

**PERANGKAT LUNAK PENGENDALI *POINTER* BERBASIS
EYE TRACKING DENGAN MENGGUNAKAN TITIK SUDUT
BAGIAN DALAM MATA SEBAGAI TITIK REFERENSI**



SKRIPSI

**Disusun Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer
Pada Jurusan Informatika/Ilmu Komputer**

Disusun oleh :

Indi Angga Wiguna Kusuma

J2F008107

**JURUSAN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO**

2014

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa dalam tugas akhir / skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Semarang, 29 Agustus 2014



Indi Angga Wiguna Kusuma
J2F008107

HALAMAN PENGESAHAN

Judul : Perangkat Lunak Pengendali *Pointer* Berbasis *Eye Tracking* dengan Menggunakan Titik Sudut Bagian Dalam Mata Sebagai Titik Referensi.

Nama : Indi Angga Wigun Kusuma

NIM : J2F008107

Telah diujikan pada siding tugas akhir pada tanggal 22 Agustus 2014 dan dinyatakan lulus pada tanggal **28 Agustus 2014**.

Semarang, 29 Agustus 2014

Mengetahui,

Sekretaris Jurusan Ilmu Komputer/ Informatika

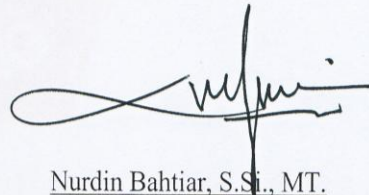


Ragil Saputra, S.Si, M.Cs.

NIP. 1980 10 21 2005 01 1 003

Panitia Penguji Tugas Akhir

Ketua,



Nurdin Bahtiar, S.Si., MT.

NIP. 1979 07 20 2003 12 1 002

HALAMAN PENGESAHAN

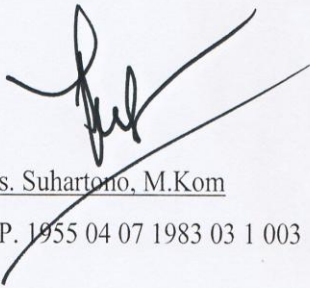
Judul : Perangkat Lunak Pengendali *Pointer* Berbasis *Eye Tracking* dengan
Menggunakan Titik Sudut Bagian Dalam Mata Sebagai Titik Referensi.

Nama : Indi Angga Wigun Kusuma

NIM : J2F008107

Telah diujikan pada sidang tugas akhir pada tanggal 22 Agustus 2014.

Pembimbing Utama

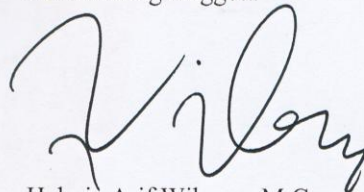


Drs. Suhartono, M.Kom

NIP. 1955 04 07 1983 03 1 003

Semarang, 29 Agustus 2014

Pembimbing Anggota



Helmie Arif Wibawa, M.Cs.

NIP. 1978 05 16 2003 12 1 001

ABSTRAK

Eye tracking ialah salah satu teknologi terkini yang dapat diimplentasikan di berbagai area, salah satunya ialah *human computer interaction* untuk orang yang memiliki cacat tubuh motorik. Pemanfaatan teknologi *eye tracking* umumnya menggunakan perangkat yang mahal dan tidak nyaman digunakan menjadi kendala dalam pemanfaatan teknologi ini. Pada penelitian ini dibangun suatu perangkat lunak pengendali *pointer* berbasis *eye tracking* dengan titik sudut dalam mata menggantikan pantulan sinar inframerah sebagai titik referensi. Pembangunan perangkat lunak dibagi kedalam tiga modul yaitu: pendeteksian koordinat mata, kalibrasi, dan proses penggerakan *pointer*. Proses pendeteksian koordinat mata dilakukan dengan beberapa tahap. Tahapan pertama ialah mendeteksi wajah menggunakan metode *Haar Cascade Clasifier*. Tahap selanjutnya ialah mendeteksi koordinat titik tengah pupil menggunakan metode *Mean of Gradient* dan mendeteksi koordinat sudut mata bagian dalam menggunakan *Subpixel*. Tahap terakhir dalam proses pendeteksian koordinat mata ialah normalisasi koordinat titik tengah pupil dengan titik koordinat dari sudut mata bagian dalam. Proses kalibrasi digunakan untuk menentukan titik acuan yang akan digunakan dalam perhitungan estimasi pandangan yang akan digunakan pada tahap pengendalian *pointer*. Berdasarkan pengujian yang dilakukan, perangkat lunak masih memiliki kekurangan dalam ketepatan penunjukan.

Kata kunci : *Eye Tracking, Haar Cascade Clasifier, Mean of Gradient, Subpixel.*

ABSTRACT

Eye tracking is one of the latest technologies that can be implemented in various areas, one of which is human computer interaction for people with motoric disabilities. Eye tracking technology generally uses expensive device and inconvenient that were made a constraint in the utilization of this technology. This research developed a pointing device software based eye tracking with inner corner of the eye replacing infrared reflection as reference point. Software development was divided into three modules, that is: the detection of eye coordinates, calibration, and pointer movement process. Eye coordinate detection process performed with several stages. The first stage was face detected using Haar Cascade Classifier method. The next stage was detected the coordinates of the midpoint of the pupil using Image Gradient method and detected the coordinates of the inner corner of the eye using corner filtering method. The last stage in the process of detection of eye coordinates were normalized coordinates of the midpoint of the pupil with the coordinates of the inner corner of the eye. Calibration process was used to determine the reference point to be used in the calculation of the estimated view will be used in the *pointer* controlling phase. Depend on the tests has performed, the software still have some weakness at pointing accuracy.

Keywords : Eye Tracking, Haar Cascade Classifier, Mean of Gradient, Subpixel.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyusun tugas akhir yang berjudul “**Perangkat Lunak Pengendali Pointer Berbasis *Eye Tracking* dengan Menggunakan Titik Sudut Bagian Dalam Mata Sebagai Titik Referensi**” untuk mendapatkan gelar sarjana strata satu Jurusan Ilmu Komputer/ Informatika pada Fakultas Sains dan Matematika Universitas Diponegoro (FSM UNDIP).

- 1) Dr. Muhamad Nur, DEA selaku Dekan FSM UNDIP.
- 2) Nurdin Bahtiar, S.Si., MT selaku Ketua Jurusan Ilmu Komputer / Informatika FSM UNDIP.
- 3) Drs. Suhartono, M.Kom selaku pembimbing I dan Helmie Arif Wibawa, M.Cs selaku pembimbing II yang telah meluangkan waktu untuk membimbing dan mengarahkan Penulis dalam menyelesaikan tugas akhir ini.
- 4) Semua pihak yang telah membantu hingga selesainya tugas akhir ini, yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan tugas akhir ini, untuk itu penulis mohon maaf dan mengharapkan saran serta kritik yang membangun dari pembaca. Semoga laporan tugas akhir ini dapat bermanfaat bagi pengembangan ilmu dan pengetahuan, khususnya pada bidang Teknik Informatika.

Semarang, 29 September 2013

Penulis

DAFTAR ISI

	Hal
Halaman Judul	i
Halaman Pernyataan Keaslian Skripsi	ii
Halaman Pengesahan	iii
Abstrak	v
Abstract	vi
Kata Pengantar	vii
Daftar Isi	viii
Daftar Gambar	xii
Daftar Tabel	xiv
Daftar Kode	xv
BAB I PENDAHULUAN	1
1.2. Latar Belakang	1
1.3. Rumusan Masalah	3
1.4. Tujuan dan Manfaat	3
1.5. Ruang Lingkup	4
1.6. Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.2. Mata	6
2.3. Citra Digital	6
2.4. Pengolahan Citra	8
2.4.1. Penskalaan Citra	8
2.4.2. Proses <i>Grayscale</i> pada Citra	9
2.4.3. Binerisasi Citra	10
2.4.4. Equalisasi Histogram	12

2.5.	<i>Computer Vision</i>	14
2.6.	Metode <i>Haar Cascade Clasifier</i>	15
2.7.	Penentuan Lokasi Area Mata	18
2.8.	<i>Means of Gradient</i>	19
2.9.	Deteksi Sudutmata dengan <i>Subpixel</i>	21
2.10.	<i>Kalman Filter</i>	22
	2.10.1. Proses Mengestimasi	22
	2.10.2. Algoritma Kalman Filter	22
2.11.	<i>Eye Tracking System</i>	23
	2.11.1. Eye Tracker.....	24
	2.11.2. Perangkat Lunak <i>Eye tracking</i>	25
	2.11.3. Estimasi Pandangan Menggunakan Sudut Mata.....	26
2.12.	OpenCV	28
2.13.	Pemrograman Berorientasi Objek	28
2.14.	<i>Unified Modeling Language(UML)</i>	29
	2.14.1. <i>Things</i>	29
	2.14.2. <i>Relationship</i>	30
	2.14.3. Diagram	31
2.15.	<i>Unified Process</i>	35
2.16.	Bahasa Pemrograman C++.....	38
BAB III	ANALISIS DAN PERANCANGAN	40
3.1.	Definisi Kebutuhan Perangkat Lunak.....	40
	3.1.1. Deskripsi Umum Perangkat Lunak	40
	3.1.2. <i>Use Case Diagram</i>	44
	3.1.3. Detail <i>Use case Diagram</i>	45
	3.1.4. Kebutuhan <i>Non-fungsional</i> Perangkat Lunak	47
3.2.	Analisis	48

3.2.1. Realisasi <i>Use Case</i> Tahap Analisis	49
3.2.2. Anilysis Class	53
3.3. Perancangan	54
3.3.1. <i>Use case</i> Realization Menangkap Citra dengan <i>Webcam</i>	55
3.3.2. <i>Use case</i> Realization Mendeteksi Koordinat Tatapan Mata.....	56
3.3.3. <i>Use case</i> Realization Mengkalibrasi Koordinat Pandangan.....	57
3.3.4. <i>Use case</i> Realization Menggerakkan <i>Pointer</i>	59
BAB IV IMPLEMENTASI DAN PENGUJIAN	60
4.2. Implementasi	60
4.2.1. Spesifikasi Perangkat Keras	60
4.2.2. Spesifikasi Perangkat Lunak	60
4.2.3. Implementasi <i>Class</i>	60
4.2.4. <i>Eye tracking</i>	61
4.2.4.1. <i>Video Capturing</i>	61
4.2.4.2. <i>Face Detection</i>	62
4.2.4.3. Eye Localization	64
4.2.4.4. Eye Center Detection	64
4.2.4.5. Eye Corner Detection	68
4.2.4.6. <i>Eye Coordinate Normalization</i>	69
4.2.5. Implementasi Calibration	70
4.2.6. Implementasi Pemetaan Titik Pandangan	71
4.2.7. Implementasi Penggerakan Cursor	71
4.2.8. Implementasi Antarmuka	72
4.3. Pengujian.....	76
4.3.1. Lingkungan Pengujian.....	76
4.3.2. Rencana Pengujian	77
4.3.3. Pelaksanaan Pengujian	77

4.3.4. Evaluasi Pengujian.....	78
BAB V PENUTUP.....	79
5.2. Kesimpulan	79
5.3. Saran	79
DAFTAR PUSTAKA	80
Lampiran 1. Deskripsi Hasil Uji.....	83
Lampiran 2. Data Hasil Uji	87

DAFTAR GAMBAR

	Hal
Gambar 2.1. Matriks 8 x 8 piksel	7
Gambar 2.2. Titik grid yang dibutuhkan pada interpolasi bicubic	9
Gambar 2.3. Citra RGB (kiri) dan Citra <i>Grayscale</i> (kanan) (Munir, 2005)	10
Gambar 2.4. Representasi Citra Biner (McAndrew, 2004).....	11
Gambar 2.5. Citra beras yang mengalami <i>thresholding</i> (McAndrew, 2004).....	12
Gambar 2.6. Matriks Citra <i>Grayscale</i> dengan Ukuran 8x8.....	13
Gambar 2.7. Matriks Hasil Equalisasi Histogram	14
Gambar 2.8. Fitur Haar-like	15
Gambar 2.9. Nilai Citra Masukan dengan Matriks 5x5	16
Gambar 2.10. Nilai <i>Integral Image</i> dari Citra Masukan	16
Gambar 2.11. Pencarian Nilai Integral fitur Haar dari Citra	17
Gambar 2.12. Metode AdaBoost Machine-Learning.....	18
Gambar 2.13. Ilustrasi Pendeteksian Titik Tengah (Tim & Erhardt, 2011).....	20
Gambar 2.14. Hasil Persamaan 2.10 Terhadap Citra Mata (Tim & Erhardt, 2011).....	20
Gambar 2.15. Sepasang Filter Sudut Mata.....	21
Gambar 2.16. Algoritma Kalman Filter	23
Gambar 2.17. <i>Head Mounted Device (HMD)</i> (Kurniawan & Suwardi, 2012).....	25
Gambar 2.18. <i>Remote Eye Tracker Device (RED)</i> (Kurniawan & Suwardi, 2012)	25
Gambar 2.19. Ilustrasi Penentuan Titik Tengah pada Mata.....	27
Gambar 2.20. Contoh <i>Use Case Diagram</i> (Shach, 2010)	32
Gambar 2.21. Contoh <i>Activity Diagram</i> (Shach, 2010).....	33
Gambar 2.22. Contoh <i>Class Diagram</i> (Bell, 2004)	33
Gambar 2.23. Software Development Process	35
Gambar 2.24. Hirarki Elemen dalam <i>Unified Process</i>	35

Gambar 2.25. Fase dalam Unified Process	36
Gambar 3.1. <i>Activity Diagram</i> dari Aplikasi Penggerak <i>Pointer</i>	43
Gambar 3.2. Diagram <i>Use case</i> dari perangkat lunak	44
Gambar 3.3. Antarmuka Utama Perangkat Lunak.....	45
Gambar 3.4. Antarmuka Kalibrasi.....	47
Gambar 3.5. Diagram <i>Analysis Class</i> Menangkap Citra.....	49
Gambar 3.6. <i>Collaboration Diagram</i> Menangkap Citra.....	50
Gambar 3.7. <i>Analysis Class Diagram</i> Mendeteksi Koordinat Tatapan Mata	50
Gambar 3.8. <i>Collaboration Diagram</i> Mendeteksi Koordinat Tatapan Mata.....	51
Gambar 3.9. <i>Analysis Class Diagram</i> Mengkalibrasi Koordinat Pandangan	51
Gambar 3.10. <i>Collaboration Diagram</i> Mengkalibrasi Koordinat Pandangan	52
Gambar 3.11. <i>Analysis Class Diagram</i> Menggerakkan <i>Pointer</i>	52
Gambar 3.12. <i>Collaboration diagram</i> Menggerakkan <i>Pointer</i>	53
Gambar 3.13. <i>Class Diagram</i> Menangkap Citra	55
Gambar 3.14. <i>Sequence Diagram</i> Menangkap Citra	56
Gambar 3.15. <i>Class Diagram</i> Koordinat Tatapan Mata	56
Gambar 3.16. <i>Sequence Diagram</i> Koordinat Tatapan Mata	57
Gambar 3.17. <i>Class Diagram</i> Kalibrasi Pandangan	58
Gambar 3.18. <i>Sequence Diagram</i> Kalibrasi Pandangan	58
Gambar 3.19. <i>Class Diagram</i> Menggerakkan <i>Pointer</i>	59
Gambar 3.20. <i>Sequence Diagram</i> Menggerakkan <i>Pointer</i>	59
Gambar 4.1. Antarmuka Aplikasi Pengendai <i>Pointer</i>	72
Gambar 4.2. Kondisi Kamera Tidak Ditemukan	74
Gambar 4.3. Jendela Kalibrasi.....	75
Gambar 4.4. Kondisi Antarmuka Setelah Proses Kalibrasi	75

DAFTAR TABEL

	Hal
Tabel 2.1. Jumlah Intensitas Warna	13
Tabel 2.2. CDF (Fungsi Distribusi Kumulatif)	13
Tabel 2.3. Persamaan <i>Time Update</i>	23
Tabel 2.4. Persamaan <i>Measurement Update</i>	23
Tabel 2.5. Jenis-jenis <i>Relationship</i>	30
Tabel 2.6. Notasi-notasi pada <i>Use Case Diagram</i>	31
Tabel 2.7. Notasi-notasi pada <i>Activity Diagram</i>	32
Tabel 2.8. Notasi-notasi pada <i>Class Diagram</i>	34
Tabel 2.9. Notasi-notasi pada <i>Sequence Diagram</i>	34
Tabel 2.10. Jenis-jenis <i>Analysis Class</i>	38
Tabel 3.1. Daftar <i>Use case</i>	44
Tabel 3.2. <i>Use case</i> Detail Menangkap Citra	45
Tabel 3.3. <i>Use case</i> detail Mendeteksi Koordinat Tatapan Mata	46
Tabel 3.4. <i>Use case</i> Detail Mengkalibrasi Koordinat Pandangan	46
Tabel 3.5. <i>Use case</i> Detail Menggerakkan <i>Pointer</i>	47
Tabel 3.6. Hasil Identifikasi <i>Anilysis Class</i>	53
Tabel 3.7. Tanggung Jawab dan Atribut <i>Anilysis Class Diagram</i>	54
Tabel 4.1. Implementasi <i>Class</i>	61
Tabel 4.2. Detail Komponen pada Antarmuka Aplikasi	73
Tabel 4.3. Rencana Pengujian	77

DAFTAR KODE

	Hal
Kode 2.1. Menampilkan Kalimat “ <i>Hello World</i> ” Menggunakan C++	39
Kode 4.1. Video Capturing pada Fungsi Utama pada <i>Class EyeTrackingThread</i>	62
Kode 4.2. Proses Pendeteksian Wajah	63
Kode 4.3. Penetapan Lokasi Area Mata	64
Kode 4.4. Fungsi <i>getEyeCenterCoordinate</i>	65
Kode 4.5. Fungsi <i>computMatXGradient</i>	66
Kode 4.6. Fungsi <i>matrixMagnitude</i>	67
Kode 4.7. Fungsi <i>computeDynamicThreshold</i>	67
Kode 4.8. Fungsi <i>testPossibleCentersFormula</i>	67
Kode 4.9. Fungsi <i>findEyeCorner</i> pada <i>class eyeCorner</i>	68
Kode 4.10. Fungsi <i>eyeCornerMap</i>	68
Kode 4.11. <i>CornerFilter</i>	69
Kode 4.12. Fungsi <i>findSubpixelEyeCorner</i>	69
Kode 4.13. Normalisasi Koordinat Tatapan	70
Kode 4.14. <i>RunCalibration</i> ialah Fungsi Utama dari <i>Class Calibration</i>	71
Kode 4.15. Fungsi <i>gazeEstimate</i>	71
Kode 4.16. Fungsi <i>setPosition</i>	72

BAB I

PENDAHULUAN

Bab ini memaparkan latar belakang, rumusan masalah, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan tugas akhir dengan judul “Perangkat Lunak pengendalian *pointer* berbasis *eye tracking* dengan Menggunakan Titik Sudut Bagian Dalam Mata Sebagai Titik Referensi.

1.2. Latar Belakang

Perubahan bentuk sistem layanan dari manual ke elektronik (*e-service*) menjadikan penggunaan komputer menjadi suatu kebutuhan untuk mendapatkan informasi. Penggunaan komputer memiliki hubungan yang sangat erat sekali dengan cara pengguna berinteraksi dengan komputer. Salah satu jenis interaksi yang sangat sering dilakukan dalam penggunaan komputer ialah pengendalian *pointer*. Penelitian mengenai pengendalian *pointer* dewasa ini semakin banyak dikembangkan. Perangkat pengendali *pointer* yang umum digunakan ialah *mouse*, *trackball*, *touchpad*, *tauchscreen*, *pointing stick*, *joystick*, dan lain sebagainya. Perangkat pengendali *pointer* yang banyak ditemukan umumnya menggunakan gerakan tangan sebagai pengendali *pointer*.

Perangkat pengendali *pointer* berbasis gerakan tangan sangat mudah digunakan bagi pengguna normal (tidak menderita cacat tangan), namun untuk sebagian orang yang memiliki cacat fisik (khususnya pada bagian motorik tangan) menggunakan alat pengendali *pointer* menggunakan gerakan tangan, sangat sulit sekali dilakukan. Sehingga dibutuhkan suatu teknologi yang dapat menggantikan gerakan tangan sebagai pengendali *pointer*. Salah satu solusi yang dapat ditawarkan adalah dengan memanfaatkan gerakan mata untuk menggerakkan *pointer*. Teknologi ini sering disebut *eye tracking*.

Eye tracking adalah adalah terapan ilmu dari *computer vision* untuk mengukur gerakan mata seseorang sehingga komputer dapat mengetahui lokasi mana yang ditatap oleh orang tersebut (Duchowski, 2007). Perangkat untuk mengukur gerakan mata disebut *eye tracker*. Salah satu jenis *eye tracker* yang mudah digunakan ialah *eye tracker* berbasis video dengan menggunakan *webcam* untuk menangkap gerakan mata. *Eye tracker* jenis ini dibagi kedalam dua jenis perangkat yaitu *head mounted device (HMD)* dan *remote eye tracker device (RED)*. HMD ialah perangkat *eye*

tracker yang cara mengoprasikannya digunakan di kepala. *Eye tracker* jenis HMD memiliki kekurangan dalam hal kepraktisan dan kenyamanan. Perangkat *eye tracker* jenis RED lebih nyaman dan praktis digunakan karena tidak melakukan kontak fisik secara langsung antara perangkat dan pengguna (Duchowski, 2007). Proses penentuan lokasi mata pada penggunaan RED umumnya menggunakan pendekatan *multi-stage*. *Multi-stage* ialah proses penentuan lokasi mata dengan melalui beberapa tahap yang dimulai dari pencarian lokasi wajah (Cristinacce et al., 2004). Lokasi fitur mata yang digunakan akan digunakan dalam menentukan nilai estimasi pandangan mata.

Salah satu teknik untuk mendapatkan nilai estimasi pandangan mata ialah dengan penyinaran sinar inframerah. Teknik tersebut memungkinkan didapat hasil yang akurat karena menghasilkan nilai kontras yang tinggi antara pupil dan iris sehingga memudahkan pencarian titik tengah mata. Penggunaan sinar inframerah juga dapat menghasilkan pantulan sinar pada kornea yang dapat dijadikan titik referensi dalam melakukan perhitungan nilai estimasi pandangan mata. Meskipun teknik tersebut memungkinkan hasil yang akurat tetapi memiliki beberapa kekurangan, diantaranya ialah tidak cocok bila digunakan di luar ruangan pada siang hari, dan hanya terdapat pada *eye tracker* yang bersifat komersil. Selain dengan menggunakan pencahayaan inframerah, pencarian nilai estimasi pandangan juga masih dimungkinkan menggunakan pencahayaan normal.

Penelitian mengenai *eye tracking* dengan menggunakan pencahayaan normal pernah dilakukan oleh Kurniawan (Kurniawan & Suwardi, 2012). Pada penelitian tersebut, titik sudut mata digunakan sebagai titik referensi yang menggantikan pantulan sinar inframerah pada kornea. Tahapan proses penentuan nilai estimasi gerakan dilakukan melalui beberapa tahapan. Tahapan tersebut meliputi : deteksi wajah, deteksi koordinat iris mata, dan deteksi koordinat sudut mata. Metode yang digunakan pada ketiga tahapan tersebut ialah: *haar-like feature*(Viola & Jones, 2001) untuk melakukan deteksi wajah, *harris corner detector* untuk mendeteksi koordinat sudut mata , dan metode *hough circular transformation* untuk mendeteksi koordinat titik tengah mata (Kurniawan & Suwardi, 2012).

Keakuratan dalam mendeteksi koordinat lokasi titik tengah mata dan koordinat sudut mata sangat berpengaruh dalam estimasi pandangan yang didapat. Penelitian mengenai lokasi titik tengah dari iris mata yang dilakukan oleh Fabian Timm (Fabian

Timm, 2011) menggunakan metode *mean of gradient*. Berdasarkan data hasil perbandingan nilai akurasi penentuan titik tengah mata dengan metode lainnya, metode tersebut memiliki akurasi yang lebih baik dari metode *hough circular transformation* (Fabian Timm, 2011).

Selain keakuratan pendeteksian koordinat mata, keakuratan mendeteksi koordinat sudut mata juga diperlukan dalam membuat program perangkat lunak *eye tracking*. Penggunaan metode *harris corner detector* akan menghasilkan akurasi pendeteksian koordinat sudut mata yang kurang baik apabila dilakukan pada kondisi pencahayaan yang kurang baik. Menurut penelitian mengenai pendeteksian koordinat sudut mata yang dilakukan oleh Zhu (Zhu & Yang, 2002) dengan menggunakan metode *Subpixel Eye Corner Tracking* dapat menghasilkan koordinat sudut mata yang lebih stabil dan akurat (Zhu & Yang, 2002).

Pada tugas akhir ini akan dibangun suatu aplikasi perangkat lunak untuk mengendalikan *pointer* berbasis *eye tracking* dengan titik sudut mata bagian dalam sebagai titik referensi, dengan menggunakan metode *mean of gradient* untuk mendeteksi koordinat titik tengah mata, dan metode *subpixel eye corner tracking* sebagai pendeteksi koordinat titik sudut mata bagian dalam.

1.3. Rumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang, rumusan masalah yang akan dibahas dalam tugas akhir ini adalah bagaimana membangun perangkat lunak untuk mengendalikan *pointer* berbasis *eye tracking* dengan menggunakan titik sudut bagian dalam mata sebagai titik referensi.

1.4. Tujuan dan Manfaat

Tujuan yang ingin dicapai dalam penulisan tugas akhir ini adalah menghasilkan sebuah perangkat lunak pengendali *pointer* berbasis *eye tracking* dengan menggunakan titik sudut bagian dalam mata sebagai titik referensi.

Adapun manfaat dari penelitian tugas akhir ini adalah sebagai berikut:

1. Membantu penyandang cacat motorik dalam berinteraksi dengan komputer.
2. Dari sisi ekonomi perangkat lunak dengan menggunakan metode *eye tracking* ini dapat memberikan kemudahan biaya yang lebih terjangkau bagi pengguna.

3. Dapat menjadi referensi atau bahan acuan pada penelitian dengan topik yang sama.

1.5. Ruang Lingkup

Dalam penyusunan tugas akhir ini, diberikan ruang lingkup yang jelas agar pembahasan lebih terarah dan tidak menyimpang dari tujuan penulisan. Ruang lingkup perangkat lunak pengendali *pointer* menggunakan media *webcam* ini adalah sebagai berikut:

1. Penelitian ini menggunakan piranti *webcam* sebagai penangkap citra masukan.
2. Perangkat lunak pengendali *pointer* pada penelitian ini hanya ditujukan pada proses penggerakan *pointer*.
3. Penelitian ini bertujuan untuk menghasilkan perangkat lunak penggerak *pointer* yang hanya bekerja pada posisi wajah pengguna pada keadaan diam.
4. Perangkat lunak dibangun menggunakan bahasa pemrograman C++.
5. Perangkat lunak menggunakan *library* OpenCV 2.4.8 untuk menerapkan fungsi-fungsi yang berkaitan dengan *computer vision*.
6. Penelitian ini hanya mengujikan perangkat lunak pada kondisi pencahayaan normal dengan nilai intensitas cahaya 150-200lux.

1.6. Sistematika Penulisan

Sistematika yang digunakan dalam penulisan tugas akhir ini adalah:

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Bab ini berisi penjelasan mengenai konsep-konsep yang mendukung pembuatan perangkat lunak seperti mata, pengolahan citra digital, metode *Haar cascade clasifier*, metode *means of gradient*, *Subpixel*, *kalman filter* dan *opencv*.

BAB III ANALISIS DAN PERANCANGAN

Bab ini berisi tentang kebutuhan perangkat lunak untuk menganalisis masalah yang ada dalam pembangunan perangkat lunak, pemodelan

terstruktur untuk memperlihatkan keterkaitan antar fungsi yang terdapat dalam perangkat lunak serta perancangan antarmuka perangkat lunak.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari hasil analisis dan perancangan, serta hasil pengujian perangkat lunak.

BAB V KESIMPULAN

Bab ini berisi kesimpulan yang dapat ditarik dan saran-saran yang dapat diberikan untuk pengembangan perangkat lunak lebih lanjut.