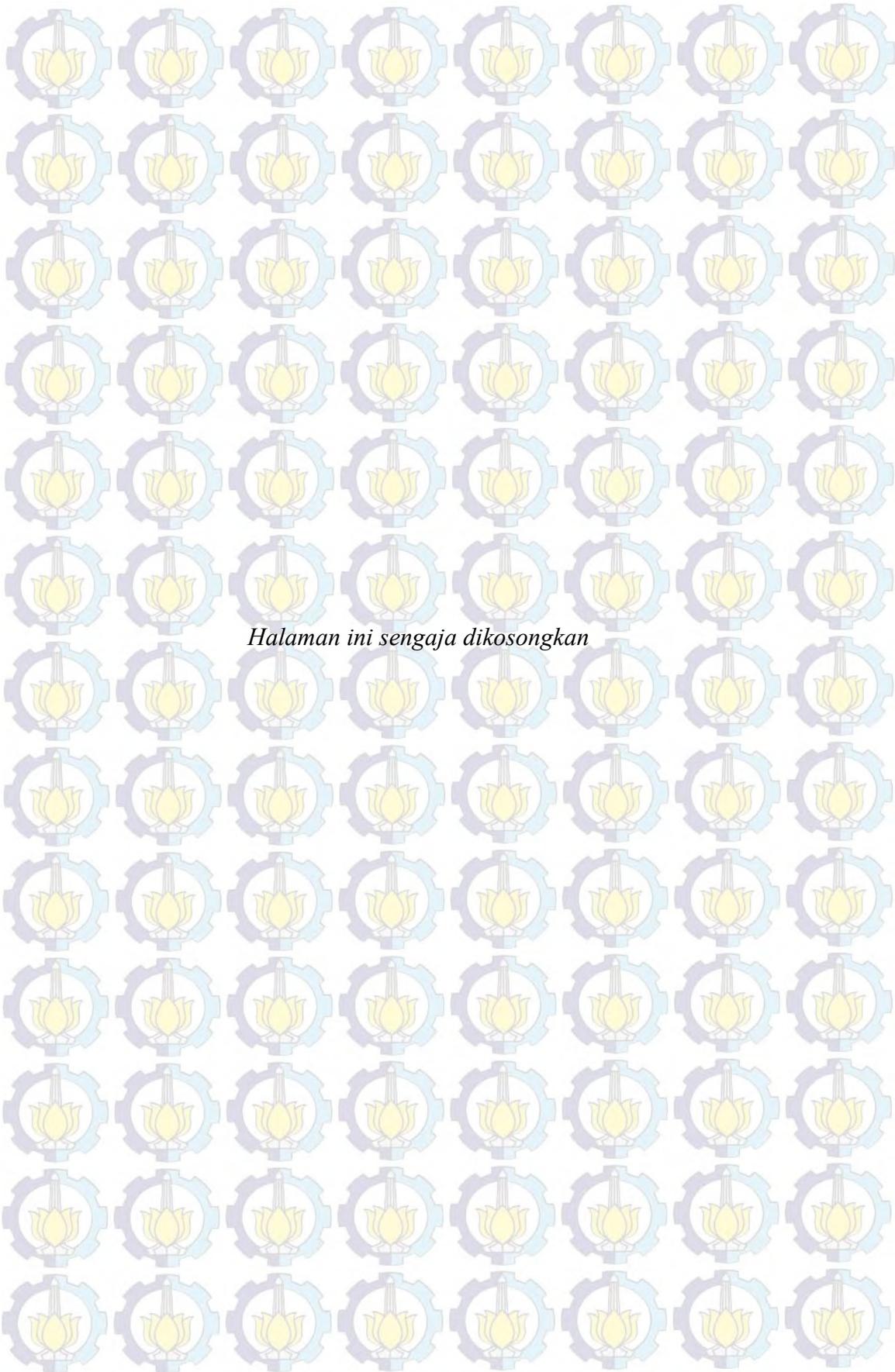
- 
1. Bagaimana sistem mendeteksi objek yang dipilih sebagai target.
 2. Bagaimana sistem menghilangkan *noise background*.
 3. Bagaimana sistem melakukan penskalaan terhadap ukuran *objek*.
 4. Bagaimana sistem melakukan penjejakan objek yang tahan terhadap gangguan objek (oklusi).
 5. Bagaimana mengontrol pergerakan kamera PTZ.

1.4. Tujuan Dan Manfaat Penelitian

Tujuan dari penelitian ini adalah :

1. Mendeteksi objek yang dipilih sebagai target.
2. Menghilangkan *noise background*.
3. Penskalaan terhadap ukuran objek.
4. Melakukan penjejakan objek yang tahan terhadap gangguan objek (oklusi).
5. Mengontrol pergerakan kamera PTZ.

Adapun manfaat dari penelitian ini adalah merealisasikan sistem penjejakan objek yang tahan terhadap oklusi pada kamera PTZ.



BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada BAB 2 akan dibahas tentang kajian pustaka tentang penjejakan objek, diagram fishbone penelitian, dan dasar teori tentang *Mean Shift*, *Visual Servoing*, *Kalman Filter* dan kamera PTZ.

2.1 Penjejakan Objek

Penjejakan objek adalah isu yang populer karena disebabkan aplikasi potensial dari komputer vision yaitu video pengawasan, diagnosis medika dan robotika (Zhiwen dkk., 2011). Berikut ini merupakan perkembangan dari penelitian penjejakan objek. Zhiwen Chen dkk mengembangkan metode baru untuk menjejak objek dengan menggunakan *optical flow*. Pada penelitian tersebut, *optical flow* digunakan untuk mendapatkan kontur dari objek. Langkah pertama yang digunakan untuk menjejak objek adalah dengan menggunakan algoritma *Horn-Schunk* untuk mendapatkan kecepatan vector pada video (Zhiwen dkk., 2011). Dari vektor tersebut maka akan diperoleh posisi dari pixel yang bergerak antara frame. Langkah berikutnya mendapatkan kontur objek dan mengekstaksi objek. Langkah ketiga menghitung posisi dan kecepatan dari objek yang bergerak. Dari hasil penelitian tersebut dapat diketahui bahwa algoritma tersebut dapat menjejak objek. Tetapi pada algoritma ini rentan terhadap *noise* lingkungan, sehingga ketika ada objek yang bergerak berseberangan memungkinkan target akan hilang. Hal ini disebabkan karena yang dideteksi berdasarkan *vector* dari pergerakan objek.

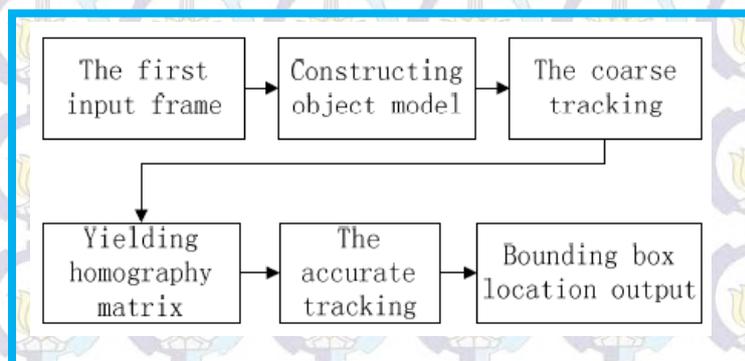
Xinghua Li dkk, melakukan pendeteksian dan penjejakan benda yang bergerak dengan menggunakan PTZ kamera. Untuk mendeteksi objek maka digunakan algoritma *background subtraction* untuk mendapatkan gambar biner setelah segmentasi gambar. Segmentasi gambar tersebut menghasilkan banyak noise sehingga digunakan metode *expansion* dan *corrosion* untuk menghilangkannya (Xinghua Li dkk., 2012).



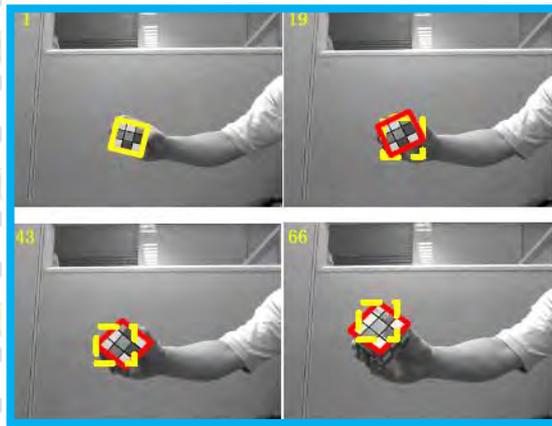
Gambar 2.1. Penjejakan objek dengan PTZ kamera (Xinghua Li dkk., 2012)

Kemudian kamera akan bergerak rotasi untuk mengikuti objek tersebut dengan menggunakan algoritma *timely replacement* berdasarkan *central region*. Pada Gambar 2.1 merupakan hasil percobaan pada penelitian ini. Dari hal tersebut dapat diketahui bahwa sistem dapat mengikuti objek berupa orang dan kamera bergerak rotasi ke arah orang tersebut. Metode yang dibangun berhasil mendeteksi pergerakan objek, tetapi metode tersebut akan terjadi masalah ketika ada dua buah objek yang bergerak.

Miaohui Zhang dkk melakukan penelitian tentang penjejak benda yang secara efektif terhadap benda 2-D *affine motion* pada benda (Zhang dkk., 2012). Langkah pertama yaitu daerah gambar objek dipilih dari frame pertama, SIFT (*Scale Invariant Feature*) yang diekstrak dari daerah gambar objek digunakan sebagai model objek.



Gambar 2. 2. Blok diagram penjejak benda 2-D *affine motion* (Zhang dkk., 2012).

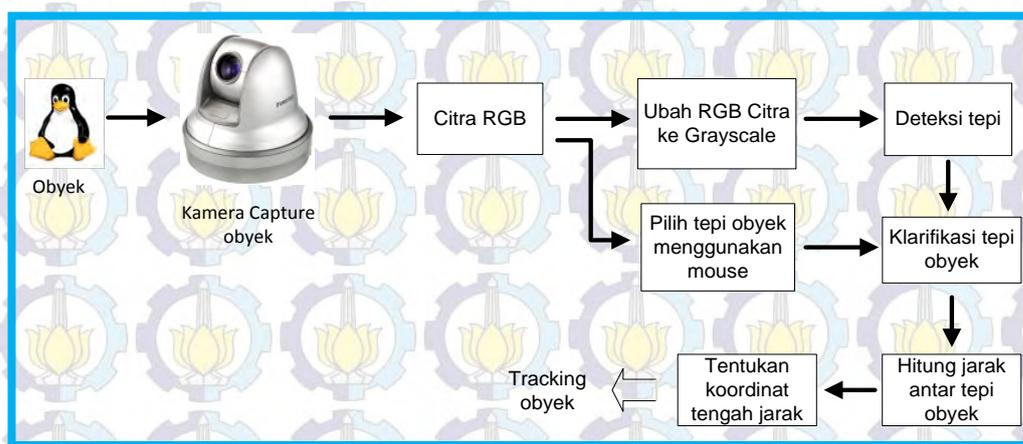


Gambar 2.3. Hasil percobaan penjejak benda 2-D *affine motion* (Zhang dkk., 2012)

Langkah yang kedua, kemiripan yang paling tinggi awalnya diposisikan dengan menggunakan algoritma partikel filter pada frame berikutnya dan lebih dari 3 titik secara acak dipilih untuk membuat matriks transform yang layak untuk bidang 2D dengan menggunakan algoritma konsensus. Kemudian lokasi tersebut diberi tanda. Pada Gambar 2.2 merupakan blok diagram yang digunakan.

Pada Gambar 2.3 merupakan hasil dari percobaan yang dilakukan. Pada hasil tersebut dapat diketahui bahwa sistem dapat mengikuti objek secara presisi. Dari paper tersebut dapat diketahui bahwa sistem *homography matriks* dapat digunakan untuk menjejak objek tetapi objek yang dideteksi harus benda yang *rigid*.

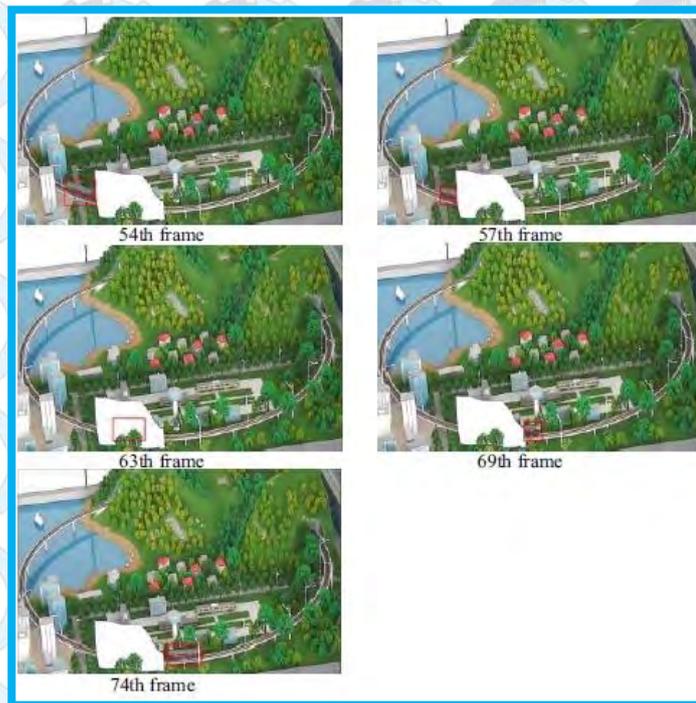
Fahrianur mengembangkan penjejakan sembarang objek dengan menggunakan kamera PTZ. Penjejakan sembarang objek dilakukan dengan menggunakan pengabungan antara deteksi tepi dan *optical flow* (Fahrianur, 2012). Deteksi tepi digunakan untuk mengetahui lebar objek, *optical flow* digunakan untuk menjejak objek target.



Gambar 2. 4. Blok diagram penjejakan sembarang objek (Fahrianur, 2012).

Titik *optical flow* diberikan pada lebar objek target tersebut. Kemudian menentukan titik tengah koordinat dari objek target yang digunakan untuk menjejak objek. Penentuan koordinat tengah objek dengan menggunakan *Eucluidian Distance*. Pada Gambar 2.4 merupakan blok diagram yang digunakan pada penelitian yang dilakukan oleh Fahrianur. Pada penelitian tersebut dihasilkan metode yang dapat menjejak sembarang objek, tetapi penjejakan objek tersebut tidak tahan terhadap oklusi. *Optical flow* pada sistem tersebut hanya digunakan sebagai penjejakannya, sedangkan untuk mengidentifikasi digunakan metode *edge detection*.

Penjejakan objek yang kuat merupakan masalah yang paling menantang pada bidang komputer vision, dimana objek yang terdeteksi sebagai target akan tetap terdeteksi ketika terhalang oleh adanya oklusi. Oklusi terjadi ketika objek yang dijejak atau atribut penting yang digunakan untuk mengenali identitas dari objek tersebut tidak tersedia pada kamera untuk menjejak ketika objek tersebut masih berada pada jangkauan kamera (Lee dkk., 2014). Oklusi dapat disebabkan karena terhalang oleh objek lain dan struktur dari *background*. Dewasa ini telah banyak pengembangan yang dilakukan untuk mengatasi oklusi tersebut. Bin Zheng dkk melakukan pengembangan algoritma penjejakan objek berdasarkan kombinasi dari *dynamic template matching* dan *kalman filter* (Zheng dkk., 2012).

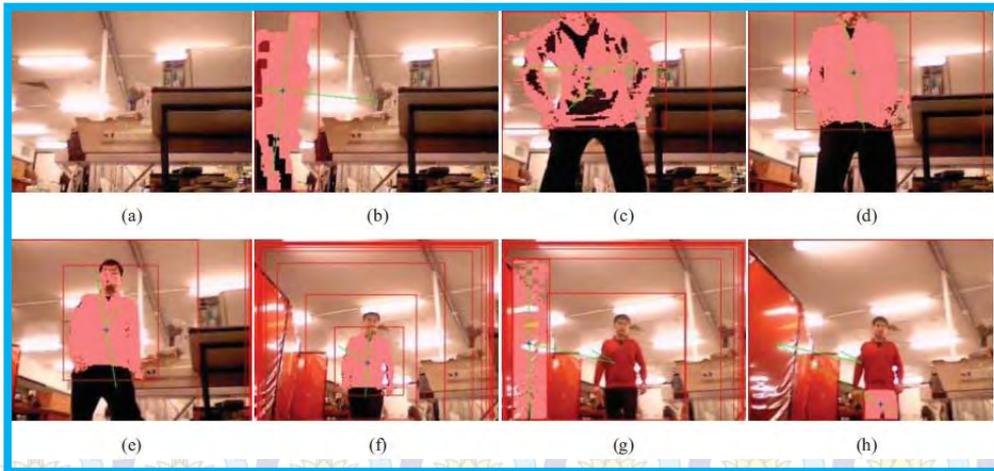


Gambar 2. 5. Penjejakan objek berdasarkan kombinasi dari *dynamic template matching* dan *kalman filter* (Zheng dkk., 2012).

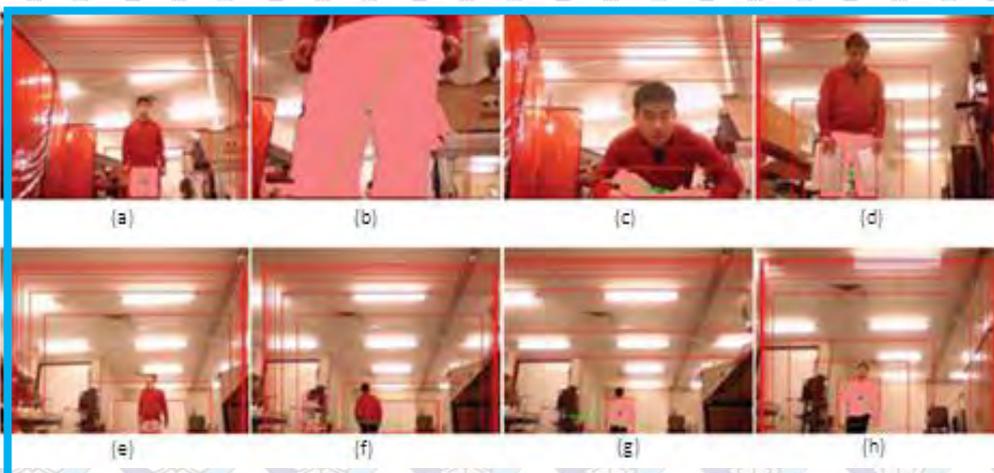
Pendeteksian objek dilakukan dengan cara mendapatkan area dari daerah penjejak objek dengan menggunakan metode *inter frame difference* dan mengambil fitur SIFT pada objek tersebut. Kemudian kalman filter akan mengestimasi lokasi dari objek yang ditarget yang cocok dengan template. Ketika fitur point hilang dan dibawah nilai *threshold* yang ditentukan maka *template* akan diupdate kembali. Sehingga sistem tahan terhadap gangguan. Hasil dari sistem tersebut dapat digunakan untuk menjejak objek yang ditargetkan dan dapat tetap menjejak objek meskipun objek tersebut terhalang oleh benda. Hal ini dapat ditunjukkan pada Gambar 2.5, tetapi pada sistem tersebut memerlukan waktu yang lama akibat dari *update template* pada saat hilangnya fitur point.

Feng Su and Gu Fang melakukan penelitian tentang metode untuk penjejakan benda yang bergerak dengan menggunakan *adaptive colour filter*. Sistem tersebut menggunakan robot sebagai penggerak dari kamera. Pada awalnya robot dan kamera berada dalam kondisi tidak bergerak. Objek yang bergerak dapat dideteksi dengan menggunakan metoda *background subtraction* (Feng Su dkk; 2012). Ketika benda bergerak terdeteksi, informasi warna objek dan

background difilter untuk mendapatkan fitur yang paling menonjol dari objek untuk dijejak. Fitur tersebut digunakan untuk menjejak objek yang digunakan. Ketika fitur warna yang dipilih menjadi tidak menonjol diakibatkan perubahan lingkungan maka metode *adaptive colour filter* digunakan untuk mengupdate pemilihan filter warna. . Pada Gambar 2.6 (a-d) merupakan inialisasi objek yang bergerak dan memilih objek yang bergerak. Gambar 2.6 (e-h) menunjukkan pengujian ketika ada warna yang sama yaitu merah pada lingkungan sekitar.



Gambar 2. 6. Hasil percobaan dengan menggunakan *adaptive colour filter* (Feng Su dkk., 2012).



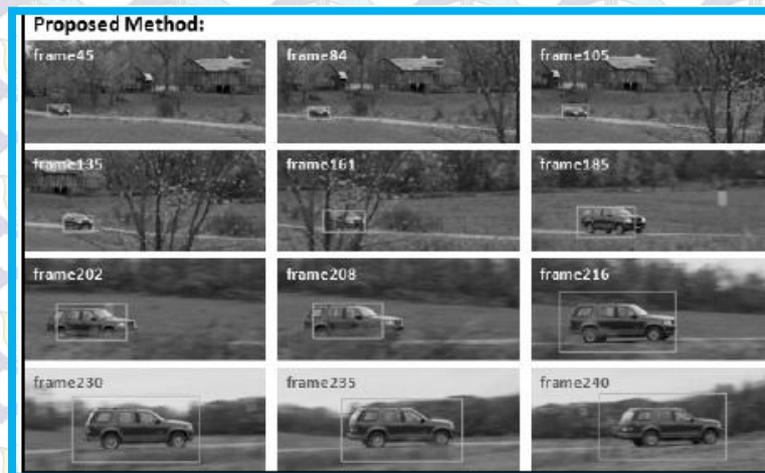
Gambar 2.7. Hasil percobaan terhadap oklusi berupa kertas (Feng Su dkk., 2012).

Maka sistem akan melakukan inisialisasi warna yang digunakan ketika objek bergerak. Ketika objek bergerak., maka sistem akan memilih warna hitam. Kemudian sistem akan menjejak warna hitam tersebut. Dari Gambar 2.7 (a-h) merupakan pengujian terhadap oklusi berupa kertas warna putih. Dari hasil percobaan tersebut maka dapat diketahui bahwa sistem tahan terhadap oklusi. Pada Gambar 2.7 (e-h) merupakan pengujian ketika target menjauh, maka dari sistem tersebut dapat mengikuti target. Kemudian terdapat warna yang sama maka sistem akan menginisialisasi warna ketika ada warna hitam yang sama dan memilih warna merah lagi. Pada sistem tersebut dapat mendeteksi objek yang bergerak dan tahan terhadap oklusi, tetapi ukuran tersebut bergantung pada warna yang dipilih, sehingga tidak berdasarkan ukuran yang dipilih.

Da Tang dan Yu-Jin Zhang menggabungkan metode *meanshift* dan *particle filter* untuk menjejak objek. Pada sistem ini *meanshift* digunakan untuk penjejakan awal sedangkan partikel filter digunakan untuk menjejak ketika *meanshift* tidak bisa menjejak dengan batas *threshold* yang telah ditentukan (Da tang dkk., 2011).



Gambar 2. 8. Hasil penjejakan dengan menggunakan metode *meanshift* (Da Tang dkk., 2011).

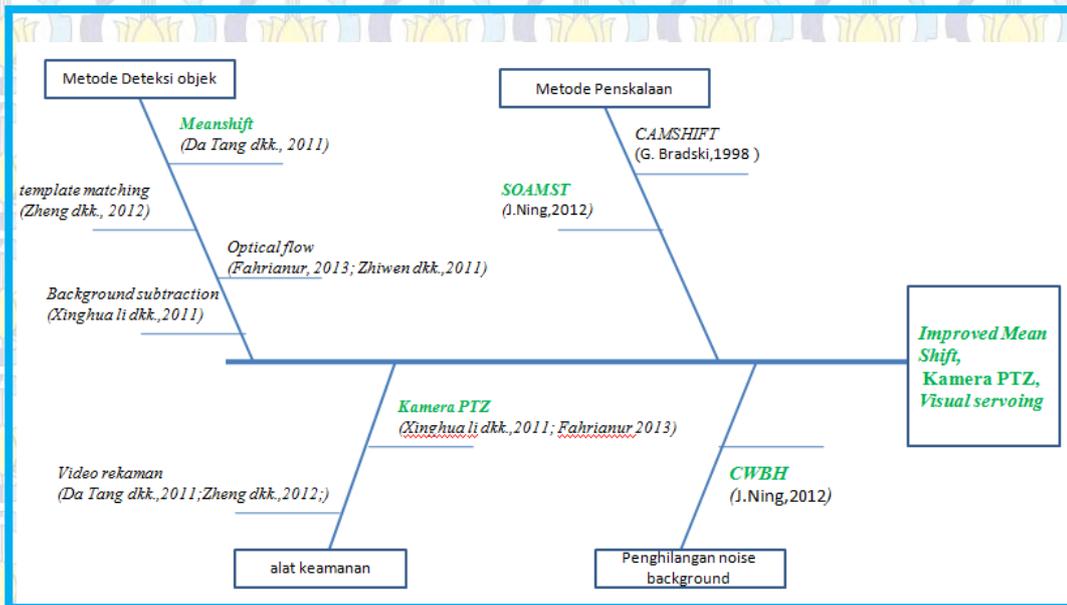


Gambar 2. 9. Hasil penjejakan dengan menggunakan metode *mean shift* dan partikel filter (Da Tang dkk., 2011).

Metode meanshift dapat digunakan sebagai segmentasi dan penjejakan (Pahani dkk., 2013). Hasil dari penjejakan tersebut digunakan untuk mengupdate data objek target dengan menggunakan *histogram fusion*. Pada Gambar 2.8 merupakan hasil dari algoritma yang digunakan dan membandingkan dengan metode *meanshift*. Pada Gambar 2.8 dapat diketahui bahwa target akan hilang pada frame 240 dengan adanya metode penggabungan maka akan diperoleh hasil yang lebih baik. Terlihat pada Gambar 2.9 dapat diketahui bahwa pada frame 240 algoritma dapat mengikuti target yang dituju.

2.2 Fishbone Diagram

Pada Gambar 2.10 merupakan fishbone diagram penelitian yang terkait dengan tesis yang dikerjakan. Pada diagram fishbone tersebut dibagi menjadi 3 bagian yaitu metode deteksi objek, metode estimasi objek dan berdasarkan alat keamanan yang digunakan. Pada metode deteksi objek, terdapat berbagai metoda yang digunakan pada penelitian sebelumnya yaitu *meanshift* yang dilakukan oleh Da Tang pada tahun 2011. Pada tahun sebelumnya, Zheng telah melakukan penelitian pendeteksian objek dengan menggunakan *template matching* dengan mencocokkan objek target dengan template yang tersedia.



Gambar 2.10. Diagram *fishbone* penelitian.

Pendeteksian objek dengan menggunakan *optical flow* yaitu memberi titik pada objek target, kemudian titik tersebut menjadi acuan perpindahan objek berdasarkan intensitas warna (Fahrianur, 2013; Zhiwen Dkk., 2011). Metode *background subtraction* digunakan untuk mendeteksi objek ketika ada objek yang bergerak. Pada bagian penskalaan objek, terdapat CAMSHIFT untuk menjejak wajah G Bradski, pada tahun 1998, kemudian pada tahun 2012 dilakukan penelitian dengan menggunakan metode SOAMST yang dilakukan L Ning. Pada tahun yang sama L Ning melakukan penelitian dengan menggunakan CBWH (*Corrected Background Weighted Histogram*) untuk menghilangkan *noise background*.

Pada bagian alat keamanan dibagi menjadi 2 yaitu dengan menggunakan video (Da Tang Dkk., 2011; Zheng Dkk., 2012) dan juga menggunakan kamera PTZ (Xinghua li Dkk., 2011; Fahrianur 2013). Pada tesis ini, digunakan metode *meanshift* sebagai pendeteksi objek. Untuk mengestimasi pergerakan posisi objek digunakan kalman filter, karena objek yang bergerak cenderung linier. Alat keamanan yang digunakan adalah kamera PTZ yang memiliki fasilitas *pan-tilt-zoom* dengan menggunakan kontrol *visual servoing* pada *pan-tilt*.

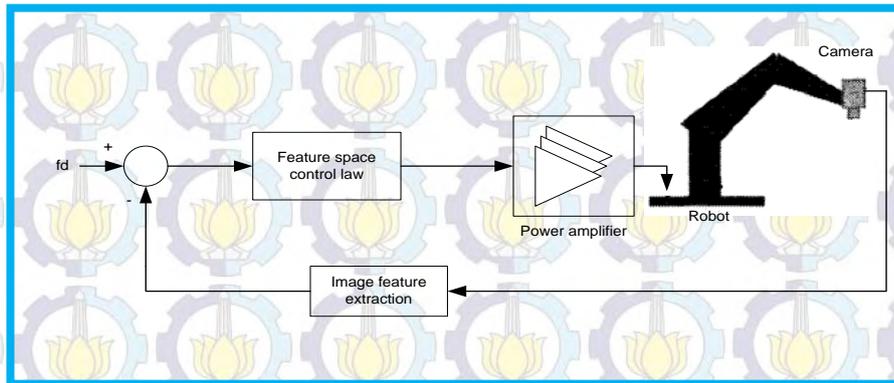
2.3 Visual Servoing

Salah satu contoh kontrol pergerakan PTZ dengan menggunakan visual servoing. Visual servo, atau dikenal sebagai *Vision Based Robot Kontrol*, merupakan suatu teknik dari pemrosesan gambar, *computer vision*, dan teori kontrol. Teknik ini menggunakan informasi umpan balik yang meng-ekstrak data dari kamera untuk mengontrol pergerakan robot. Robot yang dimaksud pada umumnya adalah robot manipulator atau *mobile robot*. Kamera diletakkan pada robot, sehingga jika terjadi pergerakan pada robot, maka kamera juga akan bergerak. Tujuan dari penggunaan visual servo, adalah untuk meminimalisasi nilai *error* $e(t)$ yang terjadi, ditunjukkan dalam Persamaan (2.1).

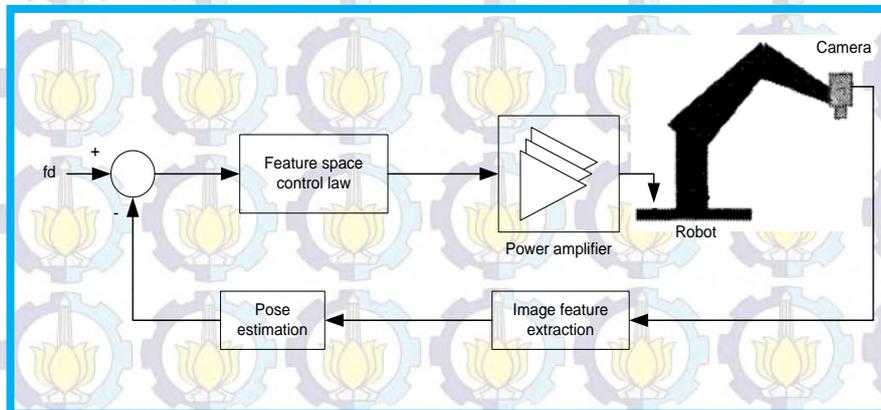
$$e(t) = s(m(t), a) - s^* \quad (2.1)$$

Persamaan (2.17) adalah persamaan umum. Vektor $m(t)$ adalah hasil yang terukur koordinat titik gambar yang diinginkan atau koordinat titik tengah gambar. Sedangkan a adalah parameter yang merepresentasikan informasi tambahan mengenai sistem, misalkan pemodelan obyek 3-D. Kemudian s^* adalah target nilai yang diharapkan dari gambar (Francois dkk., 2006).

Pada struktur kontrolnya, jika arsitektur membentuk hierarki dan menggunakan sistem *vision* untuk menyediakan *input set-point* untuk kontroller pada tingkat *joint*, yang karenanya menggunakan umpan balik dari *joint* untuk menstabilkan robot secara internal, maka sistem disebut dengan “*dynamic look-and-move sistem*”. Sebaliknya, *direct visual servo* (*visual servo* tanpa hirarki) menghilangkan kontroler robot dan mengganti seluruhnya dengan kontroller visual servo yang secara langsung menghitung *input joint*, dan karenanya hanya menggunakan penglihatan (*vision*) untuk menstabilkan keseluruhan sistem.



Gambar 2. 11. Diagram blok Image-Based Visual Servo (IBVS) kontrol.



Gambar 2. 12. Diagram blok Position-Based Visual Servo (PBVS) kontrol.

Visual servoing dikelompokkan menjadi 2 macam yaitu kontrol *Image-Based Visual Servo* (IBVS) dan kontrol *Position-Based Visual Servo* (PBVS) yang prinsip kerjanya masing-masing ditunjukkan dalam Gambar 2.11 dan Gambar 2.12 (Francois dkk., 2006; Hutchinson., 1996).

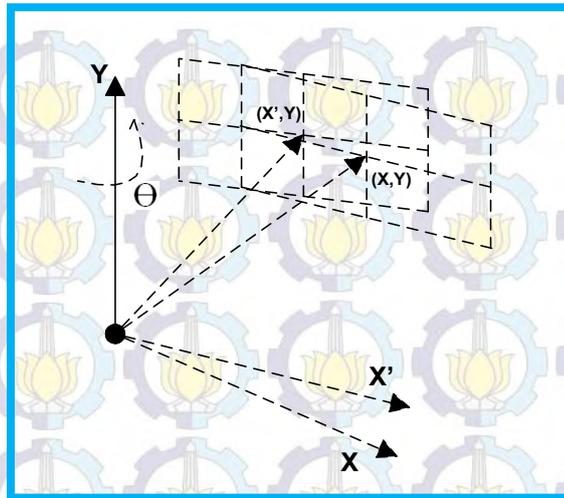
- Kontrol *Image-Based Visual Servo* (IBVS), bekerja berdasarkan keadaan dari gambar yang akan diproses, dengan cara mencari nilai *error* antara fitur sekarang dengan fitur yang diinginkan. Fitur yang dimaksud adalah berupa nilai titik koordinat gambar. Pergerakan kamera diatur dari nilai *error* tersebut sampai mencapai nilai yang diinginkan.
- Kontrol *Position-Based Visual Servo* (PBVS), bekerja dengan menggunakan arah pandangan tiga dimensi yang diperoleh dari

rekonstruksi gambar untuk mengatur pergerakan kamera ke posisi yang diinginkan. Beberapa metode untuk merekonstruksi gambar dapat digunakan dalam PBVS, namun harus diketahui lebih dahulu mengenai jarak antara kamera dengan gambar target yang dijadikan referensi.

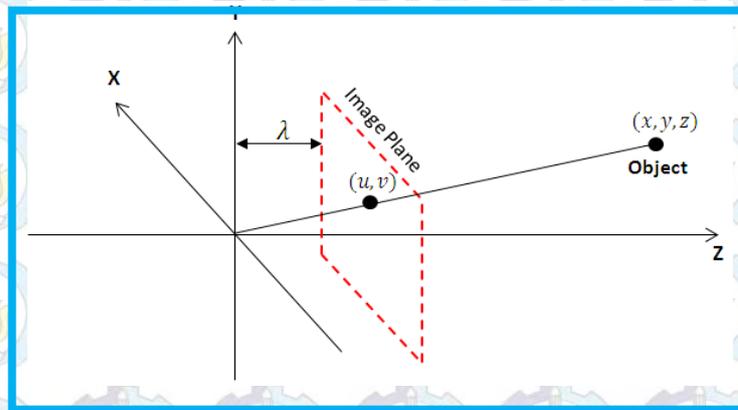
Pada penelitian yang dilakukan oleh Fahrianur (2012), input dari kontrol visual servoing berupa error koordinat x dan y . Koordinat pada kontrol pan dan tilt dilambangkan dengan s^* . Jadi dapat dituliskan $s^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Sedangkan s adalah titik koordinat obyek pada *frame* kamera dalam satuan *pixel* yang dilambangkan dengan X dan Y . Sehingga untuk nilai e dapat dituliskan sebagai berikut,

$$e = s - s^* \quad (2.2)$$

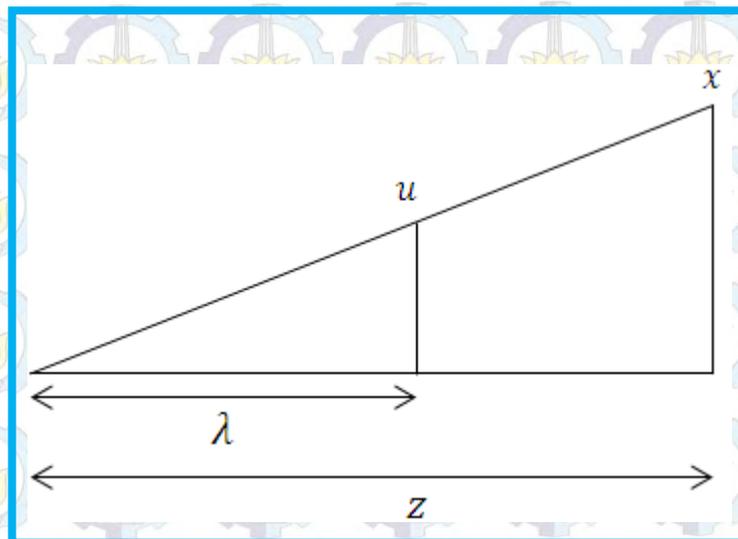
Ketika target bergerak, maka posisi koordinat pada layar akan berubah. Kontroler mengeksekusi perubahan koordinat antara posisi yang diinginkan s^* dengan posisi target yang baru s , kemudian kamera akan bergerak menuju posisi target tersebut sampai target yang tampak pada layar berada pada koordinat tengah atau mencapai s^* . Ilustrasi pergerakan kamera ditunjukkan dalam Gambar 3.7. Koordinat obyek pada *frame* kamera dan koordinat obyek sebenarnya dituliskan dalam bentuk matrik yang secara berturut-turut adalah $s = \begin{bmatrix} u \\ v \end{bmatrix}$ dan $M = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. Berikut adalah gambar proyeksi koordinat obyek sebenarnya terhadap *image plane* yang ditunjukkan dalam Gambar 2.13.



Gambar 2.13. Ilustrasi Pergerakan Kamera



Gambar 2.14. Proyeksi koordinat obyek



Gambar 2.15. Perbandingan panjang segitiga

untuk mencari nilai u dan v dapat dicari dengan cara memodelkan hasil proyeksi dengan bentuk perbandingan panjang segitiga seperti yang ditunjukkan dalam Gambar 2.15. Berdasarkan Gambar 2.15, dapat dituliskan persamaannya seperti yang ditunjukkan dalam Persamaan (2.3) dan Persamaan (2.4).

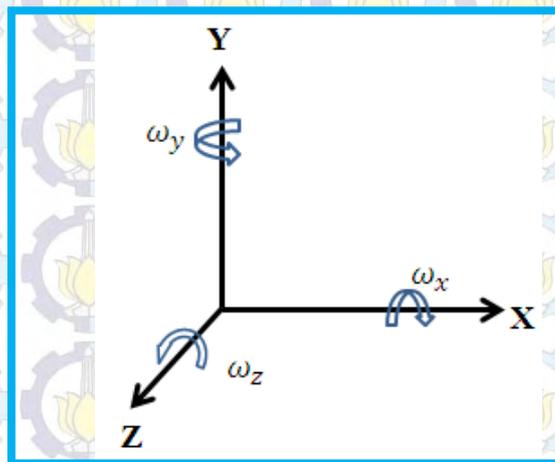
$$\frac{u}{\lambda} = \frac{x}{z} \quad (2.3)$$

$$u = \frac{x}{z} \lambda \quad (2.4)$$

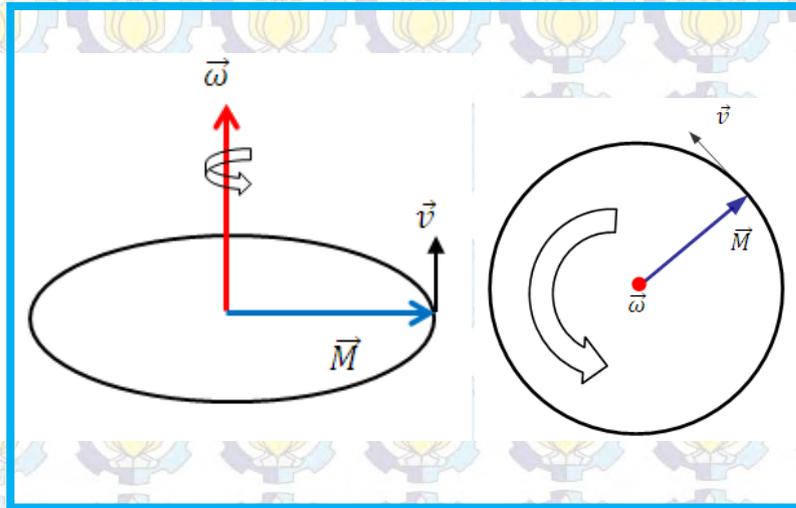
$$\frac{v}{\lambda} = \frac{y}{z} \quad (2.5)$$

$$v = \frac{y}{z} \lambda \quad (2.6)$$

Diketahui $\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$ adalah kecepatan sudut kamera yang bergerak terhadap masing-masing sumbu yaitu sumbu- x , sumbu- y , dan sumbu- z . Seperti yang ditunjukkan dalam Gambar 2.16. Jika dituliskan dalam bentuk vector seperti yang ditunjukkan dalam Gambar 2.17 hubungan antara ω dan M menghasilkan v , dimana v adalah \dot{M} atau disebut juga turunan dari M .



Gambar 2. 16. Arah Pergerakan Kecepatan Sudut



Gambar 2.17. Hubungan ω dan M dalam bentuk vektor

$$v = \omega \times M \quad (2.7)$$

$$\dot{M} = \omega \times M \quad (2.8)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.9)$$

$$\dot{x} = y\omega_z - z\omega_y \quad (2.10)$$

$$\dot{y} = z\omega_x - x\omega_z \quad (2.11)$$

$$\dot{z} = x\omega_y - y\omega_x \quad (2.12)$$

ω_z adalah kecepatan sudut pada sumbu z. Pada kamera PTZ tidak ada untuk pergerakan terhadap sumbu z jadi nilai $\omega_z = 0$. Sehingga dapat diturunkan menjadi Persamaan (2.13) hingga (2.15).

$$\dot{x} = -z\omega_y \quad (2.13)$$

$$\dot{y} = z\omega_x \quad (2.14)$$

$$\dot{z} = x\omega_y - y\omega_x \quad (2.15)$$

dengan mensubstitusikan Persamaan (2.13) dan (2.14) ke Persamaan (2.15), maka

$$\dot{z} = \frac{uz}{\lambda} \omega_y - \frac{vz}{\lambda} \omega_x \quad (2.16)$$

Jika terdapat persamaan $y = \frac{u}{v}$, maka turunannya adalah $y' = \frac{u'v - uv'}{v^2}$. Persamaan

(2.4) dan (2.5) diturunkan dengan cara tersebut, sehingga menjadi sebagai berikut,

$$\dot{u} = \lambda \frac{\dot{x}z - x\dot{z}}{z^2} \quad (2.17)$$

$$\dot{v} = \lambda \frac{\dot{y}z - y\dot{z}}{z^2} \quad (2.18)$$

Persamaan (2.13), (2.14), dan (2.15) disubstitusikan ke Persamaan (2.19) dan (2.20). Sehingga hasil persamaannya adalah sebagai berikut,

$$\dot{u} = \frac{uv}{\lambda} \omega_x - \left(\frac{\lambda^2 - u^2}{\lambda} \right) \omega_y \quad (2.19)$$

$$\dot{v} = \left(\frac{\lambda^2 + v^2}{\lambda} \right) \omega_x - \frac{uv}{\lambda} \omega_y \quad (2.20)$$

jika hasil persamaan diatas dituliskan dalam bentuk matrik, maka akan menjadi bentuk matrik 2x2 sebagai berikut,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{uv}{\lambda} & -\frac{\lambda^2 - u^2}{\lambda} \\ \frac{\lambda^2 + v^2}{\lambda} & -\frac{uv}{\lambda} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = L \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} \quad (2.22)$$

$$s = \begin{bmatrix} u \\ v \end{bmatrix}, \dot{s} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.23)$$

$$\dot{s} = L\omega \quad (2.24)$$

$$\omega = L^{-1}\dot{s} \quad (2.25)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = L^{-1} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.26)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \begin{bmatrix} \frac{uv}{\lambda} & \frac{-\lambda^2 - u^2}{\lambda} \\ \frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} \end{bmatrix}^{-1} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.27)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \begin{bmatrix} \frac{uv}{\lambda} & \frac{-\lambda^2 - u^2}{\lambda} \\ \frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} \end{bmatrix}^{-1} \begin{bmatrix} u_{act} - u_{ref} \\ v_{act} - v_{ref} \end{bmatrix} \quad (2.28)$$

berdasarkan persamaan matrik pada Persamaan 2.28, telah diketahui besar kecepatan sudut untuk pergerakan kecepatan *pan* yang dilambangkan dengan ω_y dan kecepatan *tilt* yang dilambangkan dengan ω_x . Namun dari hasil perkalian L^{-1} dan *error* koordinat, nilainya masih terlalu kecil untuk dapat menggerakkan kamera. Sehingga diperlukan gain sebesar k . Sehingga dapat ditulis dengan Persamaan (2.29).

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = k \begin{bmatrix} \frac{uv}{\lambda} & \frac{-\lambda^2 - u^2}{\lambda} \\ \frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} \end{bmatrix}^{-1} \begin{bmatrix} u_{act} - u_{ref} \\ v_{act} - v_{ref} \end{bmatrix} \quad (2.29)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \frac{k}{\lambda^2 + u^2 + v^2} \begin{bmatrix} -\frac{uv}{\lambda} & \frac{\lambda^2 + u^2}{\lambda} \\ \frac{\lambda^2 + v^2}{\lambda} & \frac{uv}{\lambda} \end{bmatrix}^{-1} \begin{bmatrix} u_{act} - u_{ref} \\ v_{act} - v_{ref} \end{bmatrix} \quad (2.30)$$

2.4 Kalman Filter

Pada dasarnya metoda kalman filter dibagi menjadi 2 step yaitu prediksi dan koreksi. Algoritma dari kalman filter dapat diketahui sebagai berikut. Langkah pertama yaitu dengan menghitung prediksi dari lokasi objek pada Persamaan (2.31), kemudian dilanjutkan dengan menghitung nilai error kovarian pada Persamaan (2.32).

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \quad (2.31)$$

$$\hat{P}_k = A\hat{P}_{k-1}A' + Q \quad (2.32)$$

\hat{P}_k merupakan estimasi dari error kovarian pada step ke- k dan \hat{P}_{k-1} adalah estimasi dari error kovarian pada step $(k-1)$. Q merupakan matrik kovarian dari *process noise*. Pada Persamaan (2.31) dan (2.32) merupakan persamaan prediksi. Setelah step prediksi dilakukan, maka dilakukan koreksi sebagai berikut,

$$K_k = \hat{P}_k H' (H \hat{P}_k H' + R)^{-1} \quad (2.33)$$

$$\hat{x}_k = \hat{x}_k + K_k (z_k - H \hat{x}_k) \quad (2.34)$$

$$P_k = (1 - K_k H) \hat{P}_k \quad (2.35)$$

dimana K_k merupakan *kalman gain* dan R adalah matriks noise kovarian dari pengukuran. Dengan menggunakan Persamaan dari (2.31) hingga (2.35) maka sistem dapat memprediksi dan mengkoreksi lokasi objek. Pada Perasamaan (2.36), (2.37) dan (2.38), nilai matrik didefinisikan sebagai berikut,

$$A = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.36)$$

$$B = \begin{bmatrix} \frac{1}{2}t^2 \\ \frac{1}{2}t^2 \\ t^2 \\ t^2 \end{bmatrix} \quad (3.37)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.38)$$

pada matriks tersebut nilai t bernilai 1. Matriks ini dihitung berdasarkan pergerakan objek yang linier dimana diperoleh dari Persamaan (2.39) dan Persamaan (3.40). Pada Persamaan (3.39) dan Persamaan (3.40) nilai percepatan merupakan *process noise*.

$$x = x_0 + vt + \frac{1}{2}at^2 \quad (3.39)$$

$$v = v_0 + at \quad (3.40)$$

2.5 Kamera PTZ

Kamera PTZ (*Pan-Tilt-Zoom*) adalah salah satu jenis kamera yang dapat bergerak secara rotasi horizontal dan vertikal serta dapat mengatur perbesaran pengambilan gambar. Dari fitur tersebut kamera ini tentunya dapat mengambil gambar dengan jangkauan yang besar pada suatu lokasi sehingga banyak difungsikan sebagai kamera keamanan untuk memantau obyek yang bergerak baik dikendalikan secara manual maupun otomatis.

Dalam penelitian ini, dipakai Kamera PTZ dengan tipe SPD-1000 yang mempunyai spesifikasi sebagai berikut,

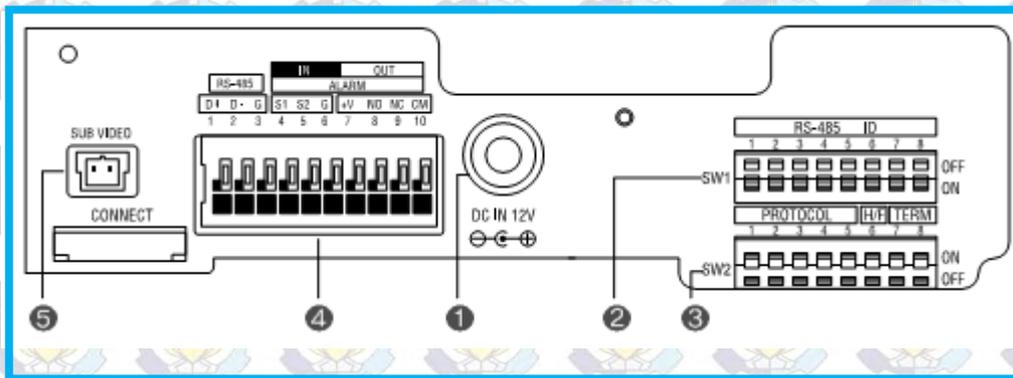
- Memiliki ukuran diameter 140 mm dan tinggi 134,8 mm.
- Dapat mengambil gambar dan merekam gambar (video).
- Mempunyai perbesaran (*zoom*) sampai 10 kali, sehingga dapat memperkecil dan memperbesar gambar dengan kejelasan yang luar biasa.
- Menyajikan kualitas gambar terbaik dengan otomatis dapat mengatur mode malam maupun siang secara otomatis dalam pengambilan gambarnya.
- Jangkauan pergerakan rotasi horizontal dan vertikal berturut-turut sebesar 350° dan $-5^\circ - 185^\circ$.
- Kecepatan pergerakan rotasi horizontal dan vertikal berturut-turut sebesar $0.5^\circ - 70^\circ / \text{detik}$ dan $0.5^\circ - 50^\circ / \text{detik}$.
- Gambar 2. 13 . Diagram blok Position-Based Visual Servo(PBVS) kontrol



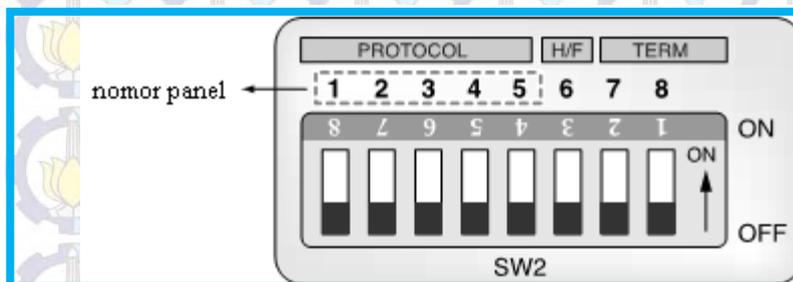
Gambar 2. 14. Kamera PTZ (Samsung, 2000).

Gambar kamera yang dipakai dalam penelitian ini ditunjukkan dalam Gambar 2.18. Kamera ini dapat diakses dengan menggunakan kartu S-Video menggunakan komunikasi RS-232 dan RS-485. Kamera dapat dikontrol dengan mengirimkan data kecepatan untuk pergerakan *pan*, *tilt*, dan *zoom*. Bagian depan panel kontrol S-Video untuk *interface* kamera PTZ dengan komputer ditunjukkan dalam Gambar 2.19. Pada gambar tersebut terdapat beberapa bagian yang dituliskan dengan angka. Berikut adalah keterangannya.

1. *Input power supply* DC 12 volt
2. *Switch* untuk mengatur nomor identitas kamera
3. Pengaturan komunikasi *protocol*, mode transmisi data, dan *switch* pengaturan penghentian kamera.
4. Koneksi terminal controller dan alarm *input / output*.
5. Video terminal.



Gambar 2. 15. Panel Depan S-Video Card kamera PTZ (Samsung, 2000)



Gambar 2. 20. Gambar panel SW2 (Samsung, 2000)



Gambar 2. 21. USB 415 PixelView

Tabel 2. 1. Tabel Konfigurasi Pengaturan *Baudrate*

Protocol	Baudrate(bps)	SW2-#1	SW2-#2	SW2-#3	SW2-#4	SW2-#5
Samsung	9600	OFF	OFF	OFF	OFF	OFF
Samsung	19200	OFF	OFF	OFF	OFF	ON
Pelco-D	2400	OFF	OFF	OFF	ON	OFF
Pelco-D	4800	OFF	OFF	OFF	ON	ON
Pelco-P	4800	OFF	OFF	ON	OFF	OFF
Pelco-P	9600	OFF	OFF	ON	OFF	ON
Samsung Elec.	9600	OFF	OFF	ON	ON	OFF
Samsung Elec.	19200	OFF	OFF	ON	ON	ON
Panasonic	9600	OFF	ON	OFF	OFF	OFF
Panasonic	9600	OFF	ON	OFF	OFF	ON
Vicon	4800	OFF	ON	OFF	ON	OFF
Vicon	9600	OFF	ON	OFF	ON	ON
AD	4800	OFF	ON	ON	OFF	OFF
Bosch	9600	OFF	ON	ON	OFF	ON
Bosch	19200	OFF	ON	ON	ON	OFF
Honeywell	9600	OFF	ON	ON	ON	ON
Honeywell	19200	ON	OFF	OFF	OFF	OFF
SUNGJIN	4800	ON	OFF	OFF	OFF	ON
Pelco-D	9600	ON	OFF	OFF	ON	OFF
Reserved						

Tabel 2. 2. Tabel Konfigurasi Pengaturan mode transmisi

	Function	ON	OFF
SW2-#6	Transmission Mode Selection	Full (DVR)	Half (SCC-3100A)

Tabel 2. 3. Tabel Konfigurasi Pengaturan mode terminasi RS-485

Location of Camera connection	SW2-#7	SW2-#8
Termination of Longest Path	ON	ON
On the path	OFF	OFF

Untuk lebih jelasnya, gambar panel depan pada Gambar 2.15 akan dijelaskan secara lebih terperinci. Pada Gambar 2.15 terdapat nomor panel 1 sampai 8. Nomor panel 1 sampai 5 digunakan untuk mengatur *baudrate* yang akan dipakai. Berikut adalah konfigurasinya yang ditunjukkan dalam Tabel 2.1 Selanjutnya, untuk pengaturan mode transmisi dan terminasi RS-485 berturut-turut ditunjukkan dalam Tabel 2.2 dan Tabel 2.3. Untuk video terminal, dikoneksikan dengan USB 415 PixelView yang kemudian disambungkan dengan perangkat USB computer. Gambar 2.16 merupakan PixelView.

Pelco-D terdiri dari 7 hexadecimal byte, yang keterangan untuk masing byte nya terlihat dalam Tabel 2.4 dibawah ini. Untuk pengaturan fitur kamera yang lain dapat dilihat pada Tabel 2.6.

Tabel 2. 4. Konfigurasi byte 1 – byte 7

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Sync	Camera Address	Command 1	Command 2	Data 1	Data 2	Checksum

Tabel 2.5. Konfigurasi *command* byte 3 dan byte 4

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command 1	Sense	Reserved	Reserved	Auto / Manual Scan	Camera On/Off	Iris Close	Iris Open	Focus Near
Command 2	Focus Far	Zoom Wide	Zoom Tele	Tilt Down	Tilt Up	Pan Left	Pan Right	Fix to 0

Tabel 2.6. Konfigurasi pengaturan lain kamera

Command	Byte 3	Byte 4	Byte 5	Byte 6
Go to Preset	00	07	00	01 sampai FF
Set Zoom Speed	00	25	00	00 sampai 33
Set Focus Speed	00	27	00	00 sampai 33

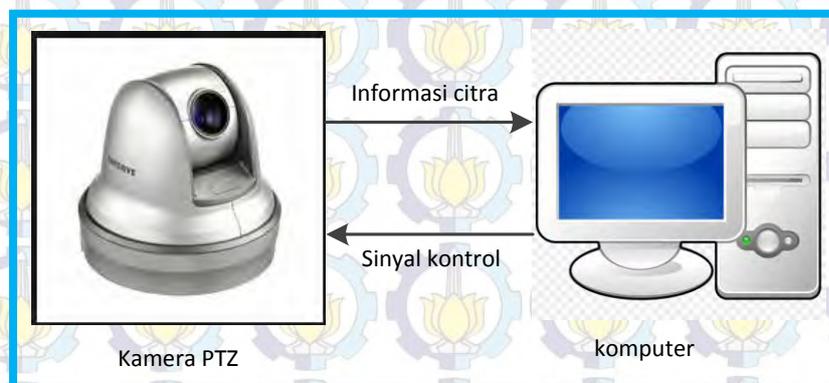
Keterangan:

- Byte 1, untuk memberikan sinyal sinkronisasi ke kamera, dimana nilai *default* nya adalah FF.
- Byte 2, merupakan alamat kamera yang dikontrol. Setiap kamera diatur alamatnya berdasarkan *dip-switch* yang terdapat pada kamera.
- Byte 3 dan 4, untuk mengatur pergerakan kamera. Keterangan lebih lanjut dapat dilihat dalam tabel 2.

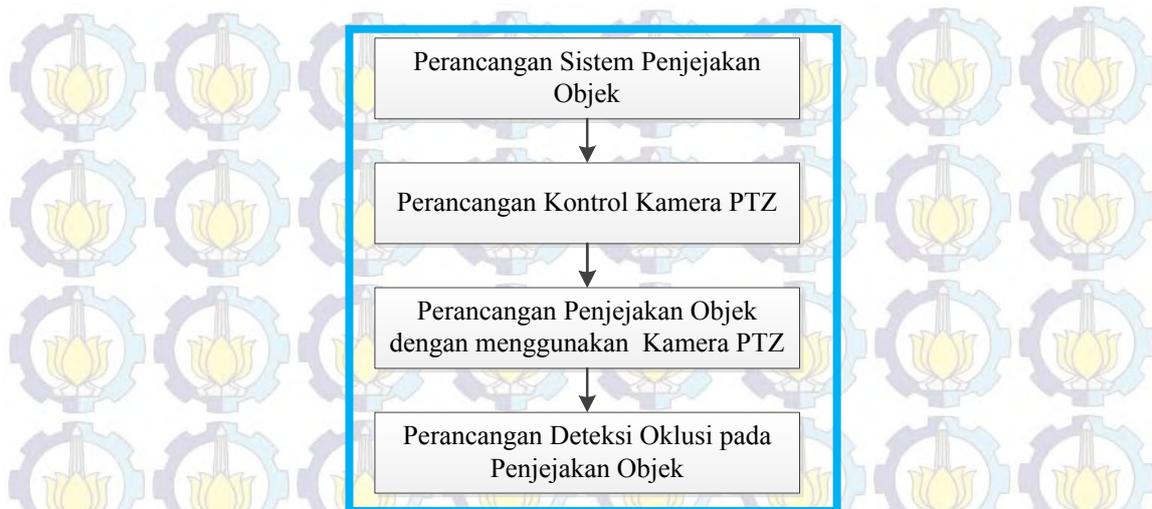
- Byte 5, digunakan untuk mengatur kecepatan pergerakan kamera ke kiri dan ke kanan (*pan speed*) dengan jangkauan nilai 00 (berhenti) sampai 3F (kecepatan tinggi) dan FF untuk kecepatan maksimum kamera.
- Byte 6, digunakan untuk mengatur kecepatan pergerakan kamera ke atas dan ke bawah (*tilt speed*) dengan jangkauan nilai 00 (berhenti) sampai 3F (kecepatan maksimum).
- Byte 7, penjumlahan byte kemudian dibagi 100.

BAB 3 METODOLOGI PENELITIAN

Pada penelitian ini telah dilakukan sistem penjejakan objek dengan menggunakan kamera PTZ. Pada Gambar 3.1 merupakan diagram blok dari sistem yang dibuat. Sistem terdiri dari kamera PTZ sebagai penangkap informasi citra. Kamera PTZ yang digunakan adalah tipe SPD-1000 dari Samsung. Kamera ini dapat berotasi horizontal sejauh 0° - 350° dan berotasi vertical sejauh -5° – 185° , dan dapat melakukan zoom sampai 10 kali, Kemudian informasi citra RGB yang diambil dari kamera PTZ ke computer dilewatkan melalui perangkat TV Tuner dengan mode *composite video*. Komputer kemudian mengolah informasi citra RGB. Dari citra RGB tersebut diubah menjadi koordinat dengan menggunakan *Improved Mean Shift*. Kemudian dari koordinat objek target tersebut difilter dengan menggunakan kalman filter untuk menghasilkan estimasi koordinat objek target. Untuk mengontrol pergerakan kamera PTZ, input yang digunakan adalah error koordinat dengan menghitung kalkulasi error koordinat. Dari error koordinat tersebut digunakan input sebagai kontrol visual servoing untuk menggerakkan kamera PTZ. Output dari kontrol visual servoing adalah sinyal kontrol berupa kecepatan pan dan tilt kamera PTZ. Kemudian sinyal kontrol tersebut dikirim dengan menggunakan Prolific USB to Serial yang diteruskan oleh perangkat RS 232 to RS 485 Converter. Dari sinyal kontrol tersebut akan digunakan untuk menggerakkan kamera PTZ.



Gambar 3. 1. Diagram blok sistem kerja

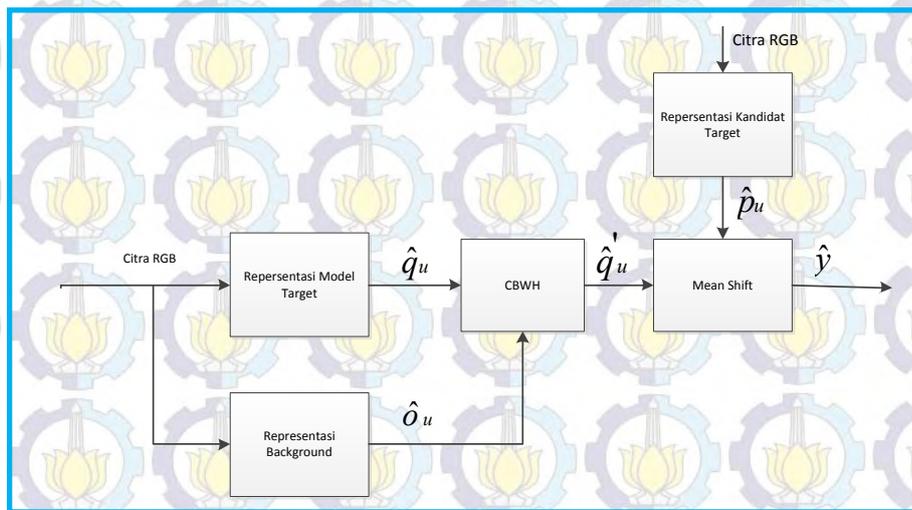


Gambar 3. 2. Tahapan perancangan metode penelitian

Tahapan perancangan metode penelitian yang dilakukan terdapat pada Gambar 3.2, yang terdiri dari perancangan sistem penjejakan objek, perancangan kontrol kamera PTZ, perancangan penjejakan objek dengan menggunakan kamera PTZ, dan perancangan penanganan oklusi pada penjejakan objek.

3.1 Perancangan Sistem Penjejakan Objek

Penjejakan objek yang baik merupakan penjejakan objek yang dapat tahan terhadap *noise background*, oklusi dan dapat window penjejakan dapat mengikuti ukuran objek. Pada penelitian ini, dikembangkan metode penjejakan objek yang dapat tahan terhadap *noise background* dan dapat melakukan penskalaan. Metode dasar yang akan dikembangkan adalah algoritma *Mean Shift*. Algoritma *Mean Shift* dipilih karena metode ini efisien digunakan dalam penjejakan objek dan tahan terhadap oklusi sebagian (*partial occlusion*). Tetapi metode tersebut tidak tahan terhadap *noise background* dan tidak dapat melakukan penskalaan. Pada penelitian ini, dikembangkan algoritma *Mean Shift* agar dapat mengurangi *noise background* dan dapat melakukan penskalaan, dan dinamakan algoritma *Improved Mean Shift*.

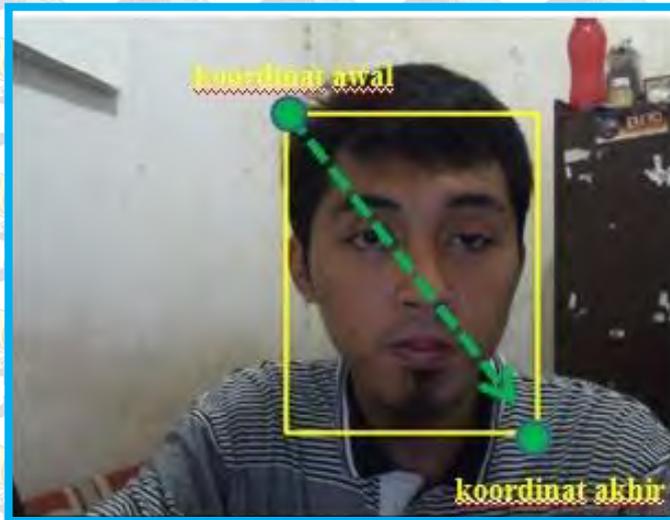


Gambar 3.3. Diagram blok sistem penjejakan objek

Algoritma Mean Shift adalah pendekatan yang efisien yang digunakan dalam penjejakan objek dimana digunakan histogram probabilitas distribusi warna sebagai fitur yang dijejak. Histogram distribusi probabilitas fitur warna pada target dijadikan sebagai representasi objek target yang akan dijejak. *Noise background* pada objek tersebut direduksi dengan menggunakan metode CBWH (*Corrected Background Weighed Histogram*). Histogram distribusi probabilitas dari fitur warna target pada frame selanjutnya dijadikan sebagai representasi kandidat target. Lokasi target pada frame selanjutnya diperoleh dari perhitungan bobot dari representasi objek target dan representasi kandidat pada frame tersebut. Pada Gambar 3.3 merupakan diagram blok sistem penjejakan yang dirancang pada penelitian ini.

3.1.1 Representasi Model Target

Representasi dari model target objek adalah histogram probabilitas warna dari objek target yang dipilih. Objek dipilih dengan mendrag mouse pada bagian visual user. Pemilihan objek dilakukan dengan cara mendrag pada bagian kiri atas objek menuju bagian bawah kanan objek, sehingga lebar dan tinggi objek dapat ditentukan. Pada Gambar 3.4 merupakan cara memilih objek, warna hijau menunjukkan arah pergerakan *mouse*, sedangkan warna kuning merupakan window target setelah objek dipilih.



Gambar 3. 4 Pemilihan objek target

Probabilitas dari warna $u = 1 \dots m$, dapat didefinisikan pada Persamaan (3.1).

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta(b(x_i^*) - u) \quad (3.1)$$

dimana m adalah jumlah warna yang digunakan. Warna yang digunakan pada penjejakan objek ini warna RGB (Red, Green, Blue) dengan masing-masing nilai dinormalisasi menjadi $16 \times 16 \times 16$. Sehingga untuk nilai $u=1$ merupakan nilai dari $R=1, G=1, B=1$, untuk nilai $u=2$ merupakan nilai dari $R=1, G=1, B=2$, dan seterusnya hingga $u = 4096$ merupakan nilai dari $R=16, G=16, B=16$. Normalisasi ini bertujuan untuk mempercepat proses perhitungan untuk mendapatkan koordinat hasil penjejakan. $\{x_i^*\}_{i=1 \dots n}$ adalah posisi koordinat piksel dari objek target. Posisi koordinat piksel ini berada pada persegi panjang yang dipilih oleh user. $b(x_i^*)$ menunjukkan nilai RGB pada piksel yang terletak pada x_i . $k(\|\cdot\|)$ merupakan fungsi kernel. Kernel yang digunakan adalah kernel Epanechnikov. Kernel Epanechnikov dapat didefinisikan dengan Persamaan (3.2).

$$k(\|\cdot\|) = \begin{cases} (1 - \|x_i\|) & \|x_i\| < 1 \\ 0 & \text{lainnya} \end{cases} \quad (3.2)$$

$\delta(a)$ adalah kroneker delta function yang dapat didefinisikan dengan Persamaan (3.3).

$$\delta(a) = \begin{cases} 1 & \text{jika } a = 0 \\ 0 & \text{lainnya} \end{cases} \quad (3.3)$$

maka dapat disimpulkan bahwa nilai probabilitas dari fitur warna dapat bernilai pada Persamaan (3.4).

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \quad \text{jika} \quad \delta(b(x_i^*)) = u \quad (3.4)$$

dimana C merupakan konstanta normalisasi yang didefinisikan pada Persamaan (3.5).

$$C = \left(\sum_{i=1}^n k(\|x_i^*\|^2) \right)^{-1} \quad (3.5)$$

3.1.2 Representasi *Background*

Representasi *background* adalah histogram distribusi probabilitas dari warna *background* yang terdapat pada target. Ukuran *background* diperoleh dari 2 kali ukuran target. Pada Gambar 3.5 ukuran *background*. Warna kuning menunjukkan ukuran objek, sedangkan warna biru menunjukkan ukuran *background*. Representasi *background* diperoleh dari histogram distribusi probabilitas fitur dari area *background*.



Gambar 3. 5. Ukuran *background*

Probabilitas dari warna $u = 1 \dots m$, dapat didefinisikan pada Persamaan (3.6).

$$\hat{\sigma}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta(b(x_i^*) - u) \quad (3.6)$$

dimana m adalah jumlah fitur warna yang digunakan. Fitur warna yang digunakan pada penjejakan objek ini warna RGB (Red, Green, Blue) dengan masing-masing nilai dinormalisasi menjadi $16 \times 16 \times 16$. Sehingga untuk nilai $u=1$ merupakan nilai dari $R=1, G=1, B=1$, untuk nilai $u=2$ merupakan nilai dari $R=1, G=1, B=2$, dan seterusnya hingga $u = 4096$ merupakan nilai dari $R=16, G=16, B=16$. Normalisasi ini bertujuan untuk mempercepat proses perhitungan untuk mendapatkan koordinat hasil penjejakan. $\{x_i^*\}_{i=1 \dots n}$ adalah posisi koordinat pixel dari background. $b(x_i^*)$ menunjukkan nilai RGB pada pixel yang terletak pada x_i . $k(\|\cdot\|)$ merupakan fungsi kernel. Kernel yang digunakan adalah kernel Epanechnikov. Kernel Epanechnikov dapat didefinisikan dengan Persamaan (3.2), maka dapat disimpulkan bahwa nilai probabilitas fitur warna pada background dapat bernilai pada Persamaan (3.7).

$$\hat{\sigma}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \quad \text{jika} \quad \delta(b(x_i^*)) = u \quad (3.7)$$

C merupakan konstanta normalisasi yang didefinisikan pada Persamaan (3.8).

$$C = \sum_{i=1}^n k(\|x_i^*\|^2)^{-1} \quad (3.8)$$

3.1.3 Corrected Background Weigthed Histogram (CWBH)

Corrected Background Weighted Histogram adalah metode yang digunakan pada *Mean Shift* yang digunakan untuk meningkatkan akurasi lokasi pada penjejakan objek. Ukuran *background* yang digunakan adalah 2 kali dari ukuran target objek. Nilai transformasi dibutuhkan mengetahui hubungan antara background dan model target. koefisien v_u pada Persamaan (3.9) merupakan nilai transformasi antara objek target dan background.

$$\left\{ V_u = \min \left(\frac{\hat{\delta}^*}{\hat{\delta}_u}, 1 \right) \right\}_{u=1..m} \quad (3.9)$$

dimana nilai koefisien v_u dibutuhkan mereduksi nilai fitur *background* yang terdapat pada model target. $\hat{\delta}^*$ adalah nilai minimal non zero dari $\{\hat{\delta}_u\}_{u=1..m}$. Probabilitas dari warna yang baru $u = 1 \dots m$, dapat diturunkan pada Persamaan (3.10).

$$\hat{q}'_u = C' v_u \sum_{i=1}^n k(\|x_i^*\|^2) \delta(b(x_i^*) - u) \quad (3.10)$$

dimana m adalah jumlah fitur warna yang digunakan. Fitur warna yang digunakan pada penjejakan objek ini warna RGB (Red, Green, Blue) dengan masing-masing nilai dinormalisasi menjadi $16 \times 16 \times 16$. $\{x_i^*\}_{i=1..n}$ adalah posisi koordinat pixel dari background. $b(x_i^*)$ menunjukkan nilai RGB pada piksel yang terletak pada x_i . $k(\|\cdot\|)$ merupakan fungsi kernel. Kernel yang digunakan adalah kernel Epanechnikov. Kernel Epanechnikov dapat didefinisikan dengan Persamaan (3.2) dimana nilai normalisasi nilai probabilitas warna pada Persamaan (3.11).

$$C' = \sum_{i=1}^n k(\|x_i^*\|^2)^{-1} \quad (3.11)$$

3.1.4 Representasi Kandidat Target

Representasi dari kandidat target objek adalah histogram probabilitas objek target yang berada pada frame selanjutnya. Ukuran kandidat target berubah-ubah sesuai dengan ukuran target objek, dimana ukuran awal kandidat target sama dengan ukuran target model. Probabilitas fitur kandidat target objek, $u = 1 \dots m$, dapat didefinisikan pada Persamaan (3.12).

$$\hat{p}_u = C_h \sum_{i=1}^n k(\|x_i^*\|^2) \delta(b(x_i^*) - u) \quad (3.12)$$

dimana m adalah jumlah fitur warna yang digunakan. Fitur warna yang digunakan pada penjejakan objek ini warna RGB (Red, Green, Blue) dengan masing-masing nilai dinormalisasi menjadi $16 \times 16 \times 16$. $\{x_i^*\}_{i=1 \dots n}$ adalah posisi koordinat pixel dari kandidat target. $b(x_i^*)$ menunjukkan nilai RGB pada pixel yang terletak pada x_i . $k(\|\cdot\|)$ merupakan fungsi kernel. Kernel yang digunakan adalah kernel Epanechnikov. Kernel Epanechnikov dapat didefinisikan dengan Persamaan (3.2) dimana C_h merupakan nilai normalisasi yang dapat didefinisikan pada Persamaan (3.13).

$$C_h = \sum_{i=1}^n k(\|x_i^*\|^2)^{-1} \quad (3.13)$$

3.1.5 Mean Shift

Lokasi dari koordinat yang baru y_1 , dapat ditemukan dengan mengabungkan representasi model target dan representasi kandidat target objek. Koordinat yang baru dapat didefinisikan pada Persamaan (3.14).

$$y_1 = \frac{\sum_{i=1}^n x_i \omega_i g(\|y_0 - x_i\|)^2}{\sum_{i=1}^{n_h} \omega_i g(\|y_0 - x_i\|)^2} \quad (3.14)$$

dimana $\{x_i\}_{i=1..n}$ adalah posisi koordinat pixel dari kandidat target. $g(\|\cdot\|)$ merupakan fungsi kernel. Kernel yang digunakan adalah kernel Epanechnikov. Kernel Epanechnikov dapat didefinisikan dengan Persamaan (3.2). \hat{y}_0 merupakan posisi koordinat lama dari objek. Bobot pixel ω_i , dapat didefinisikan dengan Persamaan (3.15).

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta(b(x_i^*) - u) \quad (3.15)$$

dimana m adalah jumlah fitur warna yang digunakan. $b(x_i^*)$ menunjukkan nilai RGB pada pixel yang terletak pada x_i . u adalah warna yang digunakan. \hat{q}_u merupakan distribusi probabilitas model target, \hat{p}_u merupakan distribusi probabilitas kandidat target. $\delta(a)$ adalah fungsi kroneker delta.

3.1.6 Metode Penskalaan

Penskalaan adaptif digunakan untuk menyesuaikan ukuran window terhadap ukuran target objek. Metode Mean Shift tidak dapat digunakan untuk penskalaan terhadap ukuran objek, sehingga ukuran objek tetap. Pada penelitian ini, Mean Shift dikembangkan agar dapat adaptif terhadap ukuran objek. Metode penskalaan menggunakan *zeroth order moment*, *second order moment* dan koefisien Bhattacharyya antara model target dan kandidat target untuk mengestimasi lebar dan tinggi objek. *Zeroth order moment* adalah jumlah semua bobot piksel pada kandidat target. *Zeroth order moment* digunakan untuk mengestimasi area target yang didefinisikan pada Persamaan (3.16).

$$M_{00} = \sum_{i=0}^n \omega(x_i) \quad (3.16)$$

Pada SOAMST, koefisien Bhattacharyya digunakan untuk menyesuaikan *zeroth order moment* untuk mengestimasi area target karena jika hanya menggunakan *zeroth order moment*, maka nilai error estimasi akan membesar. Area target yang diestimasi dapat didefinisikan pada Persamaan (3.17).

$$A = c(\rho)M_{00} \quad (3.17)$$

dimana,

$$c(\rho) = \exp\left(\frac{\rho-1}{\sigma}\right) \quad (3.18)$$

M_{00} adalah *zeroth order moment*, ρ adalah koefisien Bhattacharyya, σ adalah *corrected area*, pada penelitian ini $\sigma = 2$. Koefisien Bhattacharyya dapat didefinisikan pada Persamaan (3.19).

$$\rho = \sum_{i=1}^m \sqrt{\hat{q}_u \hat{p}_u} \quad (3.19)$$

Fitur moment dapat diperoleh dengan menggunakan *zeroth order momen*, dengan menggunakan Persamaan (3.20) hingga (3.24).

$$M_{10} = \sum_{i=0}^n \omega x_{i,1} \quad (3.20)$$

$$M_{01} = \sum_{i=0}^n \omega x_{i,2} \quad (3.21)$$

$$M_{11} = \sum_{i=0}^n \omega x_{i,1} x_{i,2} \quad (3.22)$$

$$M_{20} = \sum_{i=0}^n \omega x_{i,1}^2 \quad (3.23)$$

$$M_{02} = \sum_{i=0}^n \omega x_{i,2}^2 \quad (3.24)$$

dimana $(x_{i,1}, x_{i,2})$ adalah koordinat dari piksel i pada kandidat target. ω_i adalah bobot piksel. *First order moment* dapat diperoleh dengan menggunakan Persamaan (3.16), (3.20) dan (3.21). *First order moment* dapat didefinisikan pada Persamaan (3.25).

$$y = (\bar{x}_1, \bar{x}_2) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \quad (3.25)$$

Second order moment dapat diturunkan dari Persamaan (3.16), (3.22), (3.23), (3.24) dan (3.25). *Second order moment* dapat didefinisikan pada Persamaan (3.26), (3.27) dan (3.28).

$$\mu_{20} = \frac{M_{20}}{M_{00}} - \bar{x}_1^2 \quad (3.26)$$

$$\mu_{02} = \frac{M_{02}}{M_{00}} - \bar{x}_2^2 \quad (3.27)$$

$$\mu_{11} = \frac{M_{02}}{M_{00}} - \bar{x}_1 \bar{x}_2 \quad (3.28)$$

Matriks kovarian dapat didefinisikan pada Persamaan (3.29) dengan menggunakan Persamaan (3.26), (3.27) dan (3.28).

$$\text{COV} = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad (3.29)$$

Matrik kovarian dapat didekomposisi dengan menggunakan *Singular Value Decomposition* (SVD) dengan menggunakan Persamaan (3.30).

$$\text{cov} = U \times S \times U^T \quad (3.30)$$

dimana,

$$U = \begin{bmatrix} \mu_{21} & \mu_{11} \\ \mu_{11} & \mu_{12} \end{bmatrix} \quad (3.31)$$

$$S = \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix} \quad (3.32)$$

dimana λ_1 dan λ_2 adalah *eigenvalues* dari matriks. Pada CAMSHIFT, *eigenvalue* digunakan untuk menentukan lebar dan tinggi suatu objek. Lebar dan tinggi objek didefinisikan dengan p dan l . Kemudian λ_1 dan λ_2 digunakan untuk mengestimasi nilai p dan l . Sehingga nilai $p = k\lambda_1$ dan $l = k\lambda_2$ Sehingga estimasi area dapat didefinisikan pada Persamaan (3.33).

$$A = pl = k^2 \lambda_1 \lambda_2 \quad (3.33)$$

dimana nilai faktor skala dapat didefinisikan pada Persamaan (3.34).

$$k = \sqrt{\frac{A}{\lambda_1 \lambda_2}} \quad (3.34)$$

sehingga lebar dan tinggi object dapat diestimasi pada Persamaan (3.35) dan (3.36).

$$p = \sqrt{\frac{A}{\lambda_2}} \quad (3.35)$$

$$l = \sqrt{\frac{A}{\lambda_1}} \quad (3.36)$$

3.1.7 Algoritma Improved Mean Shift

Algoritma dari penjejakan objek visual dengan menggunakan *Improved Mean Shift* dapat disimpulkan sebagai berikut,

1. menghitung probabilitas dari model target dengan Persamaan (3.1),
2. menghitung probabilitas dari *background* dengan Persamaan (3.7),
3. menghitung probabilitas dari model target yang baru dengan Persamaan (3.10),
4. $k=0$,
5. menghitung probabilitas dari kandidat target dengan Persamaan (3.12),
6. menghitung bobot piksel dengan menggunakan Persamaan (3.15),
7. menghitung posisi baru pada wilayah kandidat target dengan menggunakan Persamaan (3.14),
8. menghitung estimasi lebar dan tinggi objek dengan menggunakan Persamaan (3.35) dan (3.36),
9. menghitung $d = \|y_1 - y_0\|$, $y_1 = y_0$, $k = k + 1$. Berikan nilai threshold error ε (nilai $\varepsilon : 0.1$), N sebagai nilai maksimum iterasi,
10. Jika $d < \varepsilon$ or $k > N$,

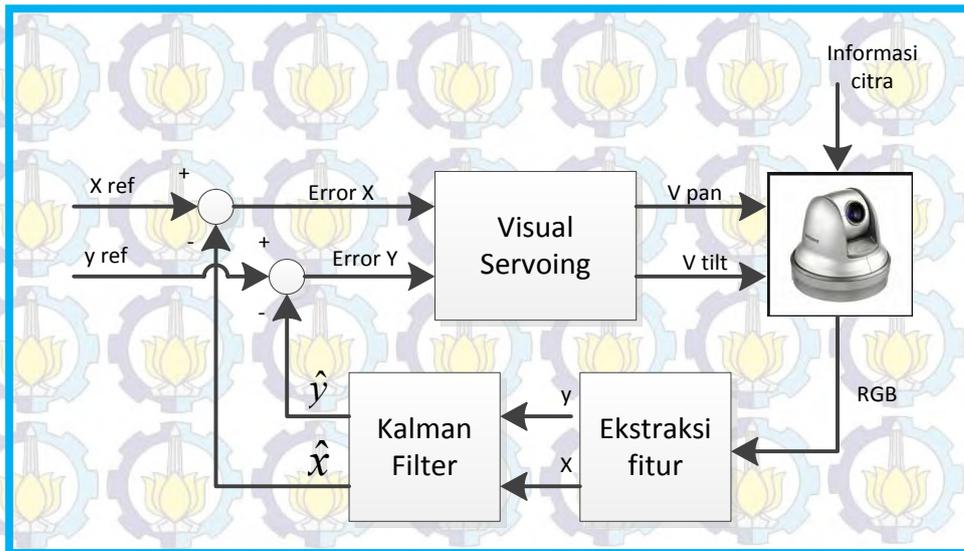
Iterasi berhenti

Lainnya

Lanjut ke langkah ke-2.

3.2 Perancangan Sistem Kontrol Kamera PTZ

Kontrol yang digunakan pada penelitian ini adalah *visual servoing* dengan struktur *Image Based Visual Servoing* (IBVS). Kontrol pergerakan kamera PTZ digunakan Struktur IBVS dipilih karena tidak diperlukan pemodalanan koordinat 3D. Kontrol *visual servoing* digunakan untuk mengontrol pergerakan kamera secara vertical (tilt) dan horizontal (pan). Diagram blok kontrol pergerakan kamera secara vertikal dan horizontal ditunjukkan pada Gambar 3.6.



Gambar 3. 6. Blok diagram kontrol kamera PTZ

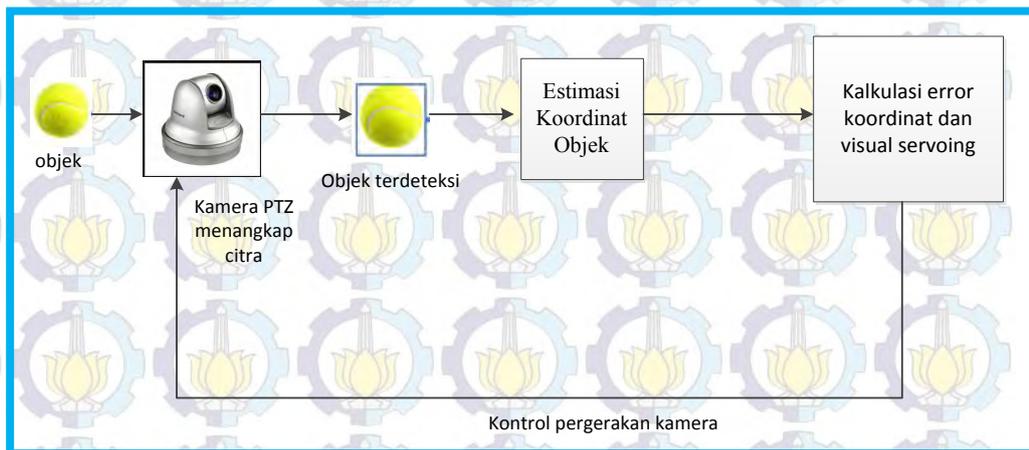
Input dari kontrol *visual servoing* adalah error x dan error y. Error koordinat x dan y diperoleh dari selisih dari koordinat referensi x dan y dengan hasil dari kalman filter koordinat x dan y. Koordinat referensi x dan y merupakan koordinat titik tengah pada tampilan visual user. Output dari visual servoing adalah kecepatan pan dan tilt yang digunakan untuk menggerakkan kamera PTZ. Kamera PTZ akan menangkap informasi citra yang kemudian akan konversi menjadi nilai RGB. Nilai RGB diekstrak fitur untuk mendapatkan nilai koordinat x dan y. Metode ekstraksi fitur yang digunakan adalah *improved Mean Shift* yang telah dijelaskan pada sebelumnya. Hasil dari ekstraksi fitur diinputkan ke filter digital. Filter digital yang digunakan adalah kalman filter. Filter digital ini digunakan untuk mengurangi *noise* hasil ekstraksi fitur.

3.3 Perancangan Sistem Penjejakan Objek dengan kamera PTZ

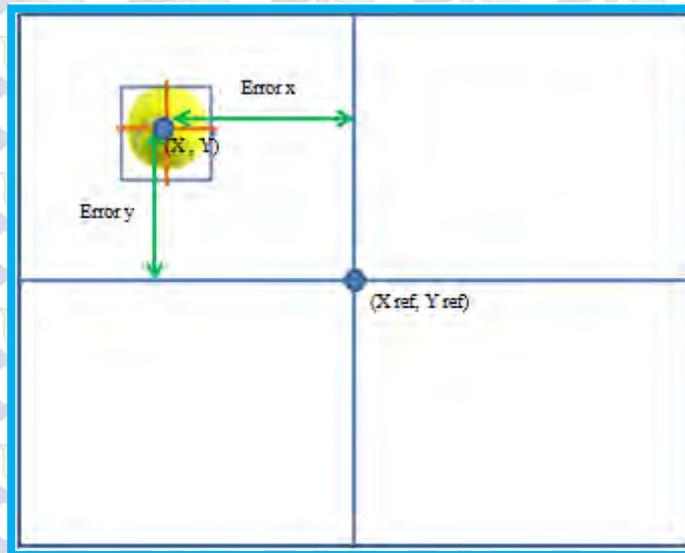
Pada Gambar 3.12 merupakan ilustrasi penjejakan objek yang merupakan penggabungan antara pendeteksian obyek, estimasi pergerakan objek dan kontrol pergerakan kamera untuk membangun sistem penjejakan menggunakan kamera PTZ. Pada Gambar 3.12 dijelaskan bahwa kamera melakukan *capture* obyek

yang kemudian obyek tersebut terdeteksi ukuran dan koordinat posisinya menggunakan *Impoved mean shift*. Ukuran dan koordinat posisi obyek tersebut dijadikan sebagai informasi untuk mengestimasi koordinat objek ketika ada oklusi dengan menggunakan metode kalman filter. Setelah itu, koordinat dan ukuran hasil dari kalman filter menjadi masukan untuk kontrol *visual servoing* untuk di hitung berapa nilai *error* nya. Selanjutnya, nilai *error* digunakan untuk kontrol pergerakan kamera PTZ sampai mencapai *set point*.

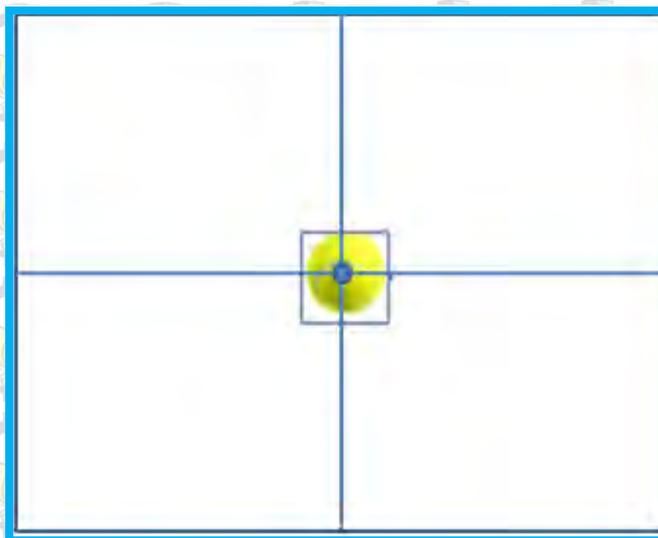
Nilai set point posisi adalah berada pada titik tengah tampilan visual user. Pada tampilan visual digunakan ukuran 400×320 , sehingga titik tengah koordinat beraada pada $x = 200$ dan $y = 160$. Nilai set point merupakan nilai titik tengah koordinat pada tampilan *visual* user. Pada penelitian ini digunakan ukuran frame adalah 400×320 , sehingga nilai koordinat x referensi adalah 200 dan nilai koordinat y referensi adalah 160. Ilustrasi posisi titik referensi objek terdapat pada Gambar 3.13. Pada Gambar 3.13 menunjukkan adanya error koordinat x dan koordinat y. Dari posisi awal yang ditunjukkan pada Gambar 3.13, sistem penjejakan objek akan mengarahkan objek hingga menuju posisi akhir yang ditunjukkan pada Gambar 3.14.



Gambar 3. 12. Ilustrasi sistem kerja pada penjejakan objek dengan menggunakan kamera PTZ



Gambar 3. 13. Posisi awal objek



Gambar 3. 14. Posisi akhir objek

3.4 Perancangan Deteksi Oklusi pada Penjejukan Objek

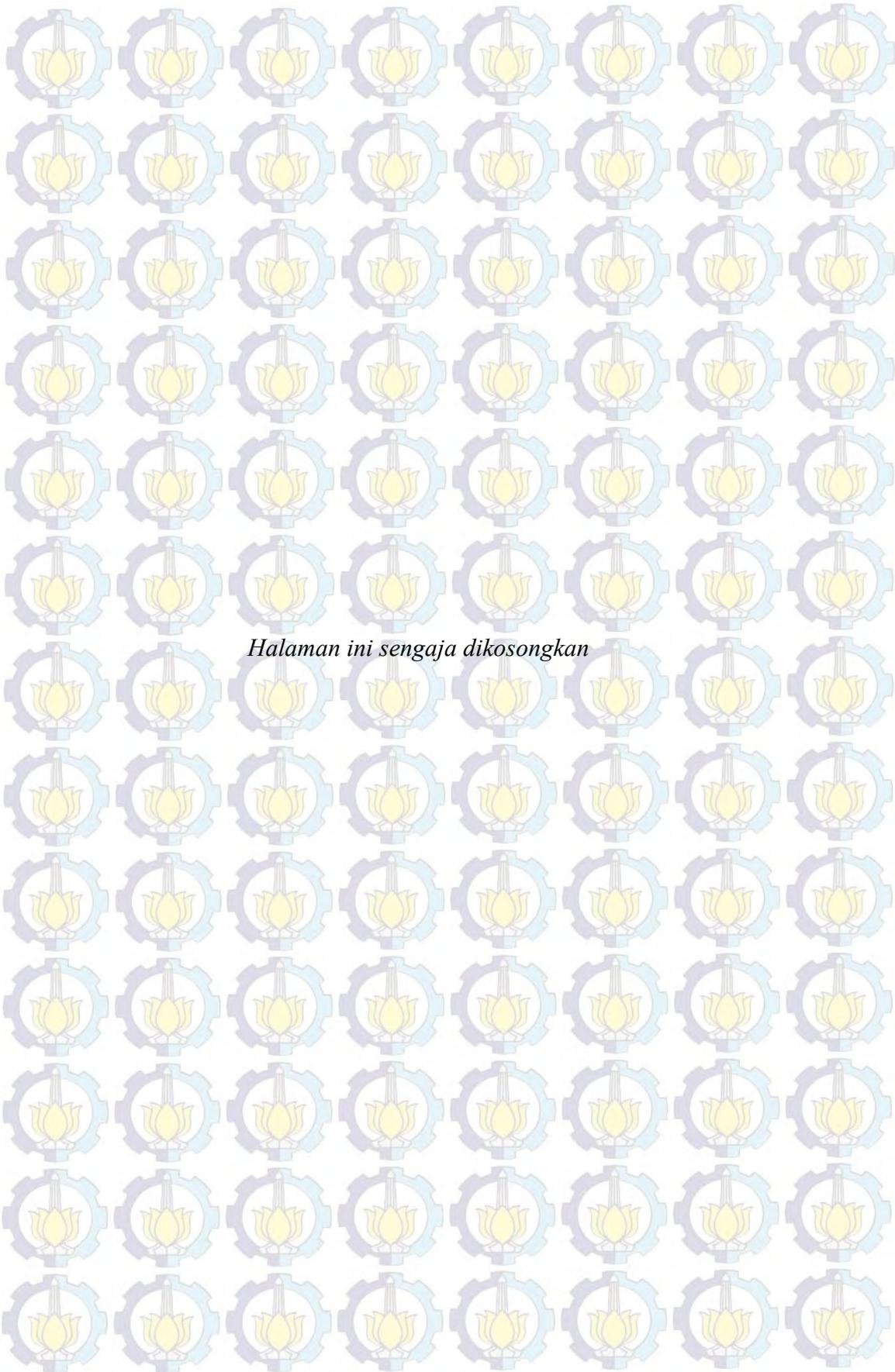
Pada penelitian ini digunakan dua parameter tambahan untuk mendeteksi adanya oklusi, yaitu koefisien Bhattacharyya dan Normalized Cross Correlation. Koefisien Bhattacharyya merupakan koefisien yang menunjukkan kesamaan dari objek target dan objek kandidat. Koefisien Bhattacharyya dapat didefinisikan pada Persamaan 3.37.

$$\rho = \sum_{i=1}^m \sqrt{\hat{q}_u \hat{p}_u} \quad (3.37)$$

dimana \hat{q}_u adalah probabilitas distribusi warna dari target model, sedangkan \hat{p}_u adalah probabilitas distribusi warna dari target kandidat. Objek dinyatakan oklusi ketika nilai koefisien bhattacharrya berada dibawah nilai threshold yaitu 0.2. Normalized Cross Correlation menunjukkan kesamaan intensitas dari dua buah gambar. Pada penelitian ini , gambar yang digunakan adalah target dan kandidat objek. NCC dapat didefinisikan dengan menggunakan Persamaan (3.38).

$$NCC = \frac{\sum_{i=1}^n (f_i - \bar{f})(t_i - \bar{t})}{\sqrt{\sum_{i=1}^n (f_i - \bar{f})^2 (t_i - \bar{t})^2}} \quad (3.38)$$

dimana f_i adalah intensitas target model, \bar{f} adalah mean dari f_i dari target, t_i adalah intensitas dari target kandidat dan \bar{t} menunjukkan mean dari t_i pada target kandidat. Objek dinyatakan oklusi ketika nilai NCC berada dibawah nilai threshold yaitu 0.1. Ketika terjadi oklusi maka window objek akan melebar. Hal ini berfungsi ketika terjadi oklusi maka sistem penjejakan akan berhenti, dan ukuran window melebar berfungsi untuk mencari target hingga terdeteksi kembali.



Halaman ini sengaja dikosongkan

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

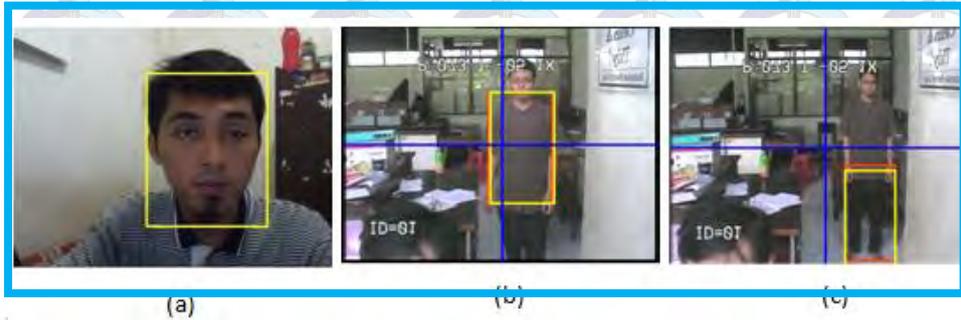
Pada bab ini akan dibahas mengenai hasil pengujian sistem yang dibangun berdasarkan perancangna yang telah dibangun pada Bab sebelumnya. Pengujian dibagi sebagai berikut.

- Pengujian Representasi Objek
- Pengujian Pengaruh Objek terhadap *Background*
- Pengujian Penjejakan Objek
- Pengujian Penskalaan Objek
- Pengujian Kecepatan Metode Penjejakan Objek
- Pengujian Ketahanan Metode Penjejakan Objek terhadap oklusi
- Pengujian kontrol Pan –Tilt
- Pengujian Objek statis
- Pengujian Objek bergerak
- Pengujian Penjejakan Objek dengan oklusi
- Pengujian Pengaruh Cahaya terhadap Objek Target
- Pengujian Kecepatan penjejakan Objek dengan menggunakan kamera PTZ
- Pengujian dengan Penjejakan Objek dengan menggunakan Objek yang Berbeda.
- Pengujian Penjejakan Objek dengan menggunakan Warna Latar yang Heterogen

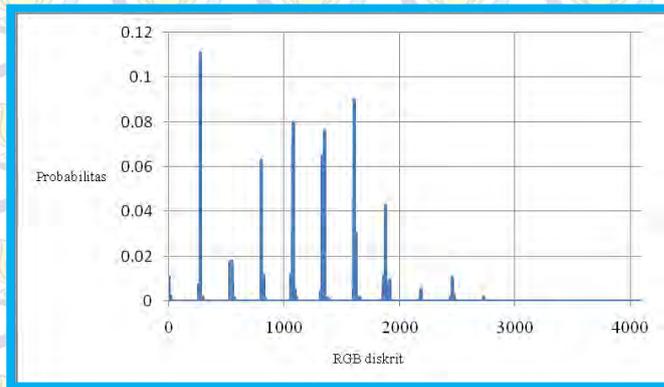
4.1 Pengujian Representasi Objek

Pada pengujian deteksi objek bertujuan untuk mengetahui apakah sistem dapat mengenali objek yang berbeda-beda. Pengujian dilakukan dengan dengan *mendrag* pada tampilan *user* untuk memilih objek yang berbeda-beda yang ditangkap oleh kamera kemudian diuji apakah nilai distribusi probabilitas warna \hat{q}_u berbeda pada masing- masing objek. Pada Gambar 4.1 merupakan hasil deteksi objek pada wajah, badan dan kaki. Warna kuning pada gambar merupakan area deteksi objek yang dipilih.

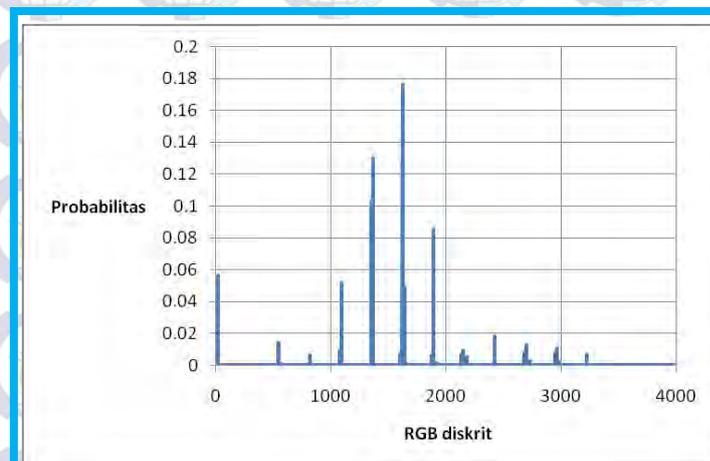
Dari area deteksi tersebut, dapat diperoleh nilai histogram probabilitas pada masing – masing objek.



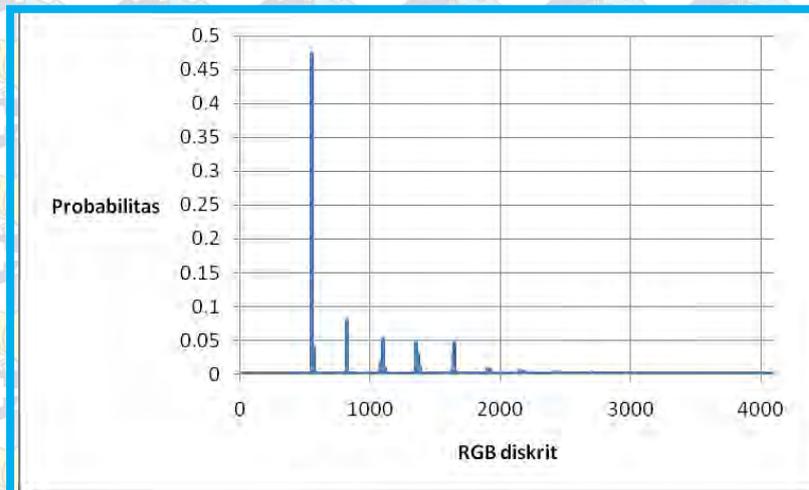
Gambar 4.1. Hasil pengujian deteksi objek, (a) target berupa wajah, (b) target berupa badan objek, (c) target berupa kaki.



Gambar 4. 2. Histogram probabilitas warna pada wajah



Gambar 4. 3. Histogram probabilitas warna pada badan



Gambar 4. 4. Histogram probabilitas warna pada kaki

Pada Gambar 4.2 hingga 4.4 merupakan histogram probabilitas distribusi warna pada masing – masing objek. Dari masing – masing histogram dapat diketahui bahwa setiap benda memiliki histogram yang berbeda – beda, sehingga dapat digunakan untuk deteksi objek berdasarkan probabilitas warna pada masing-masing objek tersebut.

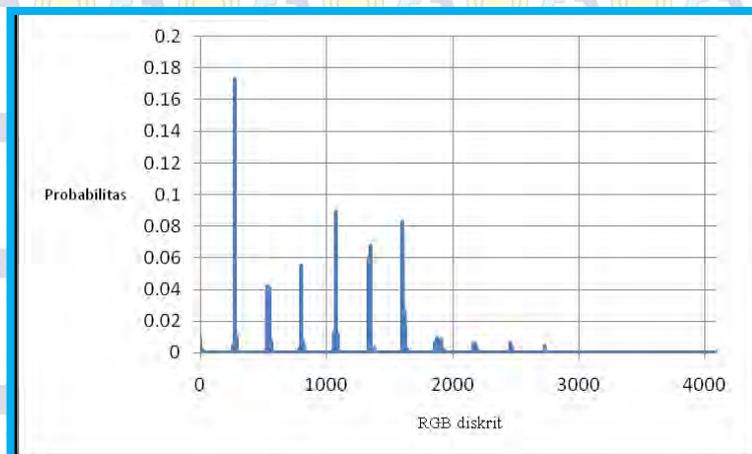
4.2 Pengujian ketahanan terhadap *noise Background*

Pada pengujian ini bertujuan untuk mengetahui ketahanan objek terhadap *background*. Pengujian dilakukan dengan dengan mendrag pada tampilan *user* untuk memilih objek yang ditangkap oleh kamera kemudian membandingkan nilai metode *Improved Mean Shift* dengan metode *Mean Shift*. Pada Gambar 4.5 merupakan objek yang dipilih merupakan wajah, sedangkan ukuran *background* adalah 2 kali ukuran target objek. Warna kuning pada Gambar 4.5 merupakan target objek, sedangkan warna biru merupakan area *background*. Pada Gambar 4.6 merupakan histogram distribusi probabilitas warna pada objek target wajah yang merupakan metode original *Mean Shift*. Pada Gambar 4.7 merupakan histogram distribusi probabilitas warna pada *background*. Dari kedua histogram tersebut dapat diketahui bahwa fitur pada objek ada yang sama dengan fitur *background* , sehingga *background* akan mempengaruhi dari target objek tersebut ketika dijejak. Pada Gambar 4.8 merupakan histogram probabilitas hasil

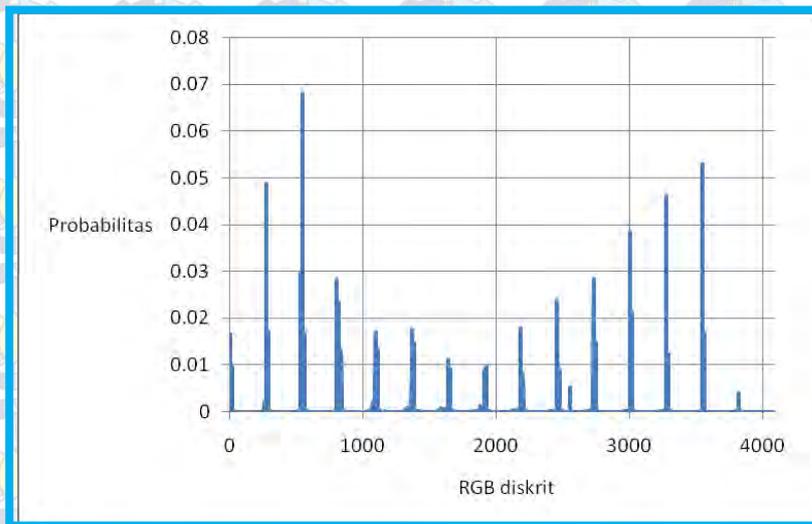
Corrected Background Weighed Histogram (CBWH). Dari Gambar 4.8, dapat diketahui bahwa nilai fitur tidak ada yang sama dengan nilai *background* sehingga target model tidak dipengaruhi lagi oleh *background*.



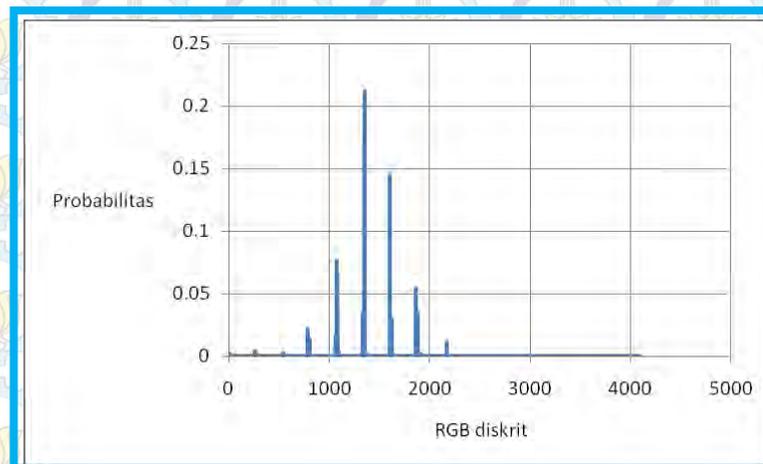
Gambar 4. 5. Objek target dan *background* yang dipilih



Gambar 4. 6. Histogram probabilitas warna pada objek target wajah



Gambar 4. 7. Histogram probabilitas warna pada *background*



Gambar 4.8. Histogram probabilitas warna hasil CBWH

4.3 Pengujian Penjejakan Objek

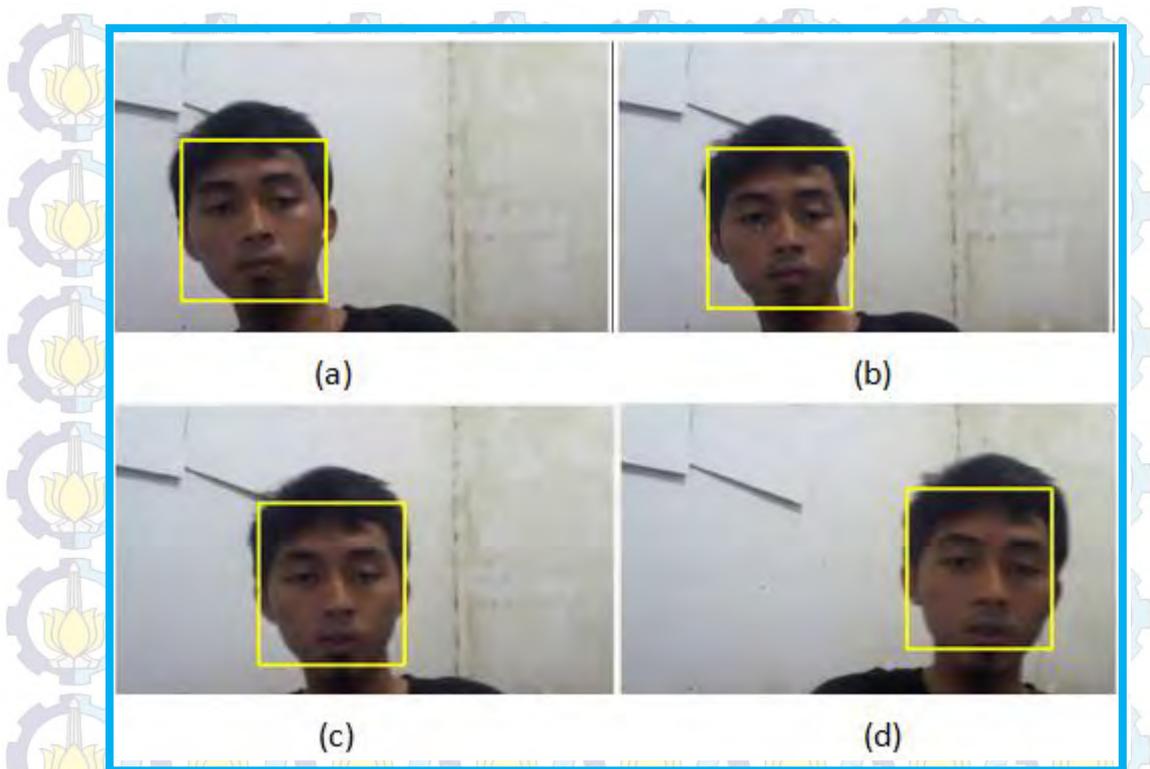
Pengujian Penjejakan objek ini bertujuan untuk menguji tingkat akurasi metode pada saat melakukan penjejakan objek. Pada pengujian ini, dibandingkan antara dua metode yaitu original *Mean Shift* dan *Improved Mean Shift*. Pengujian ini dilakukan dengan me-load video pengujian. Video pengujian yang digunakan ada dua jenis, yang pertama diuji dengan objek target yang memiliki fitur berbeda dengan background dan yang kedua pengujian dengan fitur objek target yang memiliki fitur yang sama dengan background. Objek target yang digunakan

adalah wajah, dengan background pada pengujian pertama berupa tembok berwarna putih, sedangkan pada pengujian yang kedua berupa tembok berwarna coklat yang memiliki fitur warna yang sama dengan objek yang dipilih (wajah). Pada pengujian ini, pemilihan objek (wajah) dilakukan dengan men-*drag* objek target pada tampilan user, kemudian nilai koordinat per frame disimpan dalam format **txt*. Pada pengujian pertama terdapat 160 frame. Untuk mengetahui tingkat akurasi antara dua metode tersebut maka dapat diketahui dari error yang diperoleh dari penjejakan objek yang dilakukan. Error koordinat diperoleh dengan menggunakan *euclidean distance* pada Persamaan 4.1.

$$e = \sqrt{(x_d - x)^2 + (y_d - y)^2} \quad (4.1)$$

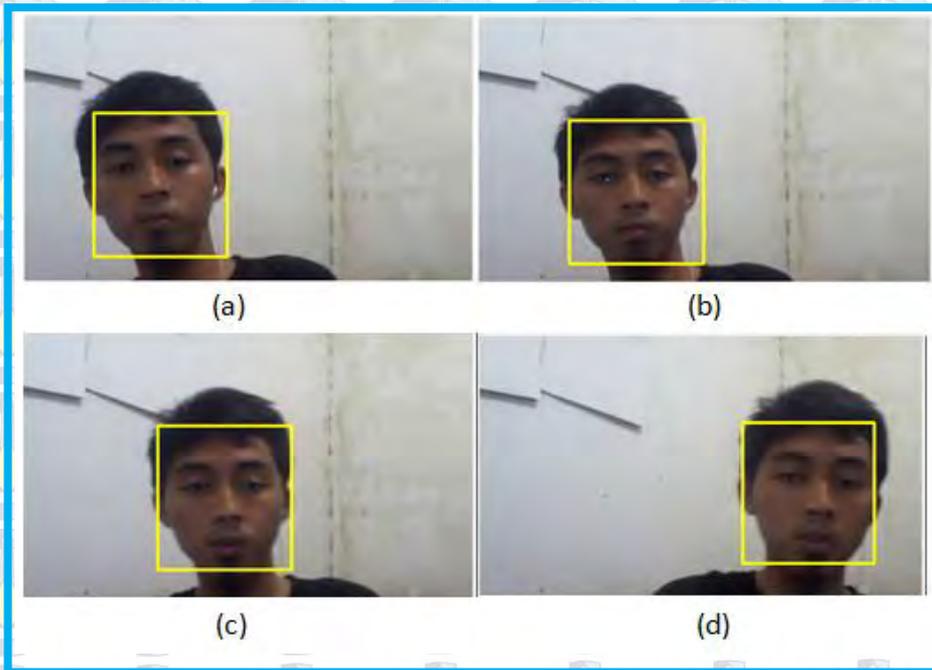
dimana e merupakan error koordinat, x_d merupakan nilai koordinat hasil penjejakan pada sumbu x , x merupakan data koordinat objek pada sumbu x , y_d merupakan nilai koordinat hasil penjejakan pada sumbu y , y merupakan data koordinat objek pada sumbu y .

Pada Gambar 4.9 merupakan hasil penjejakan objek dengan menggunakan metode *Mean Shift*, sedangkan pada Gambar 4.10 merupakan hasil penjejakan objek dengan menggunakan *Improved Mean Shift*. Dari Gambar 4.9 dan 4.10 dapat diketahui bahwa metode *Mean Shift* dan *Improved Mean Shift* masing-masing dapat melakukan penjejakan wajah dengan baik. Hasil penjejakan wajah berupa persegi panjang berwarna kuning. Pada Gambar 4.11 merupakan grafik error penjejakan objek wajah dengan menggunakan *Mean Shift* dan *Improved Mean Shift*. Untuk grafik yang berwarna merah menunjukkan error penjejakan objek dengan menggunakan *Mean Shift*, sedangkan grafik yang berwarna biru menunjukkan error penjejakan objek dengan menggunakan *Improved Mean Shift*.

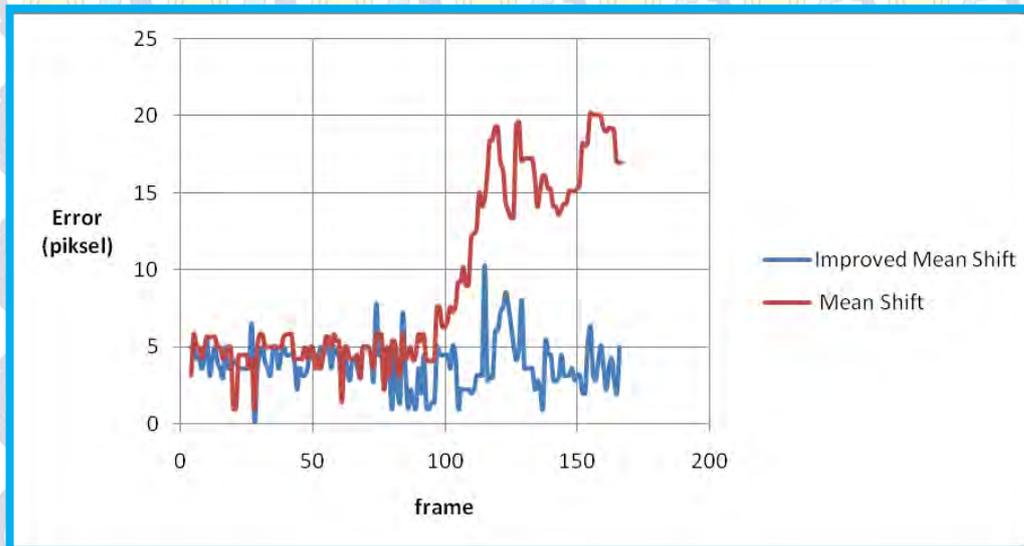


Gambar 4. 9. Hasil pengujian penjejakan objek dengan metode *Mean Shift*, (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150

Dari Gambar 4.11 dapat diketahui nilai error rata – rata dengan menggunakan metode *Mean Shift* adalah 9.01 piksel dengan standard deviasi 5.77 piksel, sedangkan dengan menggunakan metode *Improved Mean Shift* dapat diketahui bahwa nilai error rata – rata 3.87 piksel dengan standard deviasi 1.54 piksel. Dari error rata-rata tersebut dapat diketahui bahwa nilai error dengan menggunakan *Improved Mean Shift* lebih kecil dibandingkan dengan menggunakan metode *Mean Shift*.



Gambar 4. 10. Hasil pengujian penjejakan objek dengan metode *Improved Mean Shift*, (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150

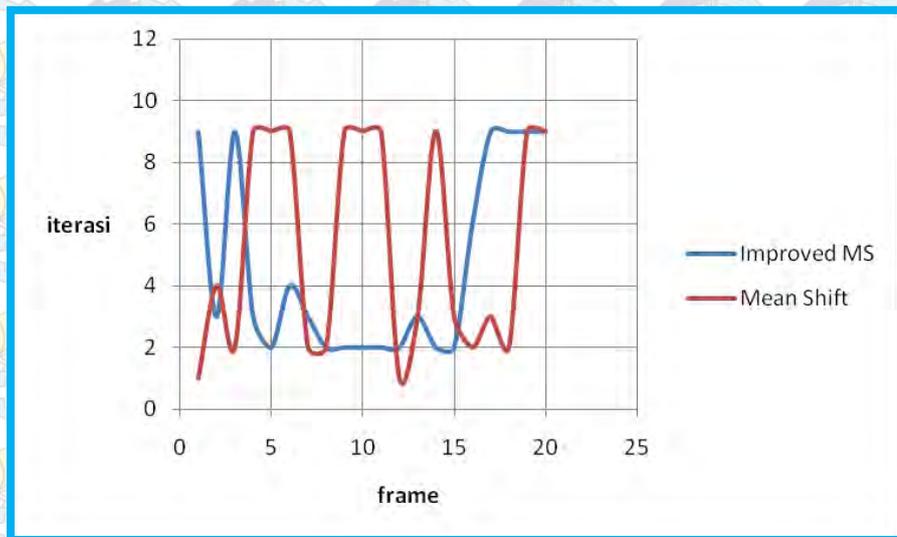


Gambar 4.11. Grafik error penjejakan objek wajah dengan menggunakan *Mean Shift* dan *Improved Mean Shift*

Tabel 4. 1. Tabel Konfigurasi hasil pengukuran waktu

frame ke-	waktu (detik)	
	Mean Shift	Improved Mean Shift
1	0.655	0.405
2	0.706	0.484
3	0.765	0.562
4	0.888	0.639
5	0.97	0.707
6	1.032	0.791
7	1.098	0.881
8	1.184	0.925
9	1.243	1.016
10	1.362	1.11
11	1.485	1.199
12	1.61	1.291
13	1.675	1.394
14	1.753	1.489
15	1.822	1.59
16	1.891	1.682
17	1.972	1.773
18	2.121	1.869
19	2.264	1.917
20	2.361	2.015

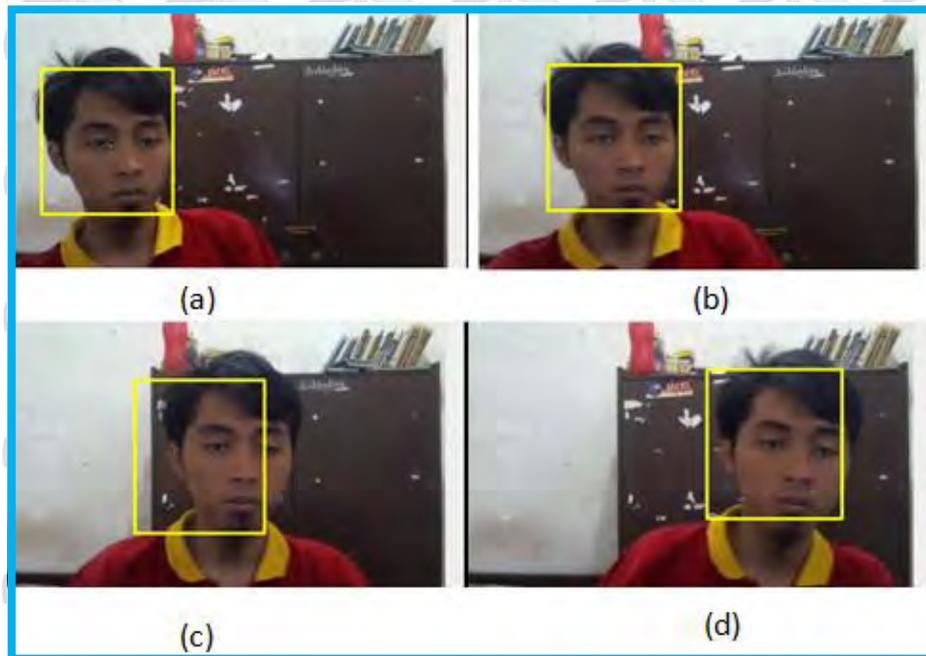
Pada Tabel 4.1 merupakan pengukuran waktu yang dilakukan dalam 20 frame pada masing- masing metode. Pada Tabel 4.1 dapat diketahui bahwa waktu yang ditempuh dengan menggunakan metode Mean Shift lebih lama dibandingkan dengan *Improved Mean Shift*, hal ini terbukti pada frame 20, metode penjejakan objek dengan menggunakan *Mean Shift* membutuhkan waktu 2.361 detik , sedangkan dengan menggunakan metode *Improved Mean Shift* membutuhkan waktu 2.015 detik. Hal ini dapat terjadi karena pada metode *Mean Shift* memerlukan waktu iterasi yang lebih banyak dibandingkan dengan menggunakan *Improved Mean Shift*, karena pada metode *Mean Shift* masih terpengaruh oleh *background* sehingga membutuhkan iterasi yang lama untuk menentukan error minimum dari hasil perhitungan *Mean Shift*. Berbeda dengan *Mean Shift*, metode *Improved Mean Shift* dimana memiliki fitur *background* yang sudah tereduksi, memiliki iterasi yang lebih rendah, sehingga waktu yang dibutuhkan untuk menentukan lokasi yang baru dapat diperoleh dengan cepat.



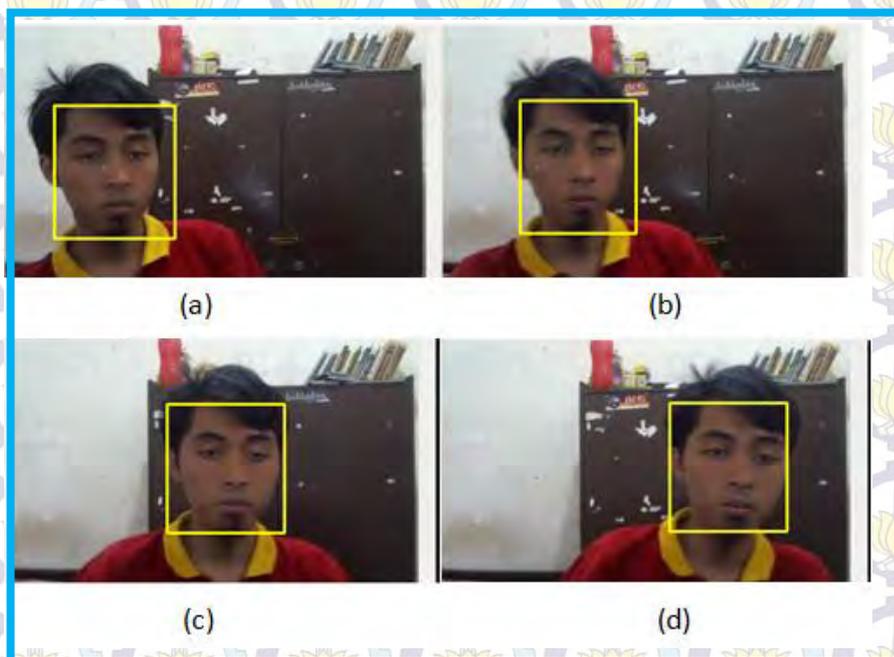
Gambar 4. 12. Perbandingan iterasi antara Improved Mean Shift dan Mean Shift dalam 20 iterasi

Pada Gambar 4.12 merupakan perbandingan iterasi antara *Improved Mean Shift* dan *Mean Shift* dalam 20 iterasi. Pada Gambar 4.12 dapat diketahui bahwa iterasi yang diperlukan *Improved Mean Shift* memiliki iterasi rata-rata yaitu 4.6 sedangkan iterasi rata-rata pada *Mean Shift* yaitu 5.3.

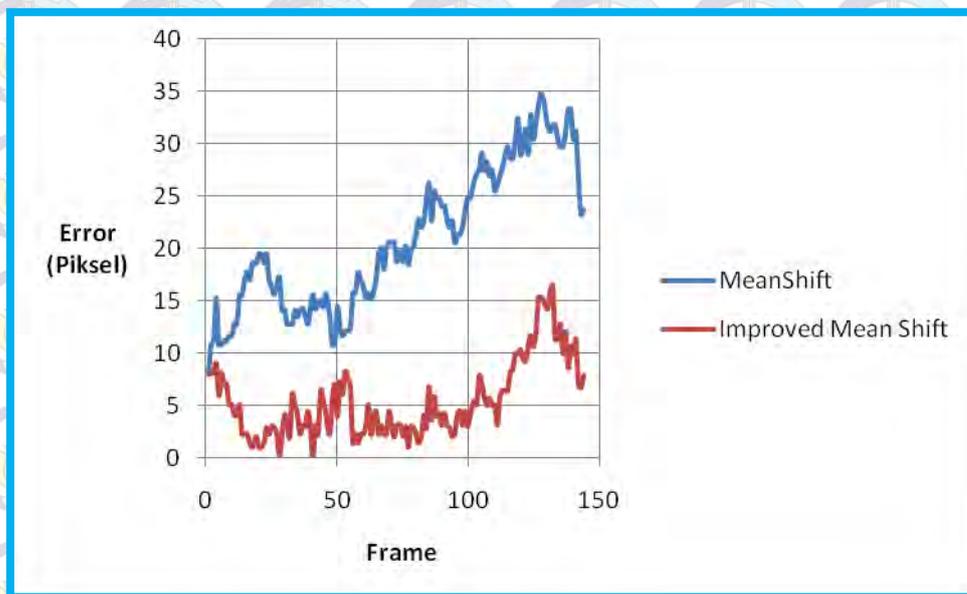
Pada Gambar 4.13 merupakan hasil penjejakan objek pada *background* yang memiliki fitur yang sama dengan objek target, dengan menggunakan metode *Mean Shift*, sedangkan pada Gambar 4.14 merupakan hasil penjejakan objek dengan menggunakan *Improved Mean Shift*. Dari Gambar 4.13 dapat diketahui bahwa penjejakan objek yang dilakukan dengan menggunakan metode *Mean Shift* akan mengalami *error* terpengaruh pada *background* berupa lemari yang memiliki fitur yang sama dengan objek (wajah), sedangkan pada Gambar 4.14 dapat diketahui penjejakan objek dapat dilakukan dengan baik, dan tidak terpengaruh oleh *background*. Hasil penjejakan wajah berupa persegi panjang berwarna kuning. Pada Gambar 4.15 merupakan grafik error penjejakan objek wajah dengan menggunakan *Mean Shift* dan *Improved Mean Shift*. Untuk grafik yang berwarna merah menunjukkan error penjejakan objek dengan menggunakan *Mean Shift*, sedangkan grafik yang berwarna biru menunjukkan error penjejakan objek dengan menggunakan *Improved Mean Shift*.



Gambar 4.13. Hasil pengujian kedua penjejakan objek dengan metode *Mean Shift*, (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150

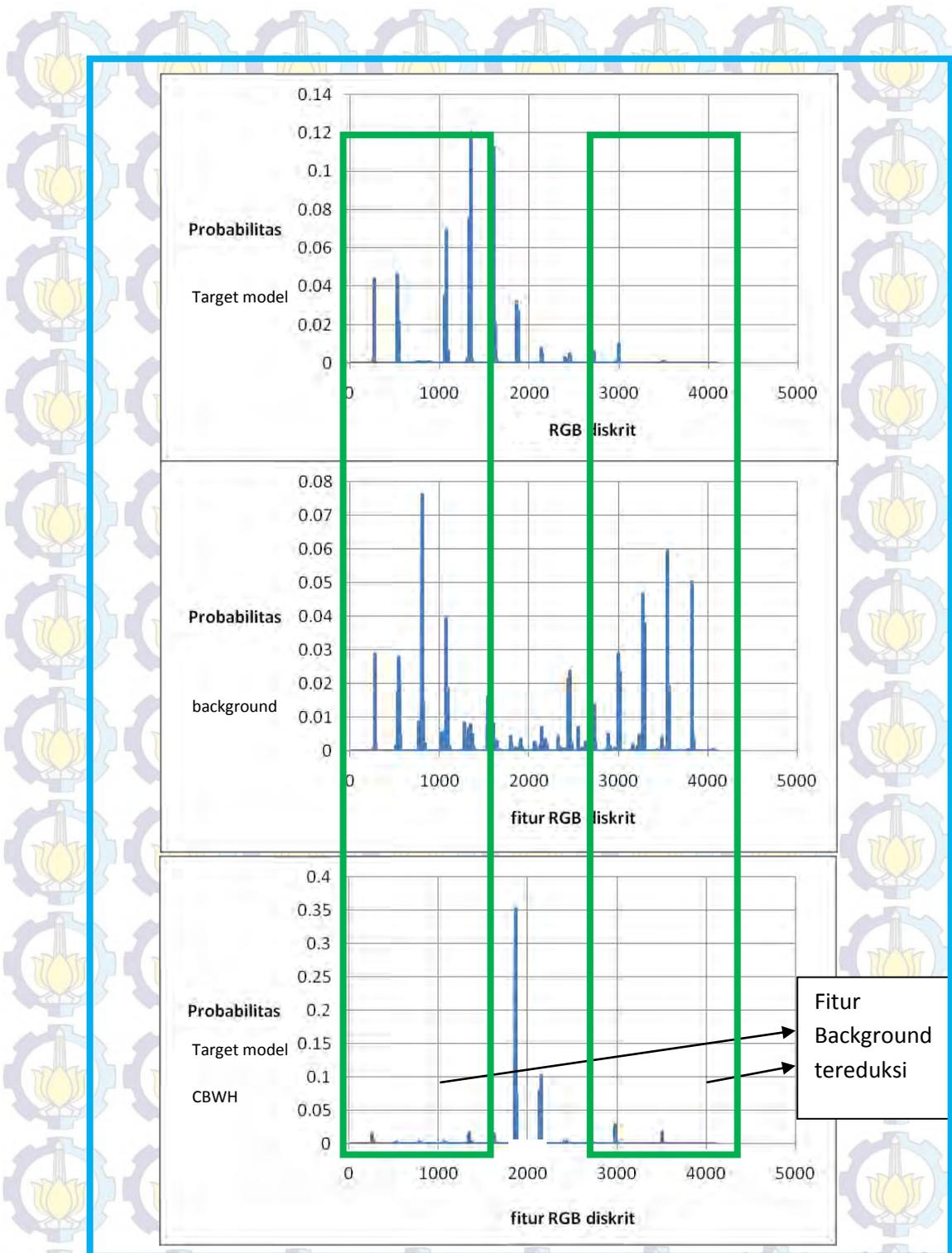


Gambar 4.14. Hasil pengujian kedua penjejakan objek dengan *Improved Mean Shift*, (a) frame ke-1, (b) frame ke-50, (c) frame ke-100, (d) frame ke-150



Gambar 4. 15. Grafik error penjejakan objek wajah dengan menggunakan *Mean Shift* dan *Improved Mean Shift*

Dari Gambar 4.15 dapat diketahui nilai error rata – rata dengan menggunakan metode *Mean Shift* adalah 9.01 piksel dengan standard deviasi 5.77 piksel, sedangkan dengan menggunakan metode *Improved Mean Shift*.dapat diketahui bahwa nilai error rata – rata 3.87 piksel dengan standard deviasi 1.54 piksel. Dari error rata-rata tersebut dapat diketahui bahwa nilai error dengan menggunakan *Improved Mean Shift*. lebih kecil dibandingkan dengan menggunakan metode *Mean Shift*. Penjejakan objek dengan menggunakan metode *Improved Mean Shift* lebih baik dibandingkan dengan menggunakan metode *Mean Shift*, disebabkan karena target model yang digunakan dalam metode *Improved Mean Shift*. tidak terpengaruh oleh *noise background*. Fitur *background* yang terdapat pada target direduksi sehingga pada saat dilakukan pencarian koordinat baru, nilai probabilitas target model yang memiliki fitur *background* tidak digunakan lagi, sehingga perhitungan koordinat yang dicari tidak terpengaruh oleh fitur *background*. Pada Gambar 4.16 menunjukkan bahwa target model *Improved Mean Shift*. tidak memiliki fitur *background*, karena fitur *background* sudah tereduksi dengan metode *Improved Mean Shif*

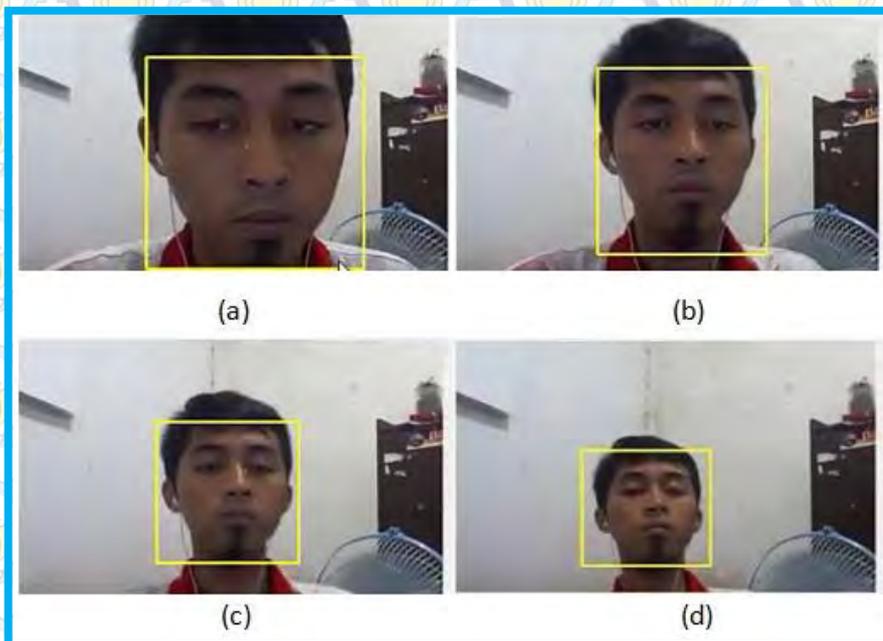


Gambar 4. 16. Histogram probabilitas distribusi warna hasil *Improved Mean Shift* pada penjejukan objek wajah

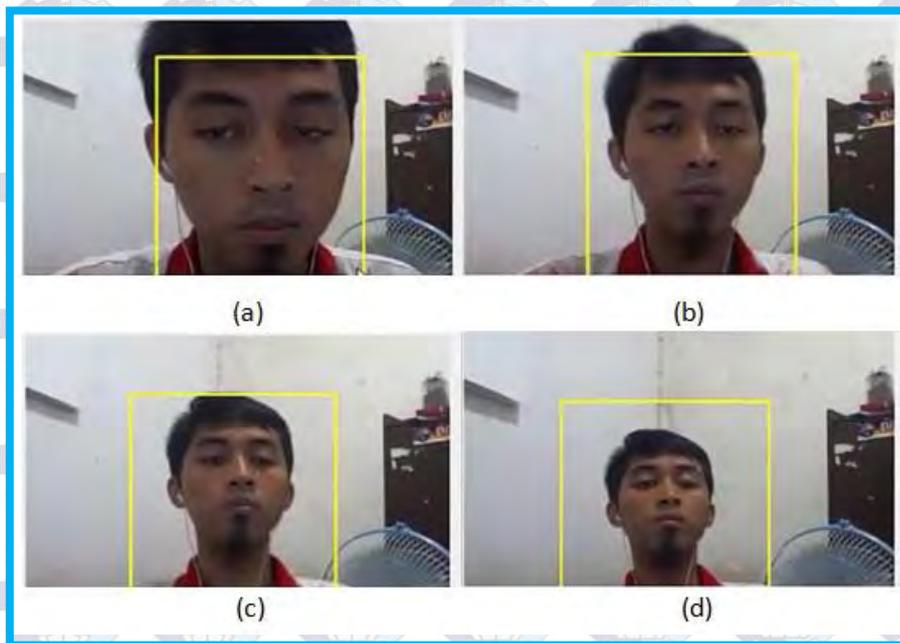
4.4 Pengujian Penskalaan

Pengujian penskalaan objek ini bertujuan untuk menguji apakah metode yang digunakan dapat melakukan pembesaran atau pengecilan ukuran *window* menyesuaikan dengan ukuran objek yang dijadikan target. Pengujian ini dilakukan dengan menggunakan video berupa pergerakan wajah yang menjauh dari kamera. Kemudian *user* memilih wajah yang digunakan dalam penjejakan objek tersebut, kemudian data berupa ukuran lebar dan tinggi disimpan dalam bentuk *.txt, yang kemudian akan dianalisa error yang diperoleh. Pada pengujian ini, dibandingkan dua metode yaitu metode *Mean Shift* dan metode yang diusulkan.

Pada Gambar 4.17 merupakan penjejakan objek dengan menggunakan *Improved Mean Shift*. Pada Gambar 4.17 metode *Improved Mean Shift* dapat melakukan penskalaan dengan baik, hal ini terbukti dengan penjejakan yang diwakili oleh persegi panjang berwarna kuning yang dapat beradaptasi mengikuti ukuran wajah yang dijejak.

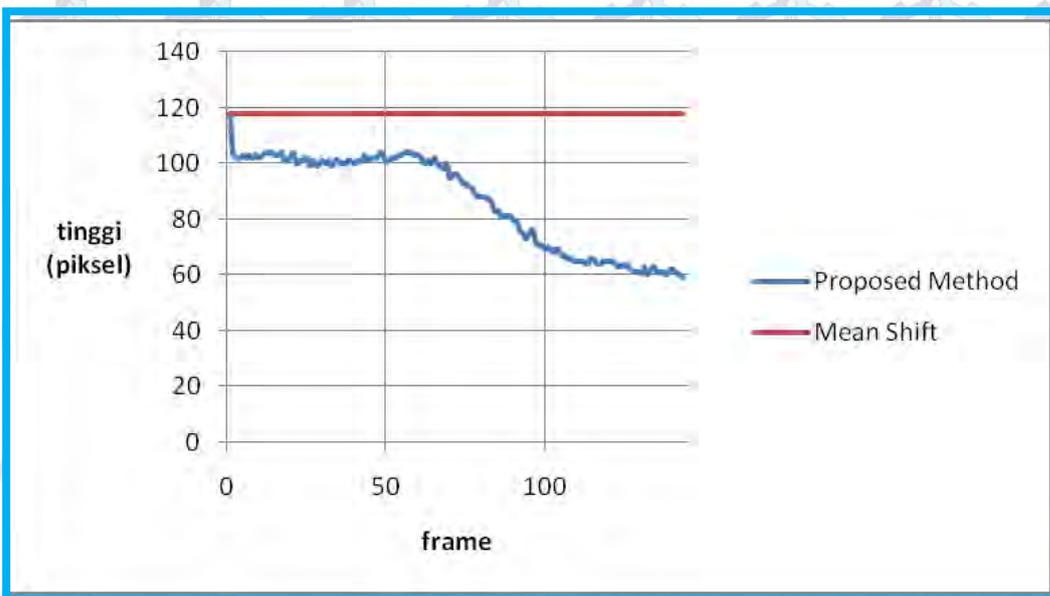


Gambar 4. 17. Grafik penskalaan penjejakan objek wajah dengan menggunakan *improved Mean Shift* (a) frame ke-1, (b) frame ke-80, (c) frame ke-100, (d) frame ke-150

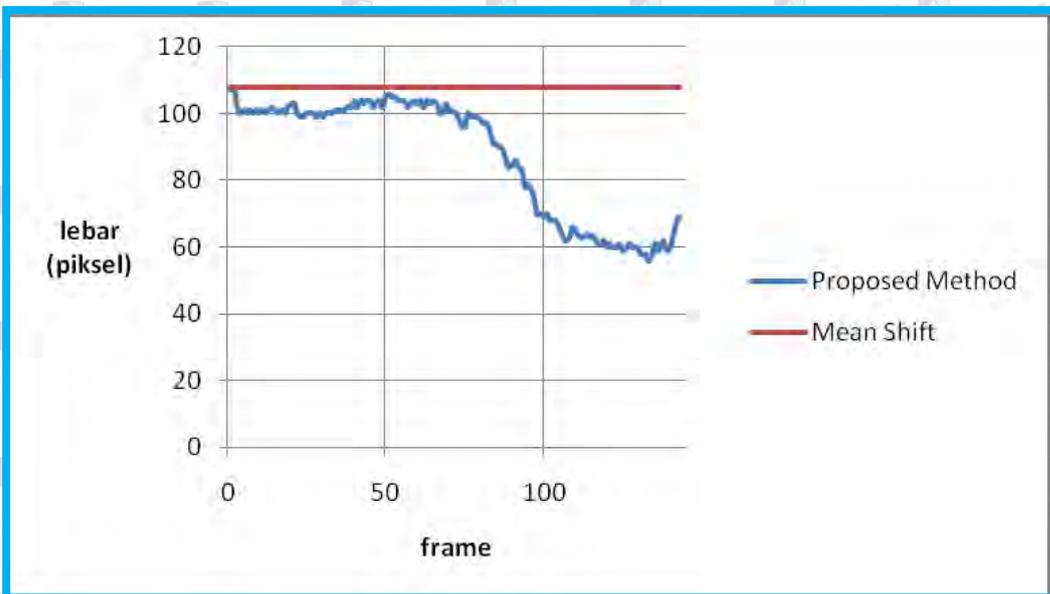


Gambar 4. 18. Grafik penskalaan penjejakan objek wajah dengan menggunakan *Mean Shift*, (a) frame ke-1, (b) frame ke-80, (c) frame ke-100, (d) frame ke-150

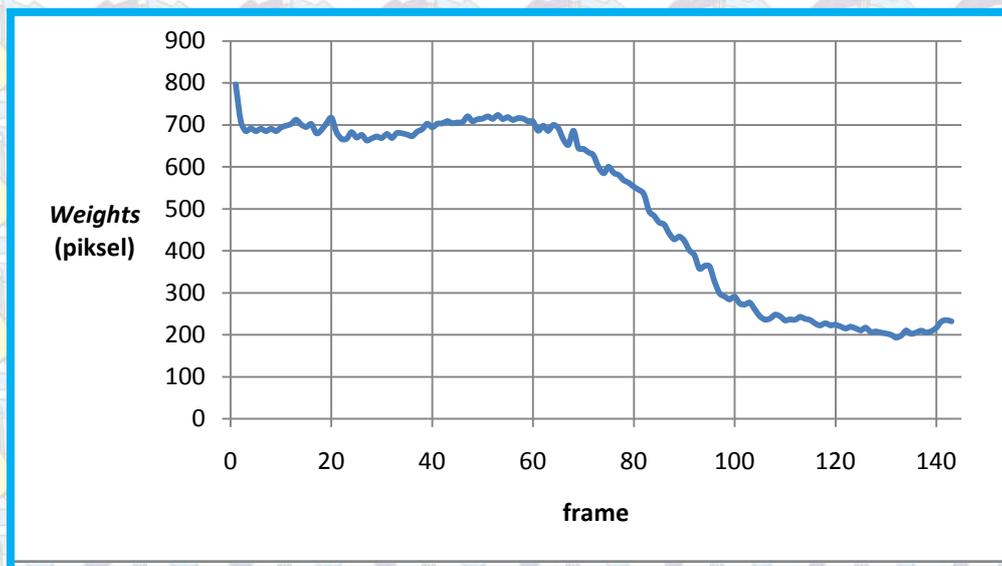
Pada Gambar 4.18 merupakan penjejakan objek dengan menggunakan metode *Mean Shift*. Pada Gambar 4.18 metode *Mean Shift* tidak dapat melakukan penskalaan dengan baik, hal ini terbukti dengan penjejakan yang diwakili oleh persegi panjang berwarna kuning yang tidak dapat beradaptasi mengikuti ukuran wajah yang dijejak. Pada Gambar 4.19 menunjukkan grafik perbandingan antara tinggi objek dengan menggunakan *Improved Mean Shift*. dan *Mean Shift*. Grafik *Improved Mean Shift*. berwarna biru sedangkan grafik untuk metode *Mean Shift* berwarna merah. Pada Gambar 4.20 menunjukkan grafik perbandingan antara lebar objek dengan menggunakan *Improved Mean Shift* dan *Mean Shift*. Dari kedua grafik pada Gambar 4.19 dan 4.20 dapat diketahui bahwa penjejakan objek dengan menggunakan *Mean Shift* tidak dapat melakukan penskalaan terhadap objek, sedangkan pada *Improved Mean Shift* dapat dilakukan penskalaan. Hal ini, karena pada metode *Mean Shift* ukuran objek telah ditentukan sehingga tidak dapat melakukan penskalaan, sedangkan pada *Improved Mean Shift*., ukuran objek berbanding lurus dengan nilai moment pada objek yang dijejak.



Gambar 4.19. Grafik tinggi objek wajah dengan menggunakan *improved Mean Shift*

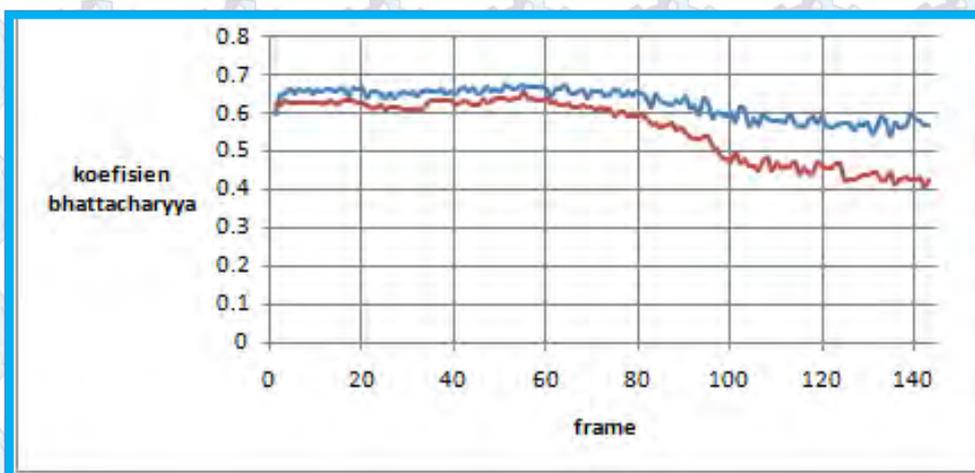


Gambar 4. 20. Grafik lebar objek wajah dengan menggunakan *improved Mean Shift*



Gambar 4. 21. Grafik bobot pada penjejakan objek

Pada Gambar 4.21 menunjukkan grafik *weights* (bobot) yang dihasilkan dari penjejakan pada Gambar 4.16. tersebut dihasilkan dari jumlah bobot (*weights*) pada semua piksel dalam area target. Sehingga ketika nilai bobot pada sebuah area target berkurang, maka ukuran lebar dan tinggi objek pun juga berkurang. Hal ini yang menyebabkan ukuran *window* penjejakan dapat menyesuaikan dengan target. Sehingga dapat disimpulkan ketika nilai moment berbanding lurus dengan ukuran objek.

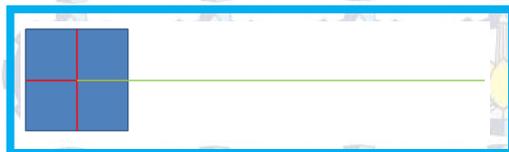


Gambar 4. 22. Grafik koefisien bhattacharyya koefisien antara metode *Improved Mean Shift* (berwarna biru) dan *Mean Shift* (berwarna merah)

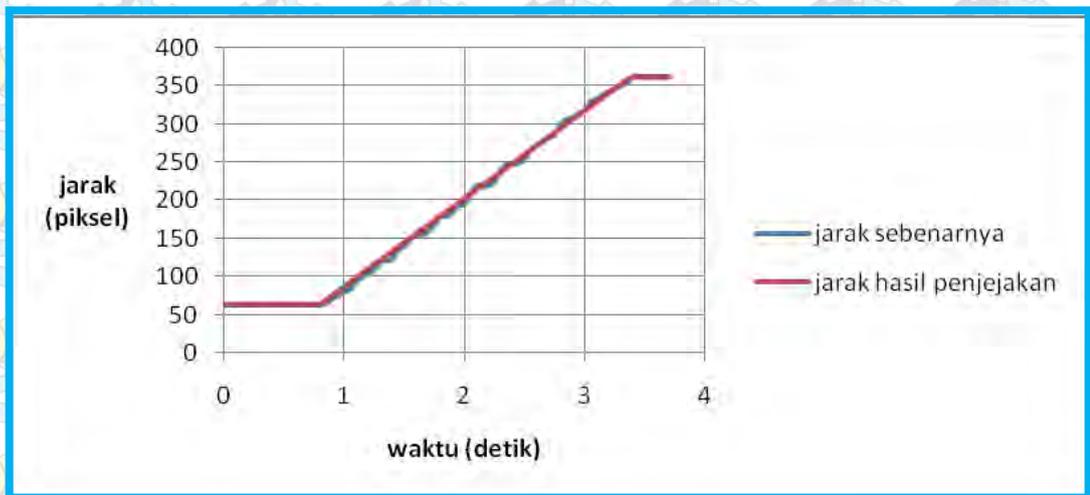
Penskalaan juga dipengaruhi dengan koefisien Bhattacharyya pada objek yang dijejak perframe. Pada Gambar 4.22 merupakan grafik koefisien bhattacharrya koefisien antara *Improved Mean Shift* (berwarna biru) dengan *Mean Shift* (berwarna merah). Pada grafik tersebut dapat diketahui bahwa masing-masing, ketika objek menjauh dari mengecil ukurannya, maka koefisien bhattacharrya juga akan semakin rendah. Hal ini disebabkan perbandingan tingkat kesamaan antara model dan kandidat berbeda, karena objek target menjauh. Pada Gambar 4.22, nilai koefisien *bhattacharya* untuk *Mean Shift* lebih rendah dibandingkan dengan *Improved Mean Shift*, hal ini disebabkan karena nilai ukuran objek pada *Mean Shift* tetap pada saat objek mengecil, sehingga tingkat perbandingan model target dan kandidat berbeda jauh, berbeda ketika ukuran *window* mengikuti ukuran objek.

4.5 Pengujian Kecepatan Metode Penjejakan Objek

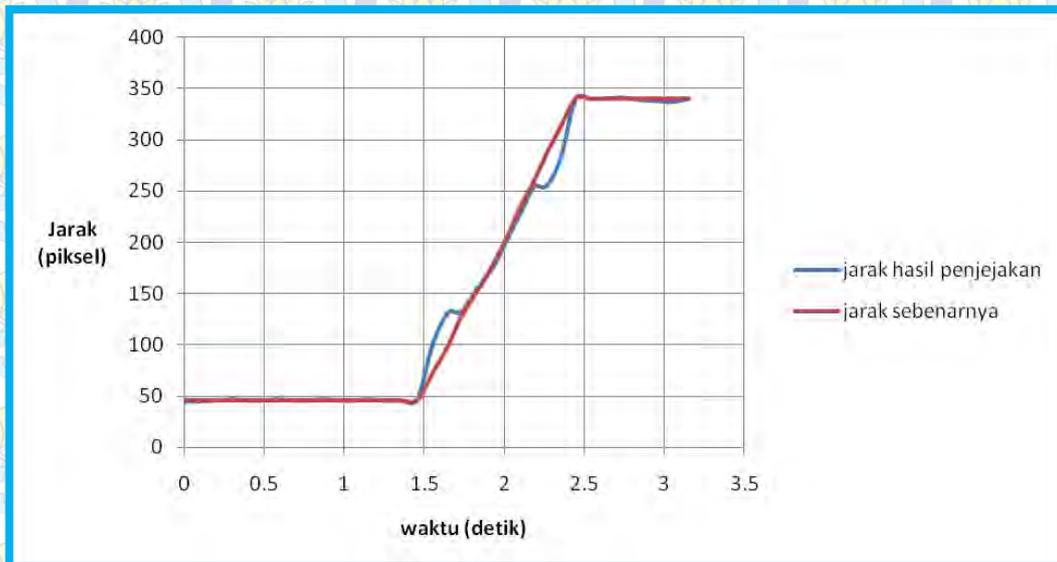
Pada pengujian kecepatan metode penjejakan objek bertujuan untuk menguji kecepatan maksimum yang dapat dilakukan metode tersebut untuk mengikuti objek yang dijejak. Pengujian dilakukan dengan cara *tracking* obyek pada video, dimana kecepatan pergerakan obyeknya dapat diatur. Obyek yang di *tracking* berupa persegi yang ditunjukkan dalam Gambar 4.23. Gambar persegi tersebut bergerak sepanjang garis berwarna hijau. Objek berwarna biru. Hasil Pengujiannya digambarkan dalam bentuk grafik pada Gambar 4.24. Grafik yang ditunjukkan adalah grafik jarak dalam satuan pixel terhadap waktu dalam satuan detik. Gambar tersebut bermaksud menjelaskan sampai jarak dan waktu berapakah metode dapat mengikuti pergerakan obyek ketika kecepatan pergerakan obyek semakin lama semakin besar. Dimana garis berwarna merah adalah jarak obyek sebenarnya dan garis berwarna biru adalah hasil penjejakan.



Gambar 4. 23. Pengujian kecepatan metode objek

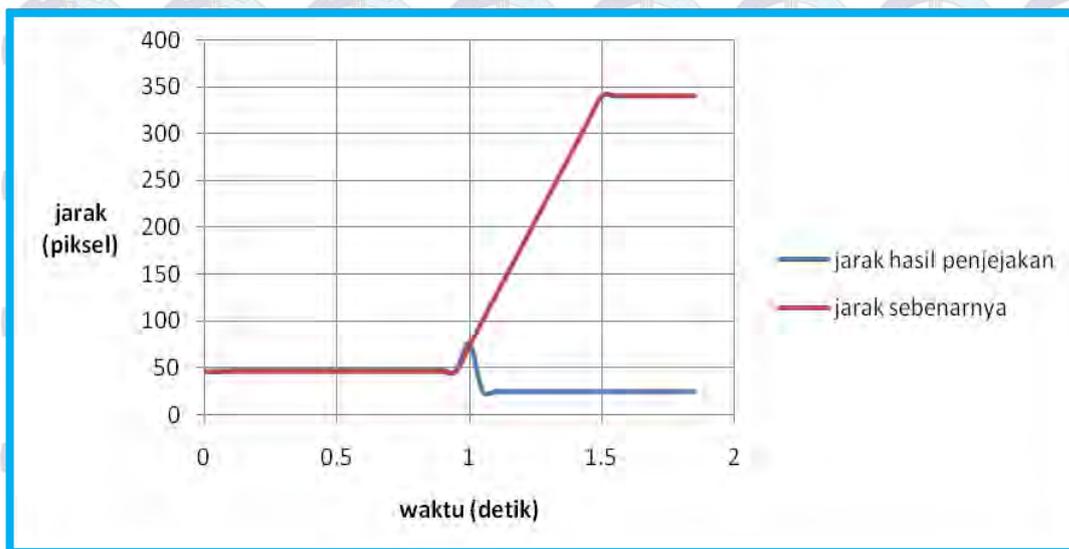


Gambar 4. 24. Grafik pengujian pada kecepatan 80.81 piksel/ detik



Gambar 4. 25. Grafik pengujian pada kecepatan 294.44 piksel/ detik

Pada Gambar 4.24 hingga 4.26 merupakan hasil pengujian kecepatan objek dengan kecepatan yang berbeda-beda. Dari hasil pengujian tersebut dapat diketahui bahwa objek dapat mengikuti objek dengan kecepatan hingga 294.44 piksel/detik, ketika kecepatan mencapai 421.22 piksel/ detik, penjejakan akan error tidak dapat mengikuti objek.

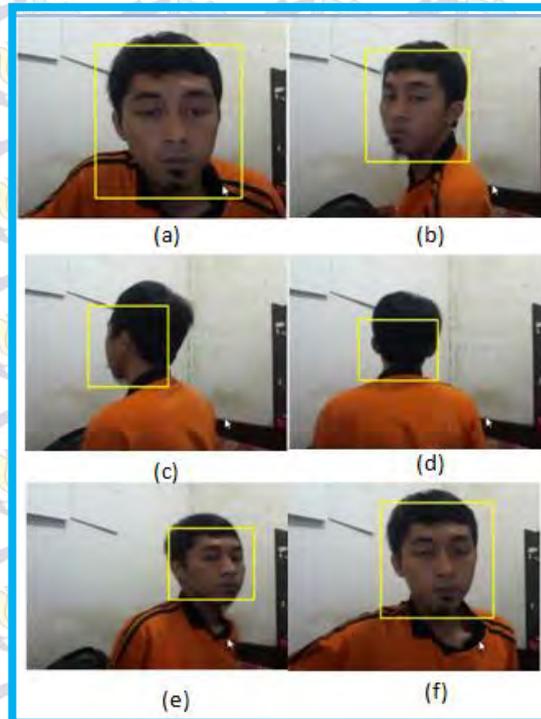


Gambar 4. 26. Grafik pengujian pada kecepatan 421.22 piksel /detik

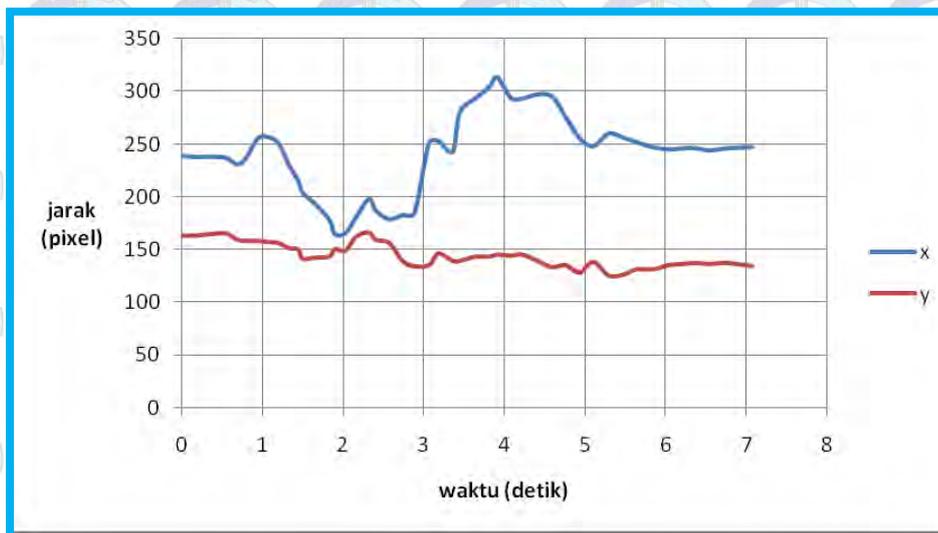
4.6 Pengujian Ketahanan Metode Penjejakan Objek

Pengujian ini bertujuan untuk mengetahui ketahanan objek pada saat penjejakan dari gangguan yang dapat membuat penjejakan objek hilang. Pengujian pertama dilakukan untuk mengetahui ketahanan metode penjejakan dari ketika objek target berotasi. Pengujian dilakukan dengan cara user memilih objek yang dijadikan sebagai target, kemudian objek berputar ditunjukkan dalam Gambar 4.27. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing –masing gambar. Pada Gambar 4.28 merupakan grafik hasil pengujian ketahanan objek terhadap sumbu x dan sumbu y. Pada Gambar 4.29 merupakan grafik untuk lebar dan tinggi objek pada saat objek berotasi. Lebar dan tinggi objek sempat mengecil hal ini disebabkan karena fitur kandidat target berbeda dengan target model. Untuk mengetahui tingkat kesamaan dari kandidat target dan model dapat diketahui dari nilai koefisien bhattacharrya. Pada Gambar 4.30 merupakan koefisien Bhattacharyya pada saat objek berotasi. Pada saat waktu 2 -4 detik nilai koefisien turun, hal ini disebabkan distribusi fitur warna antara model dan kandidat berbeda. Walaupun nilai koefisien Bhattacharyya turun, metode tetap dapat menjejak objek, hal ini disebabkan karena metode terus mencari nilai koordinat yang mendekati dengan fitur model target. Ketika wajah kembali terdeteksi, maka nilai koefisien Bhattacharyya akan naik . Pada pengujian tersebut

dapat diketahui bahwa metode penjejakan objek masih dapat mejejakan objek dengan baik.



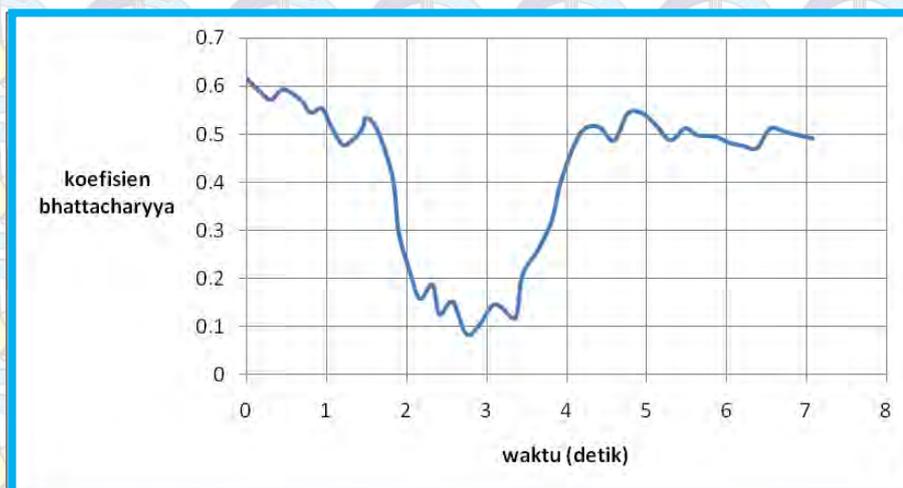
Gambar 4.27. Pengujian ketahanan ketika objek berotasi



Gambar 4.28. Grafik koordinat x dan y pada pengujian ketahanan ketika objek berotasi.



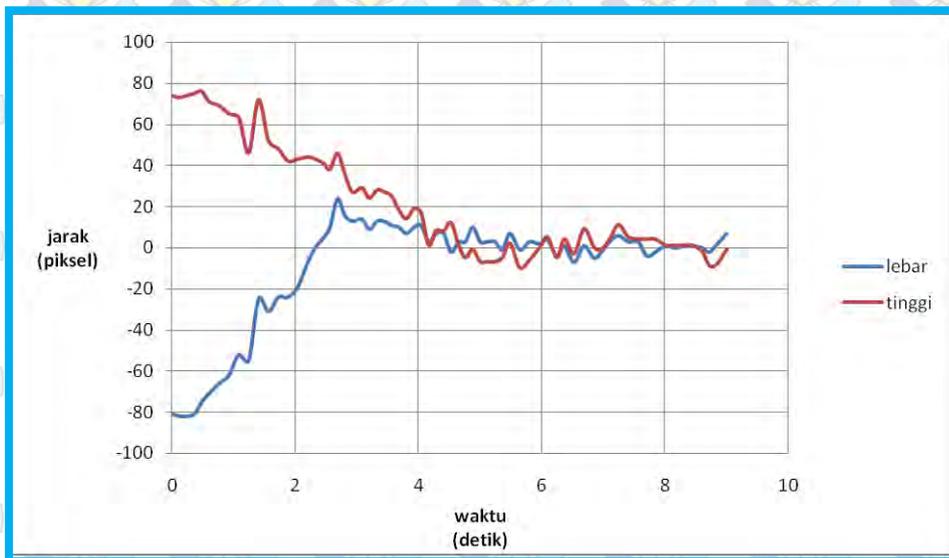
Gambar 4. 29. Grafik lebar dan tinggi pengujian ketahanan ketika objek berotasi



Gambar 4. 30. Grafik koefisien Bhattacharyya pada pengujian ketahanan objek

4.7 Pengujian Kontrol Pan Tilt

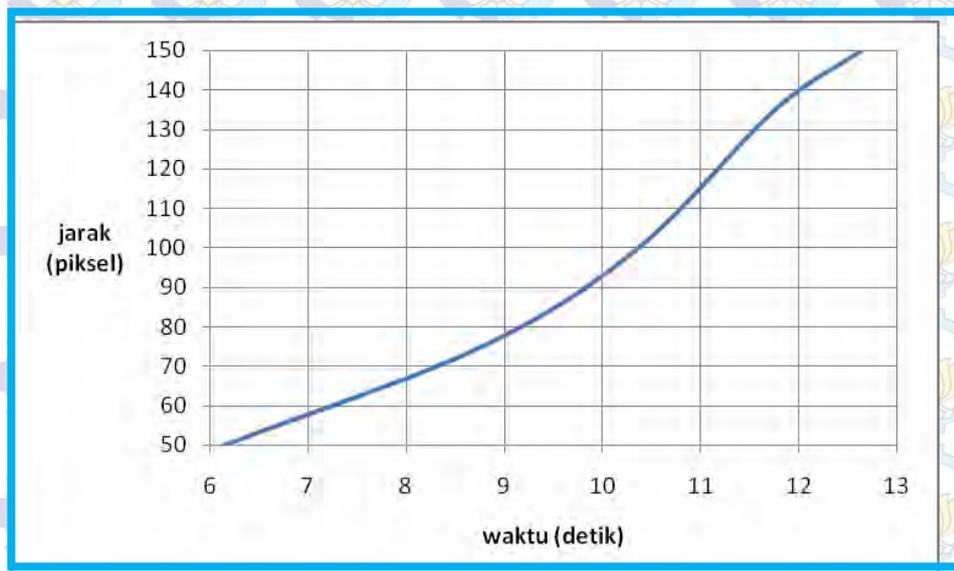
Pengujian ini bertujuan untuk melihat performansi respon pergerakan kamera secara horizontal (pan) dan vertikal (tilt). Pengujian ini dilakukan dengan cara memilih satu titik koordinat objek menggunakan mouse. Kamera akan bergerak menuju titik koordinat tersebut mencapai set point. Hasil pengujian ditunjukkan dalam Gambar 4.31 dan Tabel 4.2



Gambar 4.31. Grafik hasil pengujian kontrol pan-tilt

Tabel 4.2. Hasil *error steady state* dan *settling time* pada kontrol pan-tilt

	error steady state (%)	settling time (detik)
x	2	6.091
y	2.8125	6.363

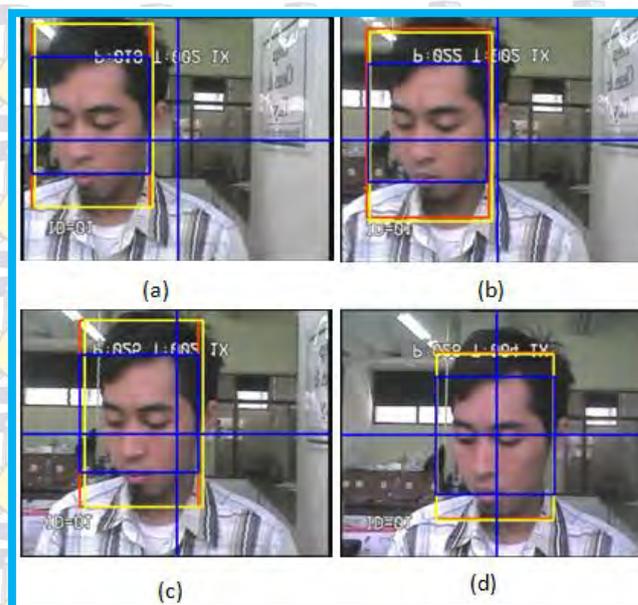


Gambar 4.32. Grafik hasil pengujian pengukuran waktu

Pada Tabel 4.2, dapat diketahui bahwa sistem dapat menuju set point secara bersamaan. Sistem dapat mencapai set point pada waktu 6.363 detik dimana jarak antara koordinat objek dengan koordinat set point adalah 110.344 piksel. Selanjutnya dilakukan pengukuran waktu pencapaian *set point*, yaitu dengan cara menentukan titik koordinat objek pada koordinat yang berbeda-beda setiap penentuan titik. Hasil pengujiannya ditunjukkan dalam Gambar 4.31 yang dapat dianalisis bahwa semakin jauh jarak antara titik koordinat obyek dengan *set point*, maka semakin lama waktu yang dibutuhkan untuk mencapai *set point*. Pada jarak terdekat 50 piksel, dibutuhkan waktu 6.156 detik, sedangkan pada jarak terjauh 150 piksel, dibutuhkan waktu 12.64 detik.

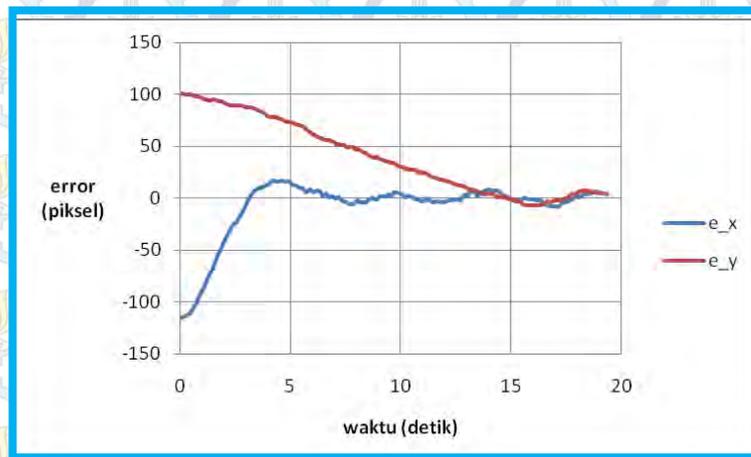
4.8 Pengujian Penjejakan Objek Statis

Pengujian ini bertujuan untuk mengetahui performansi penjejakan objek ketika objek yang dijadikan sebagai target dalam kondisi diam. Objek yang dipilih berupa wajah dengan menggunakan mouse. *Set point* merupakan perpotongan garis yang berwarna biru.



Gambar 4. 33. Pengujian penjejakan objek dengan objek diam

Sedangkan garis yang berwarna kuning merupakan objek yang dipilih. Pada Gambar 4.32 (a) merupakan awal pemilihan objek. Pada Gambar 4.32 (b) dan (c), sistem menuju ke titik *set point*. Gambar 4.32 (d) sistem berada pada titik *set point*. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing-masing gambar.



Gambar 4.34. Hasil objek dengan objek diam

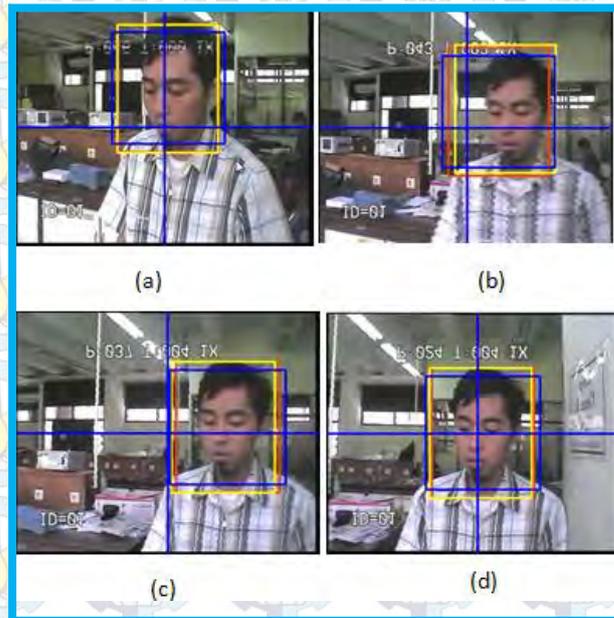
Tabel 4. 3. Hasil error steady state dan settling time pada objek statis

	settling time (detik)	error steady state (%)
x	6.279	1.5
y	16.412	1.25

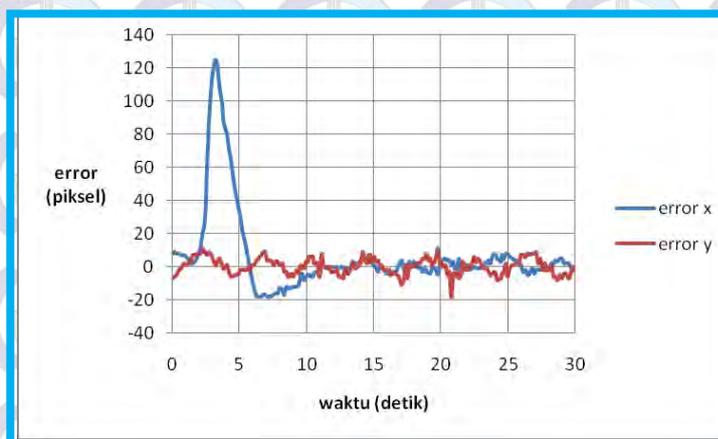
4.9 Pengujian Penjejakan Objek bergerak

Pengujian ini bertujuan untuk mengetahui performansi penjejakan objek ketika objek yang dijadikan sebagai target dalam kondisi bergerak. Kondisi objek bergerak dibagi menjadi empat yaitu bergerak kekanan, kekiri, diagonal kekanan dan diagonal kekiri. Pada pengujian pertama dilakukan pengujian objek dengan objek bergerak ke kanan. Objek yang dipilih berupa wajah dengan menggunakan mouse. *Set point* merupakan perpotongan garis yang berwarna biru. Sedangkan garis yang berwarna kuning merupakan objek yang dipilih. Pada Gambar 4.34 (a) merupakan awal pemilihan objek. Pada Gambar 4.34(b) dan (c), sistem menuju ke titik *set point*. Gambar 4.34 (d) sistem berada pada titik *set point*. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing-masing gambar.

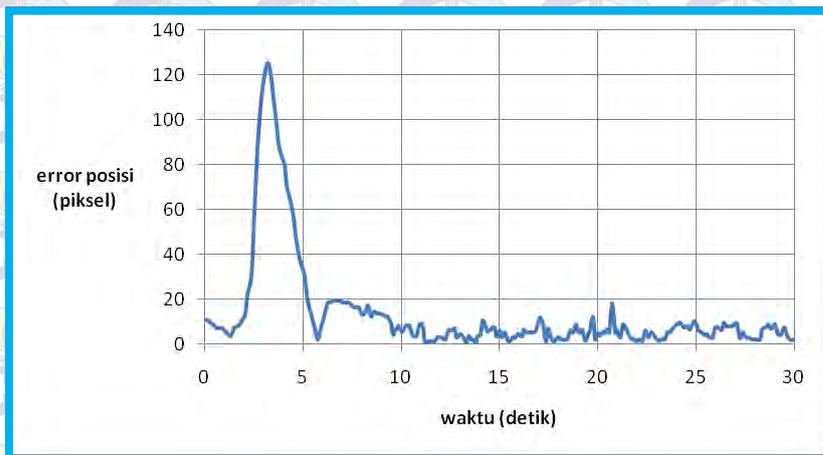
Pada Gambar 4.37 merupakan error posisi untuk objek bergerak kekanan. Error posisi diperoleh dengan menggunakan Persamaan 4.1 . Error posisi menunjukkan bahwa semakin bertambah nilai waktu waktu maka sistem error posisi akan bergerak menuju 0. Hal ini menunjukkan bahwa sistem penjejukan menuju set point pada kondisi bergerak.



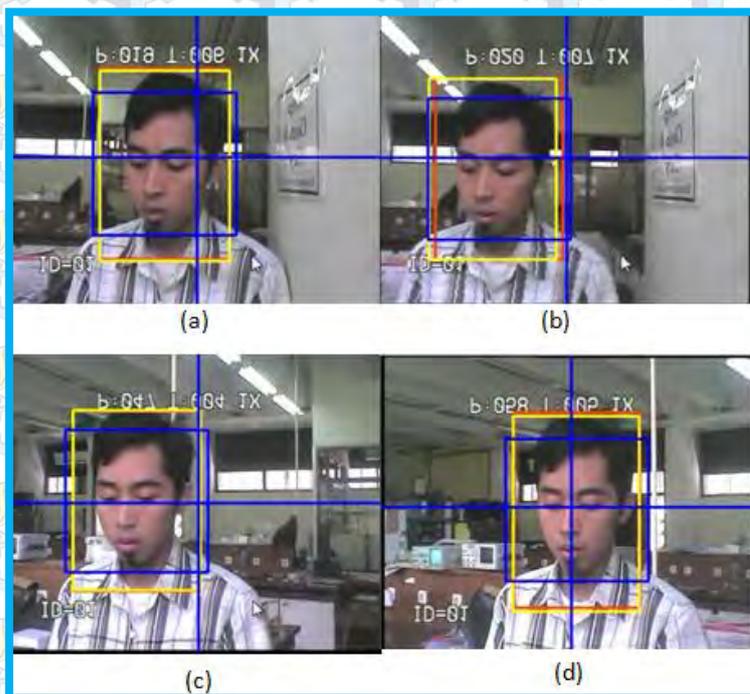
Gambar 4. 35 Pengujian penjejukan objek bergerak ke kanan



Gambar 4.36 Hasil grafik pengujian penjejukan objek dengan objek bergerak kekanan

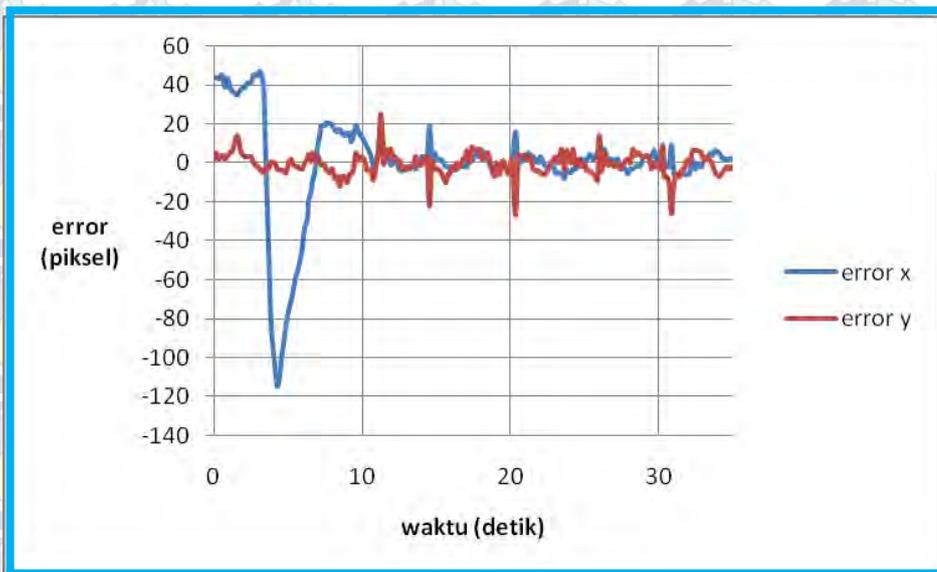


Gambar 4. 37. Hasil grafik error posisi pada penjejakan objek bergerak kekanan

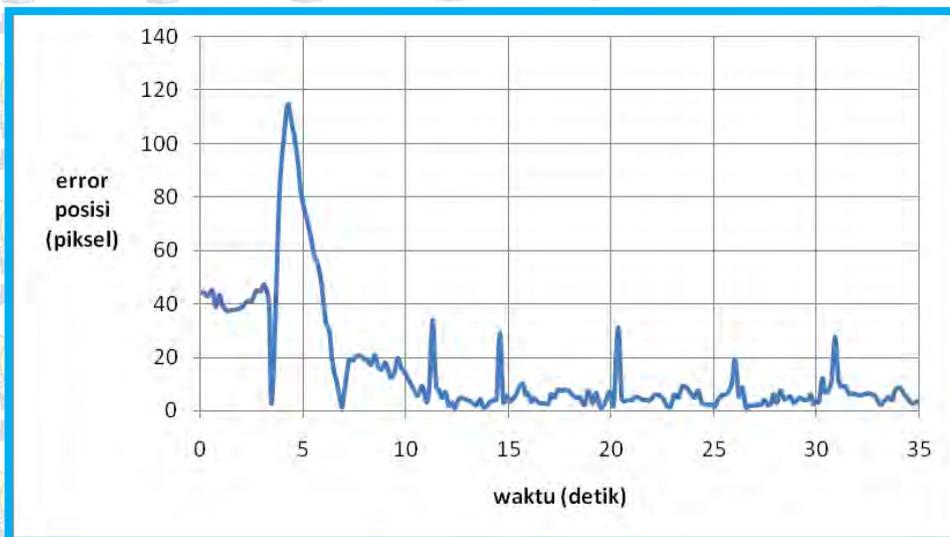


Gambar 4.38. Pengujian penjejakan objek bergerak ke kiri

Pada pengujian kedua dilakukan pengujian objek dengan objek bergerak ke kiri. Objek yang dipilih berupa wajah dengan menggunakan mouse. *Set point* merupakan perpotongan garis yang berwarna biru. Sedangkan garis yang berwarna kuning merupakan objek yang dipilih. Pada Gambar 4.38 (a) merupakan awal pemilihan objek.



Gambar 4. 39. Hasil grafik pengujian penjejakan objek dengan objek bergerak kekiri



Gambar 4. 40. Hasil error posisi dengan objek bergerak kekiri

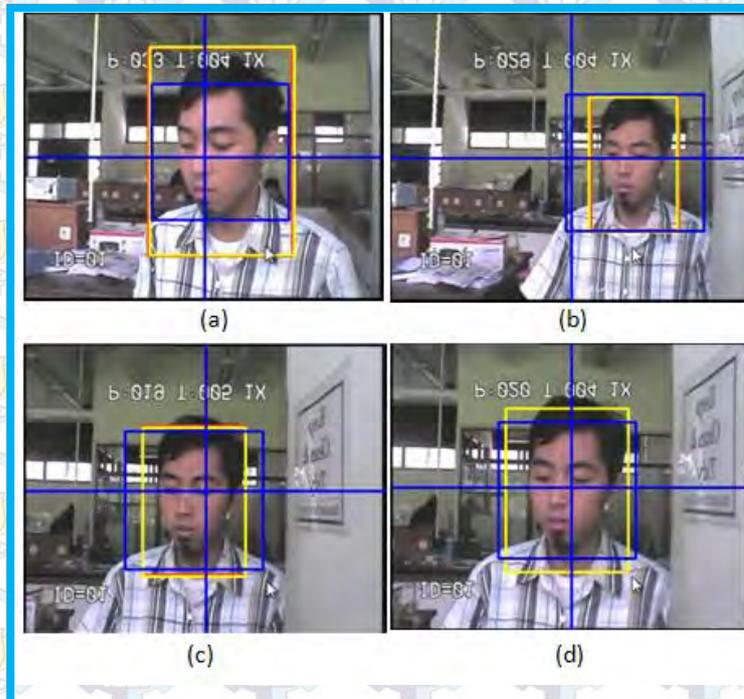
Pada Gambar 4.38(b) dan (c), sistem menuju ke titik *set point*. Gambar 4.38(d) sistem berada pada titik *set point*. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing- masing gambar. Pada Gambar 4.39 merupakan error masing-masing koordinat pada penjejakan objek bergerak kekiri, sedangkan pada Gambar 4.40 menunjukkan error posisi pada objek bergerak kekiri.

Error posisi menunjukkan bahwa semakin bertambah nilai waktu maka sistem error posisi akan bergerak menuju 0.

Pada pengujian ketiga dilakukan pengujian objek dengan objek bergerak diagonal ke kanan. Objek yang dipilih berupa wajah dengan menggunakan mouse.

Set point merupakan perpotongan garis yang berwarna biru. Sedangkan garis yang berwarna kuning merupakan objek yang dipilih. Pada Gambar 4.41 (a) merupakan awal pemilihan objek. Pada Gambar 4.41 (b) dan (c), sistem menuju ke titik *set point*. Gambar 4.41 (d) sistem berada pada titik *set point*. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing-masing gambar. Pada

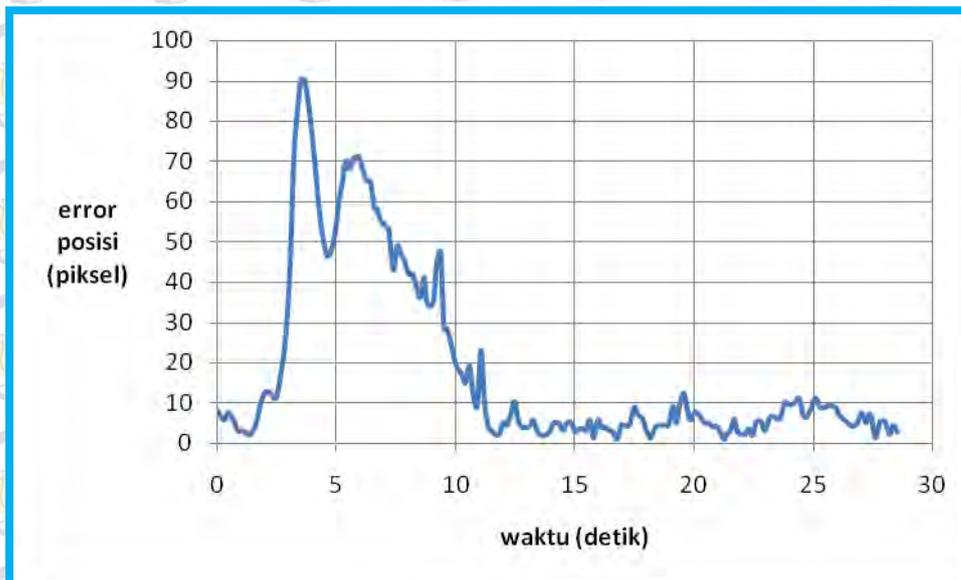
Gambar 4.42 merupakan error masing-masing koordinat pada penjejakan objek bergerak diagonal ke kanan, sedangkan pada Gambar 4.43 menunjukkan error posisi pada objek bergerak diagonal ke kanan. Error posisi menunjukkan bahwa semakin bertambah nilai waktu waktu maka sistem error posisi akan bergerak menuju 0.



Gambar 4.41. Pengujian penjejakan objek diagonal kekanan



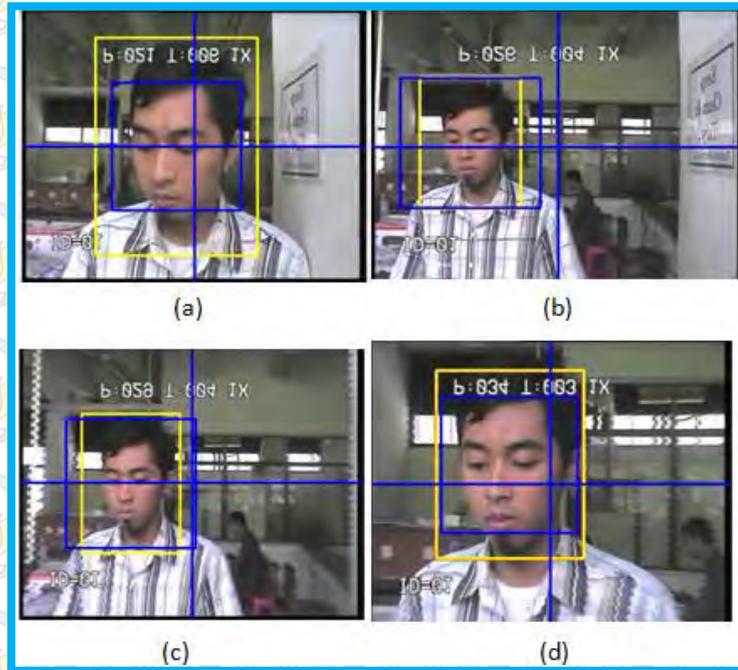
Gambar 4.42. Hasil grafik pengujian penjejakan objek dengan objek bergerak diagonal ke kanan



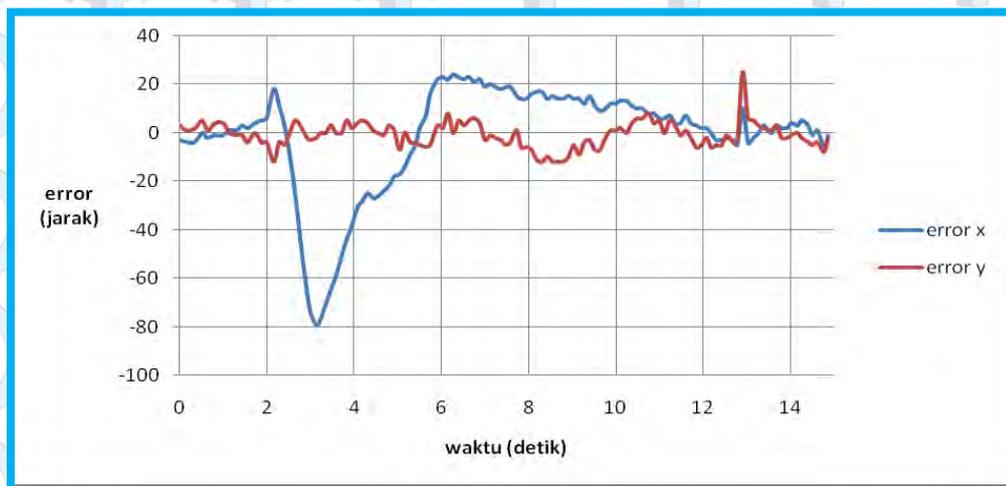
Gambar 4.43. Error posisi objek bergerak diagonal ke kanan

Pada pengujian keempat dilakukan pengujian objek dengan objek bergerak diagonal ke kiri. Objek yang dipilih berupa wajah dengan menggunakan mouse. *Set point* merupakan perpotongan garis yang berwarna biru. Sedangkan garis yang berwarna kuning merupakan objek yang dipilih. Pada Gambar 4.44 (a) merupakan awal pemilihan objek. Pada Gambar 4.44 (b) dan (c), sistem menuju

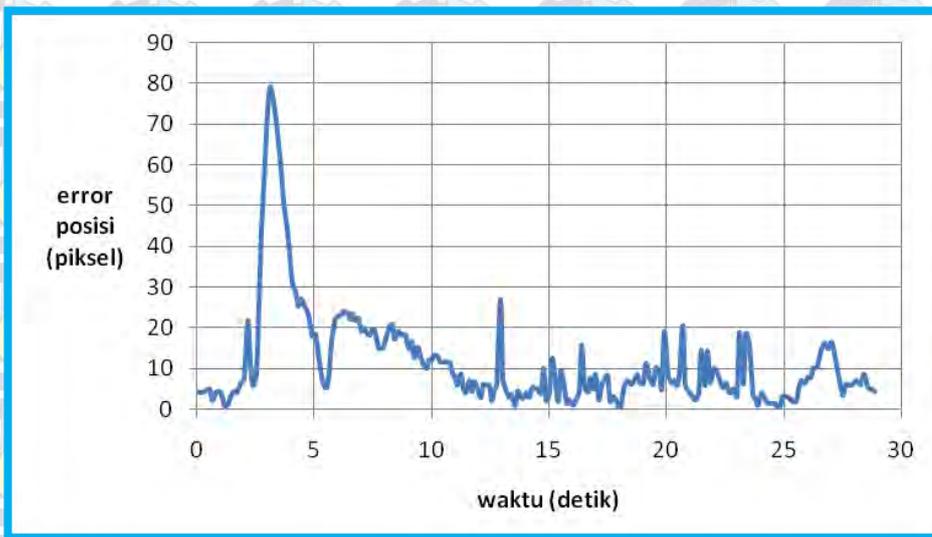
ke titik *set point*. Gambar 4.44 (d) sistem berada pada titik *set point* dan pembesaran objek. Alur pembacaan gambar sesuai urutan huruf abjad yang tercantum pada masing- masing gambar.



Gambar 4. 44. Pengujian penjejakan objek diagonal kekiri



Gambar 4.45. Grafik Pengujian penjejakan objek diagonal kekiri



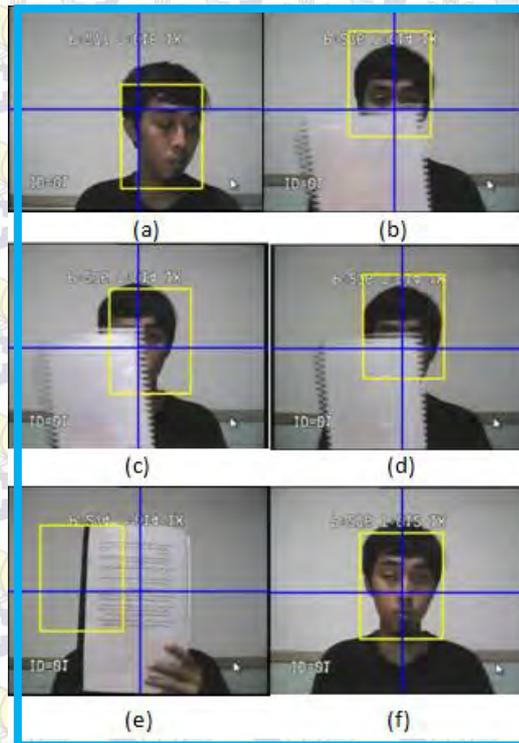
Gambar 4.46. Grafik Pengujian penjejakan objek diagonal kekiri

Gambar 4.45 merupakan error masing-masing koordinat pada penjejakan objek bergerak diagonal ke kiri, sedangkan pada Gambar 4.46 menunjukkan error posisi pada objek bergerak diagonal ke kiri. Error posisi menunjukkan bahwa semakin bertambah nilai waktu waktu maka sistem error posisi akan bergerak menuju 0.

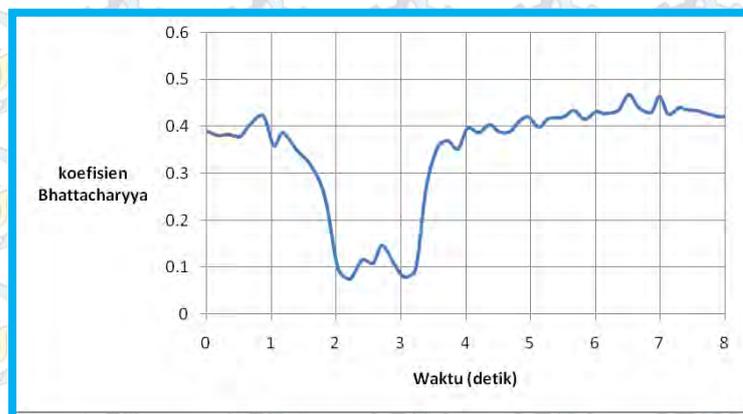
4.10 Pengujian Oklusi pada Penjejakan Objek

Pada pengujian oklusi pada penjejakan objek bertujuan untuk mengetahui ketahanan objek yang dijejak terhadap oklusi (halangan). Pengujian dilakukan dengan dua tahap yaitu pertama oklusi berada didepan objek, dan yang kedua adalah oklusi berada dibelakang objek. Pada oklusi bagian depan dibagi menjadi dua tahap yaitu oklusi dengan fitur yang mirip dengan fitur yang berbeda. Pada Gambar 4.47 merupakan pengujian dengan oklusi dengan fitur yang berbeda. Objek target yang dipilih adalah wajah, sedangkan oklusi yang digunakan adalah kertas berwarna putih. Pada Gambar 4.47 dapat menunjukkan bahwa sistem penjejakan tahan terhadap oklusi dengan fitur oklusi yang berbeda dengan objek target. Pada pengangan oklusi digunakan dua parameter untuk membantu metode Improved Mean Shift dalam mengatasi oklusi, yaitu koefisien Bhattacharyya dan Normalized Cross Correlation (NCC) . Pada Gambar 4.48 menunjukkan koefisien Bhattacharrya pada penjejakan objek. Nilai koefisien yang menunjukkan dibawah

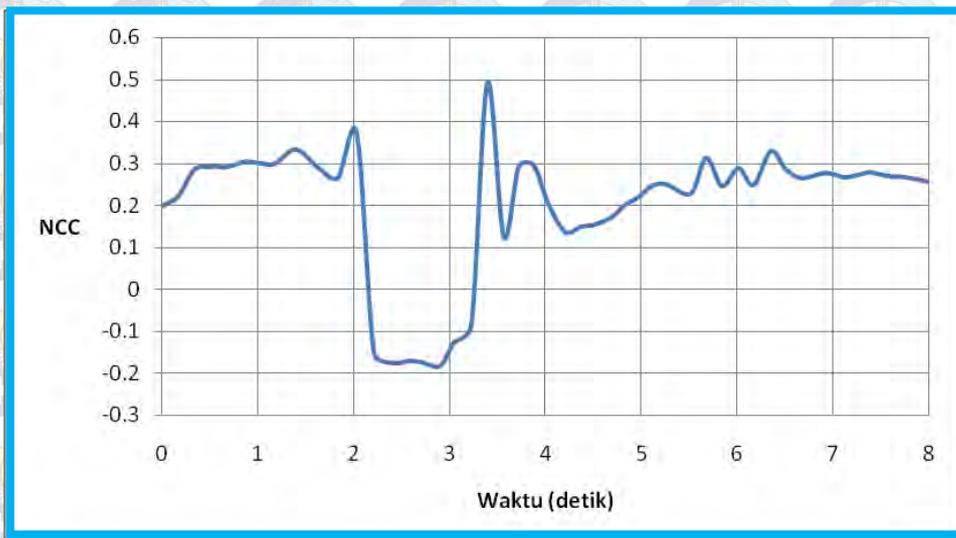
0.2 menandakan bahwa objek target terkena oklusi, maka koordinat sistem akan mengikuti koordinat terakhir. Pada Gambar 4.49 menunjukkan NCC pada penjeakan objek. Nilai NCC yang menunjukkan dibawah 0.1 menandakan bahwa objek target terkena oklusi, maka koordinat sistem akan mengikuti koordinat terakhir.



Gambar 4. 47. Pengujian oklusi dengan fitur yang berbeda.

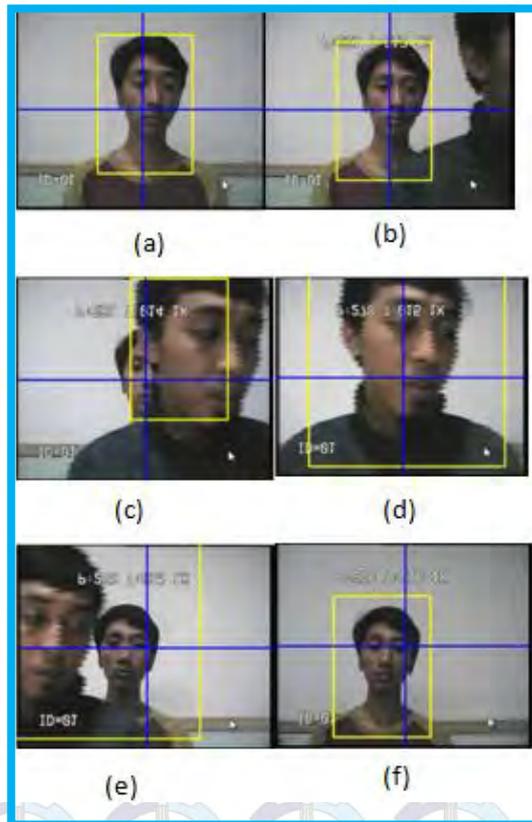


Gambar 4. 48. Koefisien Bhattacharyya pada pengujian oklusi.

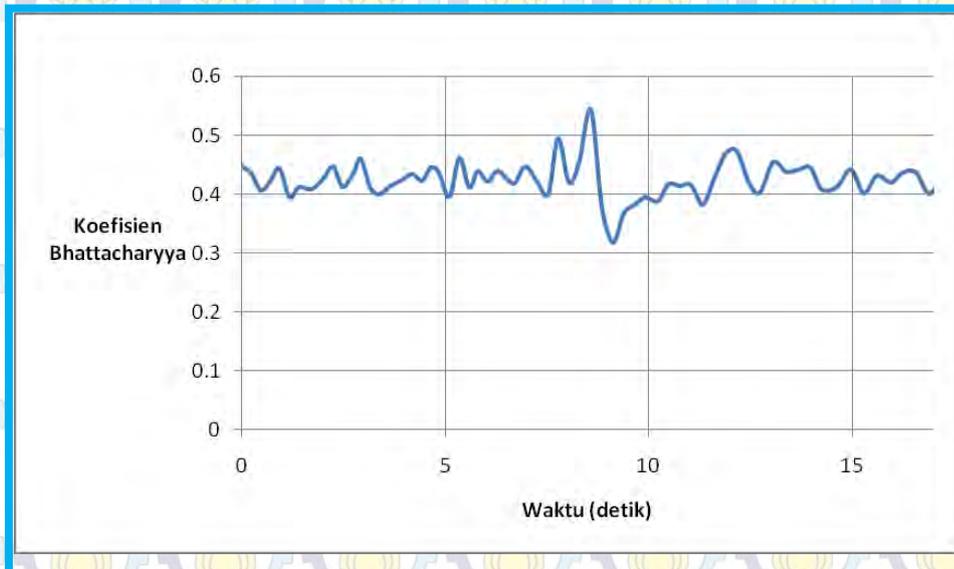


Gambar 4. 49. NCC pada pengujian oklusi.

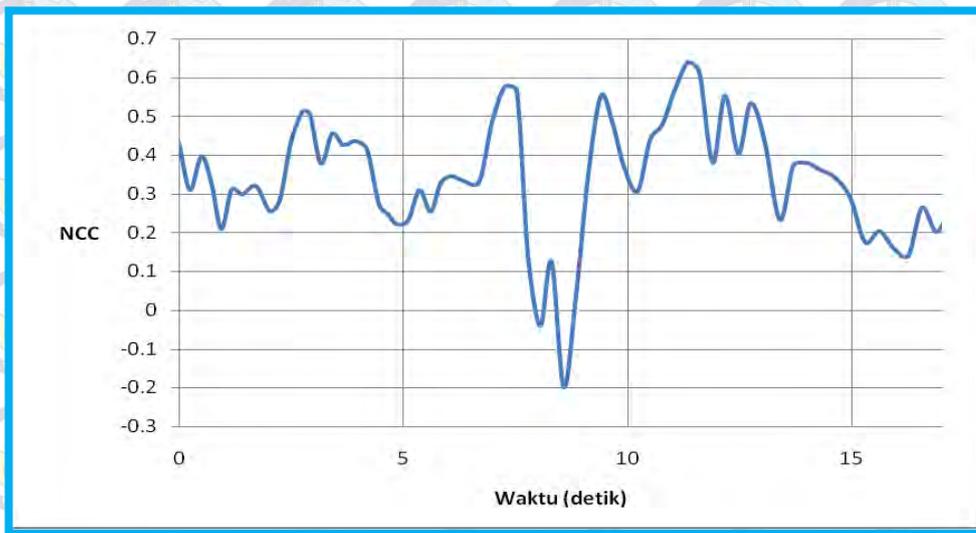
Gambar 4.50 merupakan pengujian dengan oklusi dengan fitur yang sama. Objek target yang dipilih adalah wajah, sedangkan oklusi yang digunakan adalah wajah. Pada Gambar 4.50 dapat menunjukkan bahwa sistem penjejakan tahan terhadap oklusi dengan fitur oklusi yang berbeda dengan objek target. Pada pengangan oklusi digunakan dua parameter untuk membantu metode *Improved Mean Shift* dalam mengatasi oklusi, yaitu koefisien Bhattacharyya dan Normalized Cross Correlation (NCC) . Pada Gambar 4.51 menunjukkan koefisien Bhattacharrya pada penjejakan objek. Nilai koefisien yang menunjukkan dibawah 0.2 menandakan bahwa objek target terkena oklusi, maka koordinat sistem akan mengikuti koordinat terakhir. Tetapi pada Gambar 4.52 tidak dapat menunjukkan adanya oklusi , sehingga kofisien Bhattacharyya gagal mendeteksi adanya oklusi Pada Gambar 4.53 menunjukkan NCC pada penjejakan objek. Nilai NCC yang menunjukkan dibawah 0.1 menandakan bahwa objek target terkena oklusi, maka koordinat sistem akan mengikuti koordinat terakhir. Ketika terkena oklusi maka window sistem penjejakan akan membesar dan mencari objek yang target. Perluasan window penjejakan ini berfungsi untuk menghindari pergerakan objek yang tidak teratur. Pada Gambar 4.53 menunjukkan adanya bahwa sistem dapat tahan ketika objek dilewati pada bagian belakang objek



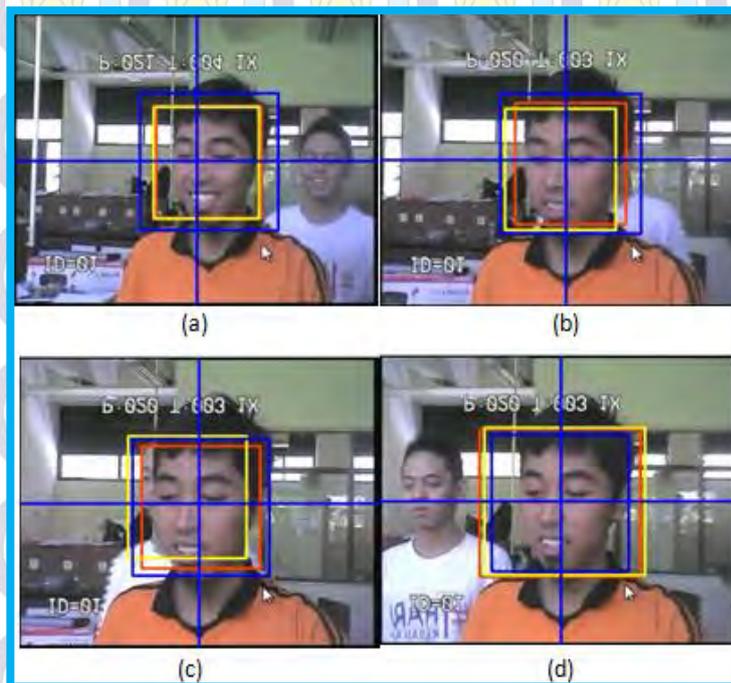
Gambar 4. 50. Pengujian oklusi dengan fitur yang sama



Gambar 4. 51. Koefisien Bhattacharyyya pada pengujian oklusi dengan fitur yang sama



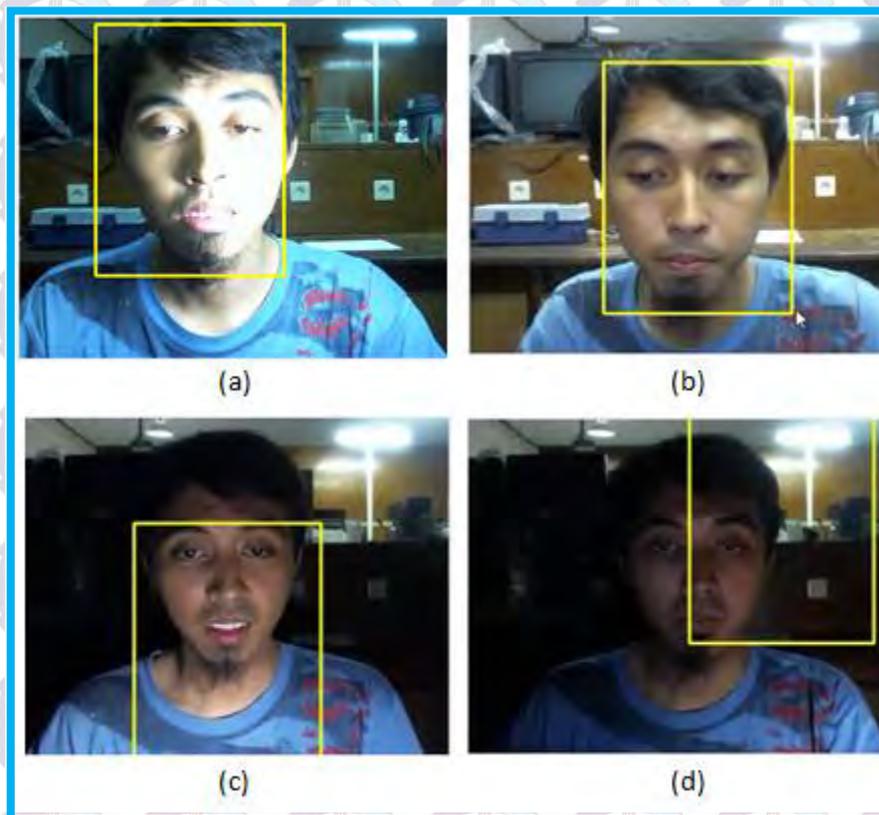
Gambar 4. 52. NCC pada pengujian oklusi dengan fitur yang sama



Gambar 4. 53. Grafik Pengujian oklusi dibelakang

4.11 Pengujian Pengaruh Cahaya terhadap Obyek Target

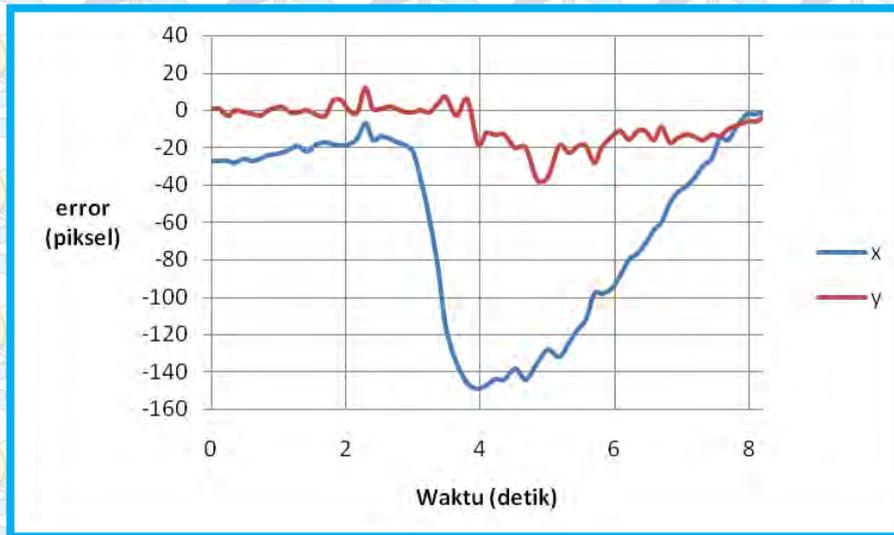
Pada pengujian ini bertujuan untuk mengetahui ketahanan metode yang digunakan terhadap pengaruh intensitas cahaya yang berbeda-beda. Pengujian dilakukan dengan menempatkan sumber cahaya pada objek sebagai target. Kemudian intensitas cahaya tersebut diubah-ubah mulai dari gelap hingga terang. Pada Gambar 4.54 merupakan hasil pengujian dengan menggunakan intensitas cahaya yang berbeda-beda. Berdasarkan pada Gambar 4.54 dapat diketahui bahwa ketika intensitas cahaya bernilai 5499 lux dan 799 lux maka sistem dapat mengikuti wajah ketika intensitas cahaya turun menjadi 50 lux dan 0 lux, maka penjejakan wajah error, terlihat pada Gambar 4.54 (c) dan (d), target wajah bergeser mencari daerah kandidat yang sesuai dengan target model.



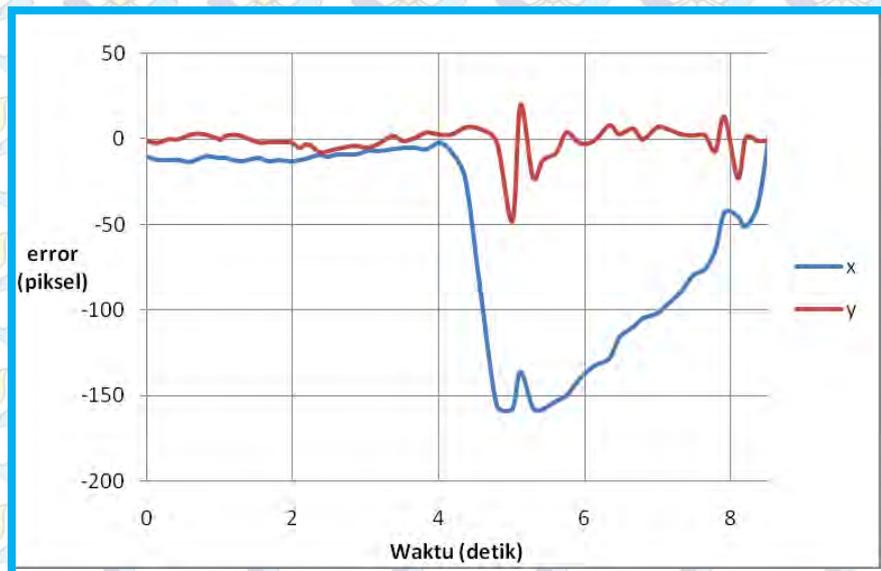
Gambar 4.54. Pengujian terhadap intensitas cahaya yang berbeda-beda (a) intensitas cahaya 5499 lux, (b) intensitas cahaya 799 lux, (c) intensitas cahaya 50 lux, (d) intensitas cahaya 0 lux.

4.12 Pengujian Kecepatan Penjejakan Objek dengan Menggunakan Kamera PTZ

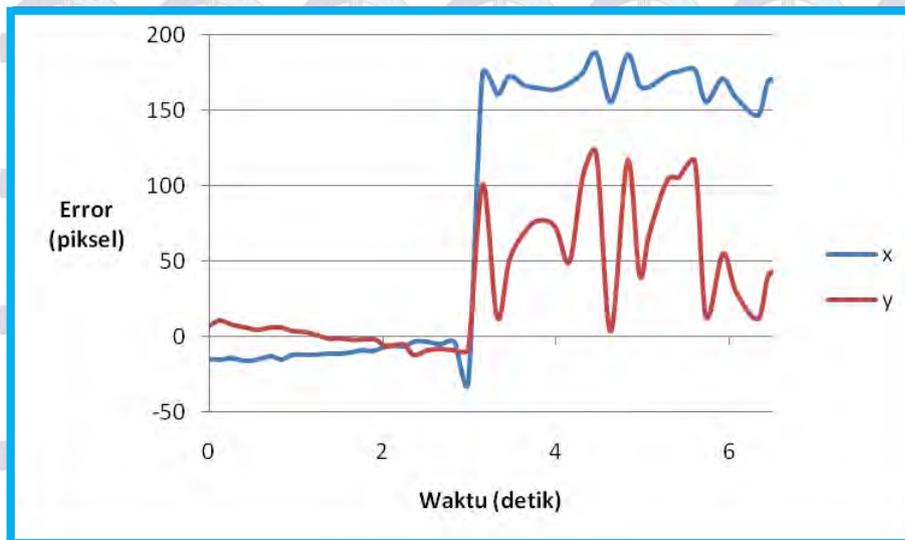
Pengujian ini dilakukan bertujuan untuk mengetahui kecepatan maksimal yang dapat dijangkau oleh kamera PTZ. Pengujian ini dilakukan dengan menjejak benda, dimana kecepatan benda tersebut dapat diatur kecepatannya.



Gambar 4.55. Pengujian penjejakan objek dengan menggunakan kecepatan 94.574 piksel/detik.



Gambar 4.56 Pengujian penjejakan objek dengan menggunakan kecepatan 189.574 piksel/detik



Gambar 4.57. Pengujian penjejakan objek dengan menggunakan kecepatan 536.974 piksel/detik.

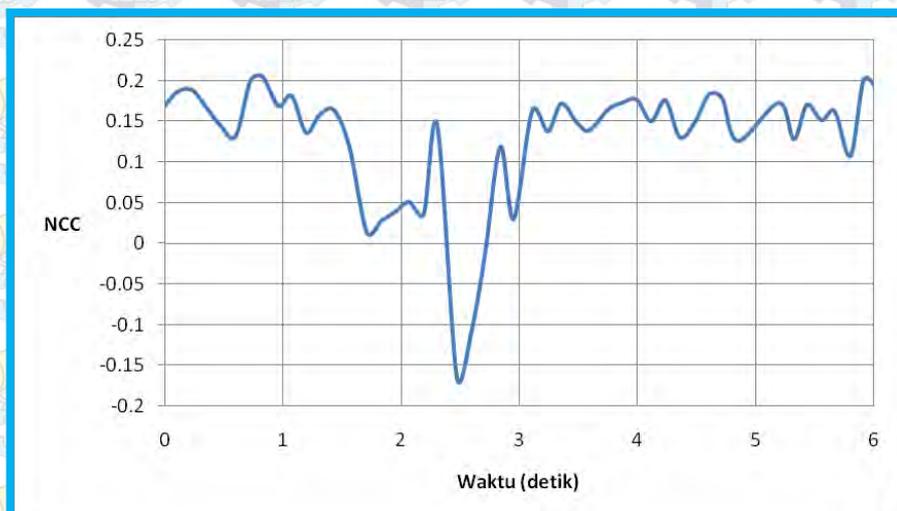
Pada pengujian kecepatan objek yang dapat dijejak dapat diketahui bahwa dari Gambar 4.54 hingga 4.57, pada kecepatan objek 94.574 dan 189.574 piksel/detik kamera PTZ masih mampu untuk menjejak. Hal ini dapat diketahui bahwa error koordinat dari x dan y mengecil menuju 0 piksel. Pada Gambar 4.57 dapat diketahui bahwa kamera PTZ tidak dapat mengikuti target ketika kecepatan objek 536.974 piksel/ detik. Hal ini dapat diketahui bahwa nilai error tidak menuju ke 0 piksel.

4.13 Pengujian dengan Penjejakan Objek dengan menggunakan Objek yang Berbeda.

Pada pengujian ini bertujuan untuk menguji apakah sistem dapat menjejak dengan objek yang berbeda. Pengujian dilakukan dengan cara memilih objek wajah perempuan kemudian objek tersebut dijejak dengan menggunakan kamera PTZ, kemudian objek tersebut dilalui oleh oklusi dengan menggunakan wajah laki-laki. Pada Gambar 4.58 merupakan hasil penjejakan objek dengan menggunakan objek yang berbeda. Pada Gambar tersebut dapat diketahui bahwa sistem dapat menjejak dengan objek yang berbeda dan tahan terhadap adanya oklusi.



Gambar 4.58. Pengujian penjejakan objek yang berbeda



Gambar 4. 59. Hasil NCC pada penjejakan objek pada Gambar 4.58

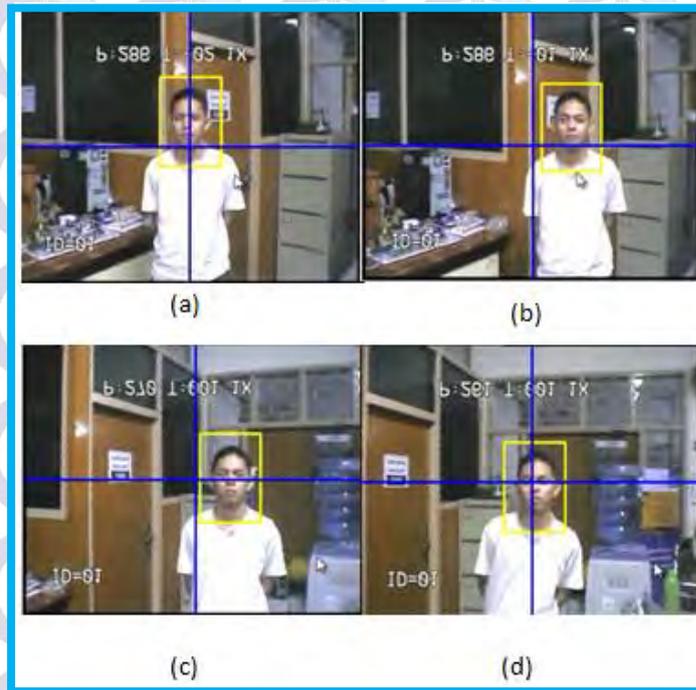
Pembacaan alur pada Gambar 4.58 sesuai dengan huruf abjad. Pada Gambar 4.59 merupakan hasil NCC pada saat oklusi terjadi, dimana nilai NCC dibawah 0.2 terjadi oklusi maka window target akan membesar untuk mencari target model awal.

4.14 Pengujian Penjejakan Objek dengan menggunakan Warna Latar yang Heterogen

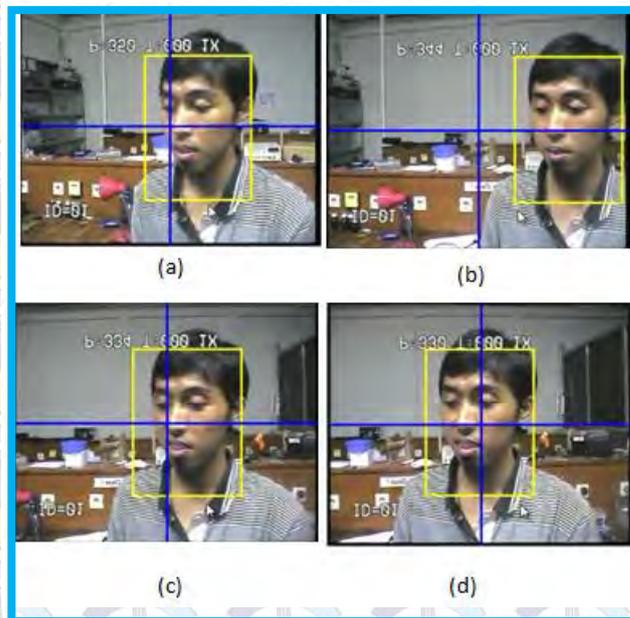
Pada pengujian ini berfungsi untuk menguji pengaruh warna latar objek terhadap objek sebagai target. Pengujian dilakukan dengan background yang berbeda-beda, dimana background memiliki fitur yang sama dengan yang dijejak. Pada Gambar 4.60 terlihat objek yang dijejak berupa wajah, dimana background yang terdapat pada objek adalah meja, kursi yang memiliki fitur warna yang mirip dengan wajah. Pada Gambar 4.60 dapat terlihat bahwa objek target dapat dijejak meskipun objek target berada pada background yang sama.



Gambar 4.60. Pengujian pertama penjejakan objek dengan menggunakan fitur background yang mirip dengan model target.



Gambar 4.61. Pengujian kedua penjejakan objek dengan menggunakan fitur background yang mirip dengan model target.



Gambar 4.62. Pengujian ketiga penjejakan objek dengan menggunakan fitur background yang mirip dengan model target.

BAB 5

PENUTUP

Pada bab ini akan disimpulkan dari hasil pengujian dan pembahasan yang telah didapatkan. Selain itu, pada bab ini juga dikemukakan saran-saran yang perlu dilakukan untuk riset lanjutan dari tesis ini.

5.1 Kesimpulan

Setelah dilakukan pengujian terhadap sistem yang dibangun dengan menggunakan metoda-metoda yang telah ditentukan, dapat disimpulkan bahwa metoda *Improved Mean Shift* dapat dipilih sembarang objek sebagai target, menghilangkan *noise background* dan dapat dilakukan penskalaan. Kecepatan maksimal yang dapat dijejak dengan metode yang digunakan adalah 294.44 piksel/ detik. Dengan menggunakan metode yang digunakan tahan terhadap oklusi dan objek yang berotasi. Pada Kemudian, untuk pengujian pergerakan *pan-tilt*, hasil yang dicapai adalah sistem dapat mencapai *set point* dalam waktu kurang dari 7 detik dengan *error steady state* kurang dari 10%. Dengan adanya parameter Bhattacharyya dan Normalized Cross Correlation maka oklusi pada penjejakan objek dapat diatasi.

5.2 Saran

Berdasarkan hasil kesimpulan maka dapat disarankan untuk pengolahancitra dalam sistem ini, dapat dikembangkan lebih lanjut dengan membangun metoda *tracking* yang bisa tahan terhadap iluminasi cahaya, dan dapat melakukan kontrol zoom. Untuk sistem kontrolnya, disarankan dapat dikembangkan lebih lanjut dengan mempercepat respon sistem terhadap pergerakan obyek dan memperkecil nilai *error steady state* nya.

DAFTAR PUSTAKA

Bradsky, G dan Kaebler, A. (2008), *Learning OpenCV : Computer Vision with the OpenCV Library*, O'Reilly Media, Inc.

Cheng M.H, dkk. (2004), "A Robust Visual Tracking of an Arbitrary-Shaped Object by a New Active Contour Method for a Virtual Reality Application", *Proceedings of IEEE International Conference on Networking, Sensing & Control*, hal. 64-69.

Da ,Tang dan Zhang. (2011), "Combining Mean shift dan Particle filter for object tracking ", *Sixth International Conference on Image and Graphics (ICIG)*, Hefei, Anhui, hal 771-776.

Fahrianur, A. (2013), *Sistem Tracking Obyek Berbasis Visual Servoing Menggunakan Kamera PTZ*. Tesis Magister Institut Teknologi Sepuluh November, Surabaya.

Feng Su dan Gu Fang. (2012), "Moving Object Tracking Using an Adaptive Colour Filter", *12th International Conference on Control, Automation, Robotics & Vision Guangzhou, China*, hal 1048 - 1052

Francois, C dan Hutchinson, S. (2006) "Visual Servo Kontrol Part I: Basic Approaches". *IEEE*, hal. 82 – 90.

G. Bradski,(1998), "Computer vision face tracking for use in a perceptual user interface", *4th IEEE Workshop on Applications of Computer Vision*, hal 214 – 219.

Hutchinson, Seth, dkk. (1996), "A Tutorial on Visual Servo Kontrol", *IEEE Transactions On Robotics And Automation*, Vol. 12, hal.651-670.

J.Ning, et al. (2012), "Robust mean shift Tracking with corrected background-weighted histogram", *IET Computer Vision*, hal .66-69,

J.Ning, et al. (2012), "Scale and Orientation Adaptive Mean Shift Tracking", *IET Computer Vision*, hal 52-61,

Jupp, V. D, dkk. (2007), "Doing Criminological Research", *London: SAGE*, hal.100–103.

Lee, B. Y, dkk. (2014), "Occlusion handling in videos object tracking : A survey", *8th International Symposium of the Digital Earth*.

Nicholson, L.G. (1999), "Security Investigations: A Professional's Guide", *Melbourne: Butterworth-Heinemann*, hal.100–101

Pahani, R dkk. (2013), "Real Time Occlusion Handling Using Kalman Filter and Mean-Shift", *8th Iranian conference on Machine Vision and Image Processing*, Zanzan, hal 320 – 323.

Patel, H. (2013), "Moving Object Tracking Using Kalman Filter", *International Journal of Computer Science and Mobile Computing*.

R.Mehmood, dkk. (2014), "Occlusion Handling in Meanshift Tracking using Adaptive Window Normalized Cross Correlation", *11th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, hal 126-129

Samsung Techwin, *SPD-1000*. Tianjin Samsung Opto-Electronics, China, hal. 1-2, (2000).

Xinghua Li, dkk (2012). "Detection and Tracking of Moving Object Based on PTZ camera", *4th ICCCNT*, Nanjing, hal 493-397

Yilmaz, A. dkk (2006). "Object tracking: A survey", *ACM Comput. Surv.* 38.

Zhang, Miaohui dkk. (2012), "Accurate object tracking based on homography matrix", *International Conference on Computer Science and Service System*, Nanjing, hal 2310-2312.

Zheng, Bin dkk. (2012), "Object Tracking Algorithm Based on Combination of Dynamic Template Matching and Kalman Filter", *4th International Conference on Intelligent Human-Machine Systems and Cybernetics*. Nanchang, Jiangxi, hal 136 – 139.

Zhiwen, Chen dkk. (2011), "Tracking of moving object based on optical flow detection" *International Conference on Computer Science and Network Technology*, Harbin, hal 1096 - 1099

RIWAYAT HIDUP PENULIS



Nama : Sulfan Bagus Setyawan

Alamat : Jl. Mutiara Indah VI no 14 Madiun

Tempat, Tanggal Lahir : Madiun, 24 Juni 1991

Jenis Kelamin : Laki-laki

Agama : Islam

Status : Belum Menikah

No. Telepon : 085606134290

Email : sulfan13@mhs.ee.its.ac.id
bagussetyawansulfan@gmail.com

Riwayat Pendidikan : 1. SDN 02 Oro-oro Ombo Madiun
2. SMPN 1 Madiun
3. SMAN 4 Madiun
4. D4 Politeknik Elektronika Negeri Surabaya