

SKRIPSI

**MODIFIKASI *BLOB ANALYSIS* UNTUK MENDETEKSI
GERAKAN DAN PERPINDAHAN TITIK PUSAT API**

Disusun dan diajukan oleh:

JABALNUR

D121181313



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

2023

LEMBAR PENGESAHAN SKRIPSI
MODIFIKASI *BLOB ANALYSIS* UNTUK MENDETEKSI GERAKAN DAN
PERPINDAHAN TITIK PUSAT API

Disusun dan diajukan oleh

JABALNUR

D121181313

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 31 Januari 2023 dan dinyatakan
telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,


Prof. Dr. Ir. Indrabayu, S.T., M.T.
Nip. 197507162002121004


Anugrayani Bustamin, S.T., M.T.
Nip. 199012012018074001

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu, S.T., M.T.
Nip. 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Jabalnur

NIM : D121181313

Departemen : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

“MODIFIKASI *BLOB ANALYSIS* UNTUK MENDETEKSI GERAKAN DAN
PERPINDAHAN TITIK PUSAT API ”

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 31 Januari 2023

Yang menyatakan,



Jabalnur

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas berkat, rahmat, dan karunia-Nya sehingga tugas akhir yang berjudul “Modifikasi *Blob Analysis* untuk Mendeteksi Gerakan dan Perpindahan Titik Pusat Api” dapat diselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan tugas akhir ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak, dari masa perkuliahan hingga penyusunan tugas akhir. Oleh sebab itu, penulis menyampaikan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Jamaluddin S, S.Pd., dan Ibu Hj. Nurbaya Nur yang tidak pernah lelah dalam mendidik dan memberikan dukungan, doa, serta semangat kepada penulis
2. Bapak Prof. Dr. Ir. Indrabayu, ST., M.T., M.Bus.Sys., IPM, ASEAN. Eng., selaku Ketua Departemen Teknik Informatika sekaligus Dosen Pembimbing Akademik dan Dosen Pembimbing I Tugas Akhir penulis yang senantiasa membimbing dan menyediakan waktu, tenaga, dan perhatiannya di tengah kesibukan yang beliau sedang hadapi di sepanjang masa perkuliahan.
3. Ibu Anugrayani Bustamin, ST., M.T, selaku Dosen Pembimbing II Tugas Akhir penulis yang tiap saat membimbing dan menyediakan

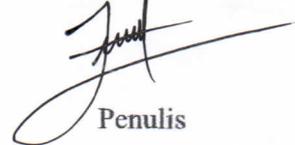
waktu, tenaga, dan perhatiannya di tengah kesibukan selama penyusunan tugas akhir ini.

4. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu semasa perkuliahan dan dalam penyelesaian tugas akhir.
5. Saudara-saudara dari BM2 atas segala dukungan, bantuan, dan semangat yang tak henti diberikan selama ini.
6. Teman-teman Synchronous18 atas dukungan, bantuan, semangat, serta pengalamannya yang telah diberikan selama ini.
7. Sahabat-sahabat dari Kabinet Jokka IPPMP-UH yang telah memberikan bantuan, semangat, waktu, serta segala macam hal yang telah membawa penulis hingga bisa menjadi seperti sekarang ini.
8. Saudara Magfirah, Fandly, Dandy, Andika, Salahuddin, dan Darul yang telah memberi dukungan dan bantuan dalam menyusun dan menyelesaikan tugas akhir.
9. Kanda-kanda senior dari Lab AI yang telah memberikan masukan dan tambahan ilmu yang menunjang penyelesaian tugas akhir.
10. Saudara Asnad Primananda, Fitrah Amalia Nurdin, Rahma Kahar, dan Muhammad Rijal, yang telah memberikan bantuan berupa harta, tenaga, dan waktu dalam pengumpulan data tugas akhir.
11. Seluruh pihak yang tidak sempat disebutkan dan tanpa sadar telah menjadi inspirasi serta banyak membantu penulis dalam menyelesaikan tugas akhir.

12. *Last but not least, I wanna thank me. I wanna thank me for believing in me. I wanna thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for, for never quitting. I wanna thank me for just being me at all times.*

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karenanya diharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat memberikan sumbangsih dan manfaat besar bagi kehidupan manusia.

Gowa, 1 Januari 2023



Penulis

ABSTRAK

Tunanetra sangat membutuhkan kontribusi teknologi untuk dapat membantu mobilitas mereka dari berbagai bahaya di sekitar. Di Indonesia sendiri masih menjadi kebiasaan masyarakat untuk melakukan pembakaran sampah di lingkungan masyarakat, di mana api pada pembakaran ini dapat membahayakan tunanetra. Sebelumnya telah ada penelitian yang menemukan cara untuk membantu tunanetra dalam mendeteksi keberadaan api menggunakan algoritma *blob analysis* dengan melakukan analisis pada perubahan area api di setiap *frame*. Akan tetapi masih diperlukan proses pengembangan agar algoritma dapat mendeteksi dengan lebih akurat. Penelitian ini dilakukan dengan memberikan modifikasi kepada algoritma *blob analysis* untuk dapat mendeteksi gerakan dan perpindahan titik pusat api agar objek bukan api yang memiliki fitur warna seperti api tidak lagi terdeteksi sebagai api. Bentuk modifikasi yang dilakukan adalah analisis blob bertingkat, yaitu penambahan *blob labeling* untuk dapat memungkinkan terjadinya *multi-detection*, analisis perpindahan jarak titik pusat objek, dan analisis perubahan area di masing-masing objek, kemudian penentuan apakah objek yang dideteksi adalah api berdasarkan dua hasil analisis sebelumnya. Untuk pengestimasi jarak api menggunakan metode *pinhole camera model*. Hasil dari penelitian ini menunjukkan bahwa dengan menerapkan modifikasi *blob analysis* pada sistem deteksi api mampu membuat objek cahaya lampu kendaraan tidak lagi terdeteksi sebagai objek api dengan tingkat akurasi 94.63%. Sedangkan sistem estimasi jarak objek api menggunakan *pinhole camera model* mampu mendeteksi api dengan sangat baik dengan nilai *Mean Square Error* sebesar 0.0014.

Kata Kunci: *blob analysis*, tunanetra, *fire detection*, *multi detection*, *pinhole model*

DAFTAR ISI

KATA PENGANTAR	iv
ABSTRAK	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	4
1.5. Batasan Masalah	4
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1. Tunanetra	6
2.2. Citra	7
2.3. Pengolahan Citra	10
2.4. Pengolahan Video	24
2.5. <i>Blob Analysis</i>	25
2.6. <i>Pinhole Model</i>	26
2.7. Penelitian Terkait	29
BAB III METODOLOGI PENELITIAN	31
3.1. Tahapan Penelitian	31
3.2. Waktu dan Lokasi Penelitian	33
3.3. Instrumen Penelitian	33
3.4. Perancangan Sistem	33
3.4.1. Pembacaan Video.....	40
3.4.2. <i>Pre-processing</i>	40
3.4.3. Segmentasi HSV	41
3.4.4. Morfologi	43
3.4.5. <i>Blob Labeling (Modifikasi <i>Blob Analysis</i>)</i>	45

3.4.6. Analisis Perpindahan Titik Pusat Objek (Modifikasi <i>Blob Analysis</i>)	51
3.4.7. Analisis Perubahan Besar Area Objek (Modifikasi <i>Blob Analysis</i>)...	56
3.4.8. Pengujian <i>Threshold</i> pada Kendaraan terhadap Beberapa Kecepatan Berbeda	67
3.4.9. Perhitungan Jarak dengan <i>Pinhole Model</i>	70
3.4.10. Keputusan.....	72
3.5. Teknik Pengambilan Data	73
3.5.1. Terdapat Api Di Depan Tunanetra	74
3.5.2. Terdapat Kendaraan Di Depan Tunanetra.....	74
3.5.3. Terdapat Api dan Kendaraan Di Depan Tunanetra.....	75
3.5.4. Skenario Pengujian <i>Threshold</i> terhadap Kecepatan Kendaraan Berbeda	78
3.6. Analisis Kinerja Sistem.....	79
BAB IV HASIL DAN PEMBAHASAN.....	81
4.1. Deteksi Api	81
4.1.1. Objek Di Depan Tunanetra	81
4.1.2. Jenis Kendaraan.....	85
4.1.3. Warna Lampu Kendaraan.....	89
4.1.4. Pergerakan Tunanetra.....	93
4.1.6. Pergerakan Kendaraan.....	96
4.1.7. Jarak Api	100
4.1.8. Keseluruhan Skenario	106
4.2. Pengukuran Estimasi Jarak	112
BAB V PENUTUP.....	118
5.1. Kesimpulan	118
5.2. Saran.....	118
DAFTAR PUSTAKA	119
LAMPIRAN.....	122

DAFTAR GAMBAR

Gambar 1. (a) Citra Biner. (b) Citra <i>Grayscale</i> . (c) Citra RGB	10
Gambar 2. Ruang Warna RGB	14
Gambar 3. Visualisasi HSV <i>Color Model</i>	16
Gambar 4. Visualisasi proses pencarian batas/kontur	18
Gambar 5. Visualisasi operasi dilasi.....	19
Gambar 6. Visualisasi operasi erosi	20
Gambar 7. Visualisasi operasi <i>closing</i>	21
Gambar 8. Visualisasi operasi <i>opening</i>	22
Gambar 9. Visualisasi operasi <i>filling</i>	24
Gambar 10. <i>Binary Large Object</i>	26
Gambar 11. (a) Pengambilan gambar dengan <i>Pinhole model</i> . (b) Model matematika dari pengambilan gambar <i>Pinhole model</i>	27
Gambar 12. (a) Ukuran jarak dalam piksel. (b) Besar sudut pengambilan video	28
Gambar 13. Bagan tahap penelitian.....	31
Gambar 14. Skema proses sistem secara umum.....	34
Gambar 15. <i>Flowchart</i> usulan sistem.....	35
Gambar 16. Perbedaan algoritma <i>blob analysis</i> sebelum dan setelah modifikasi	38
Gambar 17. Gambaran umum sistem	39
Gambar 18. <i>Cropping</i> gambar dari resolusi 1920 x 1080 rasio 16:9 menjadi 810 x 1080 rasio 3:4.....	40
Gambar 19. <i>Resize</i> gambar dari 810 x 1080 menjadi 480 x 640	41
Gambar 20. Segmentasi HSV	42
Gambar 21. Hasil dari proses morfologi	45
Gambar 22. Tampak api dan cahaya lampu kendaraan.....	52
Gambar 23. Terdapat kayu tidak terbakar yang menghalangi citra dan citra masih menyatu	64
Gambar 24. Terdapat kayu tidak terbakar yang menghalangi citra dan citra menjadi terpisah.....	65
Gambar 25. Terdapat kayu tidak terbakar yang menghalangi citra, citra menjadi terpisah dan api kecil menjadi tidak terdeteksi.....	65
Gambar 26. Perbandingan jarak aktual dengan jarak dalam piksel.....	71
Gambar 27. Tampak objek terdekat dengan kamera berdasarkan nilai sumbu y-nya.....	71
Gambar 28. Hasil keluaran objek dideteksi sebagai api dengan <i>Bounding Box</i> dan informasi jarak	72
Gambar 29. Ilustrasi penggunaan povie pada tunanetra.....	73
Gambar 30. Ilustrasi terdapat api di depan tunanetra	74

Gambar 31. Ilustrasi terdapat kendaraan di depan tunanetra.....	74
Gambar 32. Ilustrasi terdapat api dan kendaraan di depan tunanetra	75
Gambar 33. <i>Bar Chart</i> akurasi pendeteksian api pada skenario objek di depan tunanetra.....	85
Gambar 34. <i>Bar Chart</i> akurasi pendeteksian api pada skenario jenis kendaraan.....	89
Gambar 35. <i>Bar Chart</i> akurasi pendeteksian api pada skenario warna lampu kendaraan.....	93
Gambar 36. <i>Bar Chart</i> akurasi pendeteksian api pada skenario pergerakan tunanetra.....	96
Gambar 37. <i>Bar Chart</i> akurasi pendeteksian api pada skenario pergerakan kendaraan.....	100
Gambar 38. <i>Bar Chart</i> akurasi pendeteksian api pada skenario jarak api ke tunanetra.....	106
Gambar 39. <i>Bar chart</i> akurasi pendeteksian api pada seluruh skenario.....	107
Gambar 40. Hasil deteksi api.....	108
Gambar 41. Objek api terdeteksi sebagai api dan objek cahaya lampu kendaraan terdeteksi sebagai objek bukan api	109
Gambar 42. Objek cahaya lampu kendaraan yang sama sekali tidak terdeteksi sebagai objek.....	110
Gambar 43. <i>Blob</i> yang tercipta dari cahaya lampu kendaraan hanya berada di bagian pinggir dari lampu kendaraan.....	111
Gambar 44. Grafik jarak dan rata-rata tingkat <i>error</i> estimasi sistem.....	115
Gambar 45. Hasil pengukuran jarak objek api	116
Gambar 46. Jarak yang diestimasi oleh sistem pada saat tunanetra bergerak dari jarak 6-meter menuju api malah menghasilkan jarak di atas 6 meter.....	116

DAFTAR TABEL

Tabel 1. Asumsi objek dengan area, <i>centroid</i> , dan <i>bounding box</i>	50
Tabel 2. Ilustrasi pendeteksian objek sebelum dan setelah <i>labeling</i>	50
Tabel 3. Ilustrasi perhitungan jarak titik pusat antara frame pertama dan terakhir dengan nilai asumsi.....	53
Tabel 4. Sampel perubahan titik pusat objek api dan lampu kendaraan pada sumbu x.....	54
Tabel 5. Sampel perubahan titik pusat objek api dan lampu kendaraan pada sumbu y.....	55
Tabel 6. Analisis nilai <i>threshold</i> jarak perpindahan api	55
Tabel 7. Ilustrasi perhitungan jumlah perubahan besar area objek dari frame pertama hingga terakhir dengan nilai asumsi	58
Tabel 8. Sampel perubahan besar area objek api dan cahaya lampu kendaraan ..	59
Tabel 9. Analisis nilai <i>threshold</i> perubahan area api	59
Tabel 10. Ilustrasi perhitungan jumlah perubahan area <i>frame</i> pada penelitian sebelumnya dengan nilai asumsi di mana area semua objek dijumlahkan	60
Tabel 11. Ilustrasi perhitungan jumlah perubahan area <i>frame</i> pada penelitian saat ini dengan nilai asumsi di mana area semua objek dipisahkan.....	61
Tabel 12. Ilustrasi mengganti nilai 0 menjadi nilai yang sama pada <i>frame</i> sebelumnya dengan nilai asumsi.....	62
Tabel 13. Ilustrasi menghilangkan nilai 0 dan menghapus seluruh area objek jika terdapat nilai 0 lebih dari setengah jumlah <i>frame</i> dengan nilai asumsi	63
Tabel 14. Ilustrasi penggabungan <i>output</i> untuk seluruh objek dengan nilai asumsi	66
Tabel 15. Ilustrasi pemisahan <i>output</i> untuk masing-masing objek dengan nilai asumsi	66
Tabel 16. Sampel perubahan titik pusat objek cahaya lampu kendaraan pada sumbu x.....	67
Tabel 17. Sampel perubahan titik pusat objek cahaya lampu kendaraan pada sumbu y.....	68
Tabel 18. Sampel perubahan besar area objek cahaya lampu kendaraan.....	68
Tabel 19. Hasil pengujian <i>threshold</i> titik pusat dan total perubahan area objek lampu kendaraan	69
Tabel 20. Skenario pengambilan data terdapat api di depan tunanetra	74
Tabel 21. Skenario pengambilan data terdapat kendaraan di depan tunanetra.....	75
Tabel 22. Skenario pengambilan data terdapat api dan kendaraan di depan tunanetra.....	76
Tabel 23. Skenario pengambilan data untuk pengujian <i>threshold</i> terhadap kecepatan berbeda.....	78

Tabel 24. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario objek di depan tunanetra adalah api.....	81
Tabel 25. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario objek di depan tunanetra adalah kendaraan	82
Tabel 26. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario objek di depan tunanetra adalah api dan kendaraan	83
Tabel 27. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario objek di depan tunanetra	84
Tabel 28. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jenis kendaraan adalah tanpa kendaraan	85
Tabel 29. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jenis kendaraan adalah mobil	86
Tabel 30. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jenis kendaraan adalah motor	87
Tabel 31. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jenis kendaraan	88
Tabel 32. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario warna lampu kendaraan adalah tanpa lampu kendaraan	89
Tabel 33. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario warna lampu kendaraan adalah lampu berwarna kuning.....	90
Tabel 34. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario warna lampu kendaraan adalah lampu berwarna putih	91
Tabel 35. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario warna lampu kendaraan	92
Tabel 36. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan tunanetra adalah tunanetra diam	93
Tabel 37. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan tunanetra adalah tunanetra bergerak	95
Tabel 38. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan tunanetra.....	96
Tabel 39. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan kendaraan adalah tanpa kendaraan	96
Tabel 40. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan kendaraan adalah kendaraan diam	97
Tabel 41. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan kendaraan adalah kendaraan bergerak	98
Tabel 42. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario pergerakan kendaraan	100
Tabel 43. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah tanpa api	100
Tabel 44. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah 3-meter dari api	101

Tabel 45. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah 4-meter dari api	102
Tabel 46. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah 5-meter dari api	103
Tabel 47. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah 6-meter dari api	104
Tabel 48. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api adalah mulai dari 6-meter dari api (tunanetra bergerak).....	105
Tabel 49. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada skenario jarak api ke tunanetra.....	106
Tabel 50. <i>Confusion Matrix</i> dari data api dan <i>non</i> api pada seluruh skenario ...	106
Tabel 51. Objek yang terdeteksi dan tidak terdeteksi di masing-masing <i>frame</i> .	111
Tabel 52. Hasil perhitungan <i>mean square error</i> (MSE).....	112

BAB I

PENDAHULUAN

1.1. Latar Belakang

Tunanetra atau yang biasa disebut buta, adalah istilah yang digunakan kepada seseorang yang memiliki ketidakmampuan dalam menggunakan indra penglihatan. Berdasarkan tingkat keterbatasannya, tunanetra terbagi menjadi dua, yaitu yang masih memiliki sedikit penglihatan dan yang sama sekali sudah tidak dapat melihat.

Menurut data dari *National Center for Biotechnology Information* (NCBI) pada tahun 2015 bahwa diperkirakan terdapat 253 juta jiwa masyarakat dunia yang mengalami gangguan penglihatan, dan 36 juta di antaranya mengalami kebutaan (Ackland, et al., 2017). Berdasarkan hasil dari Susenas pada tahun 2012 menyatakan bahwa terdapat 1.776.912 jiwa di Indonesia yang merupakan penyandang tunanetra, dan diperkirakan jumlah tersebut akan terus bertambah (Pratiwi & Rachmawati, 2016) .

Secara tradisional, para penyandang tunanetra menggunakan tongkat untuk mendeteksi lingkungan sekitar, akan tetapi penggunaan cara tersebut tidak dapat mendeteksi lingkungan dengan cukup baik, sehingga dapat membahayakan keselamatan kaum tunanetra (Sunanto, 2005). Salah satu objek sekitar yang dapat membahayakan kaum tunanetra adalah api. Api adalah proses oksidasi cepat terhadap suatu material dalam proses pembakaran yang menghasilkan panas, cahaya, dan hasil reaksi kimia lainnya (Sukabumi, 2017). Namun dewasa ini,

teknologi mulai diterapkan untuk membantu penyandang tunanetra dalam mendeteksi objek yang ada di sekitarnya (Anwar & Aljahdali, 2017).

Sebelumnya telah ditemukan sebuah alat untuk mendeteksi objek di lingkungan sekitar dengan memanfaatkan tongkat yang diberikan sentuhan teknologi, dan berhasil mendeteksi gangguan di sekitar penyandang tunanetra dan memberikan GPS untuk membantu pergerakan mereka, akan tetapi alat tersebut terbilang mahal (Anwar & Aljahdali, 2017). Pada penelitian yang lain, sistem untuk melakukan pendeteksian objek api juga berhasil ditemukan, akan tetapi masih sulit untuk mendeteksi api dengan akurat. Mengingat karakteristik api yang terus bergerak secara dinamis, algoritma *blob analysis* dapat diterapkan untuk dapat mendeteksi api di sekitar penyandang tunanetra. Akan tetapi hasil dari penelitian tersebut tidak dapat membedakan antara objek api dan cahaya pada lampu kendaraan yang bergerak, dimana lampu kendaraan tersebut juga melakukan pergerakan secara dinamis (Indrabayu, et al., 2020).

Berdasarkan hal tersebut, penulis mengajukan judul “**Modifikasi *Blob Analysis* untuk Mendeteksi Gerakan dan Perpindahan Titik Pusat Api.**” Sistem ini dirancang dengan memodifikasi algoritma *blob analysis* untuk dapat membedakan api, cahaya lampu kendaraan, dan objek yang lain, serta menggunakan metode *pinhole camera model* untuk mengukur perkiraan jarak antara api dan penyandang tunanetra.

Algoritma *Blob Analysis* adalah metode menganalisis gambar yang telah mengalami pemrosesan binerisasi, dimana pemrosesan binerisasi adalah menetapkan nilai dari setiap pixel pada sebuah gambar menjadi 0 dan 1 (Visco

Technologies, 2021). Setelah mendeteksi objek api menggunakan algoritma *blob analysis*, dilakukan pengukuran perkiraan jarak antara api dan penyandang tunanetra menggunakan metode *pinhole camera model*. Metode *Pinhole camera model* adalah sebuah metode yang mengambil gambar berdasarkan proyeksi geometri antara titik 3D dan 2D yang sesuai dengan proyeksi pada gambar (Indrabayu, et al., 2020). Metode ini kemudian selaras dengan sistem pendeteksi api yang diusulkan.

1.2. Rumusan Permasalahan

Berdasarkan latar belakang di atas, maka dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana sistem yang diusulkan dapat mendeteksi dan membedakan antara api dan cahaya lampu kendaraan yang sedang bergerak?
2. Bagaimana sistem yang diusulkan dapat mengukur jarak antara api dan penyandang tunanetra?

1.3. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Mendeteksi dan membedakan antara api dan cahaya lampu kendaraan yang sedang bergerak di sekitar tunanetra menggunakan algoritma *blob analysis* yang sudah dimodifikasi
2. Mengestimasi jarak antara api dan penyandang tunanetra menggunakan metode *pinhole camera model*

1.4. Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Membantu tunanetra dalam mendeteksi api di sekitarnya
2. Menyediakan pendeteksi api yang lebih akurat
3. Membantu tunanetra dalam mengestimasi jarak api di sekitarnya

1.5. Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut:

1. Pengambilan data dilakukan pada malam hari sekitar pukul 19.00 – 05.00
2. Tidak sedang terjadi hujan pada saat pengambilan data
3. Penelitian ini menggunakan kamera beresolusi 1080p dengan *frame rate* 60 *fps*

1.6. Sistematika Penulisan

Laporan tugas akhir ini dibagi menjadi lima bab yang tersusun secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang pengambilan judul “Modifikasi Algoritma *Blob Analysis* untuk Mendeteksi Gerakan dan Perpindahan Titik Pusat Api” sebagai skripsi, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metode penelitian, dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang penjelasan secara umum mengenai teori-teori yang berkaitan dengan sistem deteksi objek berbasis *image detection*, metode *image processing* untuk mendeteksi api dan mengukur jarak objek.

BAB III METODOLOGI PENELITIAN

Bab ini membahas mengenai bagaimana perancangan sistem deteksi dan estimasi jarak objek menggunakan *image processing* dengan metode *color model* dan *blob analysis* dimana dilakukan proses analisis dan modifikasi algoritma *blob analysis* untuk dapat mendeteksi gerakan dan perpindahan titik pusat api, dan menggunakan *pinhole camera model* untuk mengestimasi jarak api.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas mengenai eksekusi sistem setelah dilakukan pemrograman pada MATLAB dan juga hasil dan pembahasan pada sistem berupa persentase akurasi deteksi objek dan estimasi jarak. Bab ini juga akan membahas terkait hasil implementasi sistem terhadap *live data*.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1. Tunanetra

Kata “tunanetra” dalam Kamus Besar Bahasa Indonesia berasal dari kata “tuna” yang artinya rusak atau cacat dan kata “netra” yang artinya adalah mata atau penglihatan, jadi kata tunanetra adalah rusak penglihatan. Sedangkan orang yang buta adalah orang yang rusak penglihatannya secara total (Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2022). Tunanetra tergolong menjadi dua kelompok, yaitu tunanetra dengan buta total dan tunanetra yang awas atau memiliki keterbatasan penglihatan (Kiptiya, 2014).

Orang yang mengalami cacat netra telah diklasifikasikan menurut beberapa sudut pandang, di antaranya yaitu (Kiptiya, 2014):

a. Kapan seseorang menderita tunanetra. Dapat digolongkan sebagai berikut:

- (1) Penderita tunanetra sejak lahir, yakni mereka yang sama sekali tidak memiliki pengalaman melihat
- (2) Penderita tunanetra setelah lahir atau pada usia kecil, yaitu mereka yang sudah memiliki kesan serta penglihatan visual, tetapi belum kuat dan mudah terlupakan
- (3) Penderita tunanetra pada usia sekolah atau usia remaja, kesan-kesan pengalaman visual meninggalkan pengaruh yang mendalam terhadap perkembangan pribadi

- (4) Penderita tunanetra pada usia dewasa, merupakan mereka yang dengan segala kesadaran masih mampu melakukan latihan-latihan penyesuaian diri
 - (5) Penderita tunanetra pada usia lanjut, yaitu mereka yang sebagian besar sudah sulit mengalami latihan-latihan diri
- b. Berdasarkan kemampuan daya lihat. Dapat digolongkan sebagai berikut:
- (1) Penderita tunanetra ringan, yaitu mereka yang mempunyai kelainan atau kekurangan daya penglihatan
 - (2) Penderita tunanetra setengah berat, yaitu mereka yang mengalami sebagian besar kerusakan daya penglihatan
 - (3) Penderita tunanetra berat, yaitu mereka yang sama sekali tidak dapat melihat atau yang sering disebut buta.

2.2. Citra

Citra merupakan variasi gelap-terang, redup-cerah, dan/atau warna-warni di suatu bidang datar yang diperoleh menggunakan sistem visual yang memiliki kemiripan dengan suatu objek (Sulistiyanti, et al., 2016). Secara umum citra terbagi menjadi dua sifat yaitu:

a. Citra Analog

Citra analog adalah citra yang bersifat *continue*, seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil *CT scan*, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses di komputer secara langsung. Oleh

sebab itu, agar ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, *cam*, *CT scan*, sensor *rontgen* untuk foto *thorax*, sensor gelombang pendek pada sistem radar, sensor *ultrasound* pada sistem USG, dan lain-lain (Woods & Gonzales, 2008).

b. Citra Digital

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran *pixel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dalam artian kuantisasi pada citra menyatakan jumlah warna yang ada pada citra (Woods & Gonzales, 2008).

Citra digital itu sendiri terbagi menjadi beberapa jenis, di antaranya:

(1) Citra Biner

Citra biner (*binary image*) adalah citra digital yang hanya memiliki 2 kemungkinan warna, yaitu hitam dan putih. Citra biner disebut juga dengan citra W&B (*White & Black*) atau citra monokrom. Hanya dibutuhkan 1-bit untuk mewakili nilai setiap piksel dari citra biner. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. *Piksel* dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat

keabuan lebih kecil dari nilai batas akan diberi nilai 0. Citra biner sering sekali muncul sebagai hasil dari proses pengolahan, seperti segmentasi, morfologi ataupun *dithering*. Fungsi dari binerisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola akan lebih mudah terdeteksi pada citra yang mengandung lebih sedikit warna (Woods & Gonzales, 2008). Contoh dari citra biner dapat dilihat pada **gambar 1(a)**.

(2) Citra Grayscale

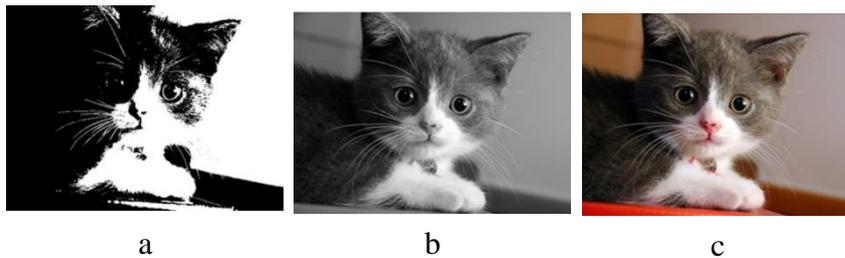
Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari $Red = Green = Blue$. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra “hitam-putih” dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu “hitam” dan “putih” saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna di antaranya sangat banyak. Citra *grayscale* seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik *single band*. Citra *grayscale* disimpan dalam format 8-bit untuk setiap sample piksel, yang memungkinkan sebanyak 256 intensitas. Untuk mengubah citra berwarna yang mempunyai nilai matriks masing-masing R, G, dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan

mengambil rata-rata dari nilai R, G, dan B (Woods & Gonzales, 2008).

Contoh dari citra *grayscale* dapat dilihat pada **gambar 1(b)**.

(3) Citra RGB

Red (Merah), *Green* (Hijau) dan *Blue* (Biru) merupakan warna dasar yang diterima oleh mata manusia. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB. Setiap titik pada citra warna membutuhkan data sebesar 3 *byte*. Setiap warna dasar memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255 (8 bit). RGB didasarkan pada teori bahwa mata manusia peka terhadap panjang gelombang 630nm (merah), 530 nm(hijau), dan 4nm (biru) (Woods & Gonzales, 2008). Contoh dari citra HSV dapat dilihat pada **gambar 1(c)**.



Gambar 1. (a) Citra Biner. (b) Citra *Grayscale*. (c) Citra RGB
Sumber : www.google.com

2.3. Pengolahan Citra

Pengolahan citra adalah pemrosesan citra menggunakan komputer untuk membuat kualitas dari citra tersebut menjadi lebih baik. Pengolahan ini dilakukan dengan cara memodifikasi setiap titik dalam citra tersebut sesuai dengan keperluan. Secara garis besar, modifikasi tersebut dikelompokkan menjadi:

- a. Operasi titik. Dimana setiap titik diolah secara independen terhadap titik-titik yang lain.
- b. Operasi global. Dimana karakteristik global (biasanya berupa sifat statistik) dari citra digunakan untuk memodifikasi nilai setiap titik.
- c. Operasi geometrik. Dimana bentuk, ukuran, atau orientasi citra dimodifikasi secara geometrik
- d. Operasi temporal/berbasis bingkai. Dimana citra diolah dengan cara dikombinasikan dengan citra lain.
- e. Operasi banyak titik bertetangga. Dimana data dari titik yang bersebelahan dengan titik yang ditinjau ikut berperan dalam mengubah nilai.
- f. Operasi morfologi. Yaitu operasi yang berdasarkan segmen atau bagian dalam citra yang menjadi perhatian (Achmad, 2003)

Operasi-operasi pengolahan citra diterapkan apabila:

- a. *Image Enhancement*. Perbaikan citra dilakukan untuk meningkatkan kualitas citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra. Sebagai contoh: memperbaiki kontras antara gelap atau terang, memperbaiki tepian objek, menajamkan citra, dll.
- b. *Image Restoration*. Adanya cacat pada citra sehingga perlu dihilangkan /diminimumkan. Sebagai contoh: menghilangkan kesamaran (*blur*) citra yang tampak kabur karena pengaturan fokus lensa yang tidak tepat.
- c. *Image Segmentation*. Elemen dalam citra perlu dikelompokkan, dicocokkan atau di ukur.

- d. *Image Analysis*. Ingin dilakukannya ekstraksi fitur tertentu yang dimiliki oleh citra untuk membantu dalam pengidentifikasian objek tersebut.
- e. *Image Reconstruction*. Untuk melakukan penggabungan citra dengan bagian citra yang lain.
- f. *Image Compression*. Perlunya memampatkan citra untuk mengurangi besar ukuran file.
- g. *Stenografi & Watermarking*. Untuk menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaannya data tersebut tidak diketahui oleh orang lain (Pengertian, 2013).

Adapun tujuan dan manfaat dari dilakukannya pengolahan citra digital adalah:

- a. Memperbaiki kualitas gambar, dilihat dari aspek radiometric dan aspek geometric. Aspek radiometric terdiri dari peningkatan kontras, restorasi citra, transformasi warna sedangkan aspek geometric terdiri dari rotasi, skala, translasi, transformasi geometric).
- b. Melakukan proses penarikan informasi. Citra dapat mengandung deskripsi atau informasi yang dapat ditarik.
- c. Melakukan pemilihan citra ciri (feature images) yang optimal. Tujuannya itu untuk proses analisis.
- d. Melakukan kompresi atau reduksi data. Tujuannya yaitu untuk penyimpanan data, transmisi data, dan waktu proses data (Pengertian, 2013).

Secara garis besar, dalam melakukan pengolahan citra digital perlu dilakukan tiga hal berikut:

a. *Pre-processing*

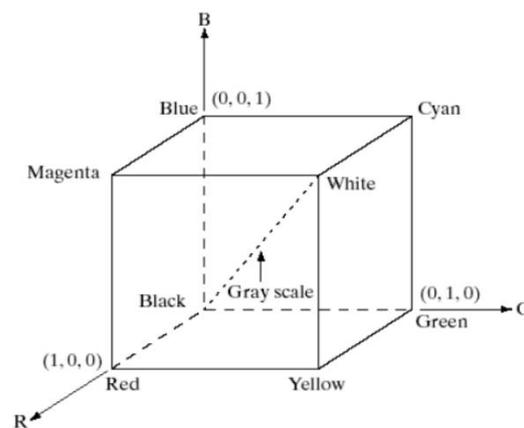
Citra yang akan diproses, terlebih dahulu dilakukan pra pemrosesan (*pre-processing*). *Pre-processing* adalah proses operasi citra pada tingkat abstraksi yang terendah dan merupakan tahapan awal yang dilakukan dalam pengolahan citra gambar. Tujuan dari *pre-processing* ini adalah untuk meningkatkan kualitas citra sehingga dapat dilakukan pemrosesan lebih lanjut. *Pre-processing* dapat menghilangkan *noise*, memperjelas fitur yang akan diekstraksi pada proses selanjutnya. Contoh proses *pre-processing* adalah *rotate*, *resize*, *cropping*, dan lain sebagainya. Sedangkan contoh degradasi pada gambar yaitu cacat atau derau (*noise*), warna terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Hal ini menyebabkan gambar menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami degradasi menjadi mudah diinterpretasi (baik oleh manusia maupun mesin), maka gambar perlu dimanipulasi menjadi gambar yang kualitasnya lebih baik. Selain perbaikan kualitas, *pre-processing* juga digunakan untuk mengubah tipe data dan ukuran data gambar. Hal ini disesuaikan dengan kebutuhan jenis data yang akan diproses pada tahap selanjutnya (Indrabulan, 2016)

b. Segmentasi

Segmentasi adalah proses yang bertujuan untuk memisahkan antara objek atau *foreground* dengan *background*. Pada umumnya keluaran hasil segmentasi citra adalah berupa citra biner di mana objek (*foreground*) yang dikehendaki berwarna putih (1), sedangkan *background* nya yang ingin dihilangkan

berwarna hitam (0). Segmentasi juga merupakan proses membagi citra menjadi beberapa bagian. Pembagian ini biasanya berdasarkan karakteristik dari piksel pada citra. Beberapa metode segmentasi citra yang biasa digunakan adalah *thresholding*, *active contour*, segmentasi warna, deteksi tepi, *watershed* dan transformasi Hough. Pada penelitian ini, metode segmentasi yang digunakan adalah segmentasi warna karena fitur api yang paling mencolok adalah warnanya (Indrabulan, 2016).

(1) RGB Color Model



Gambar 2. Ruang Warna RGB

Sumber : www.wordpress.com

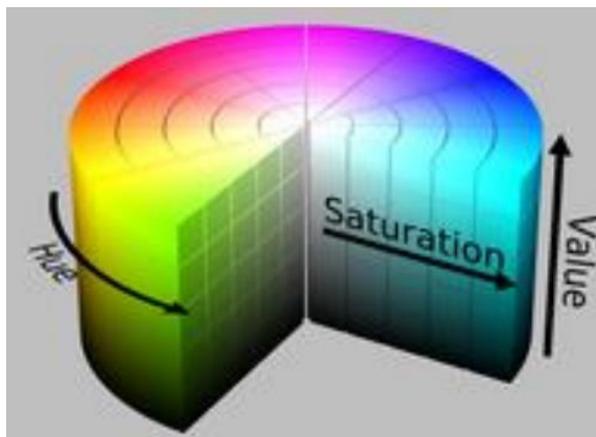
Model warna RGB memiliki ruang warna 3 dimensi, yaitu dimensi R, G dan B. Untuk lebih jelasnya bisa dilihat pada **gambar 2**. Pada **gambar 2** dilihat bahwa panjang maksimum setiap sumbu adalah 1. Angka ini telah dinormalisasi dari angka 255. Setiap titik yang berada pada ruang warna RGB merupakan warna dengan memiliki komponen R, G dan B. Dapat dilihat pada **gambar 2**, untuk titik (0,0,0) merupakan titik warna yang berwarna hitam, sedangkan titik (1,1,1) merupakan titik warna yang berwarna putih. (T.S., 2019).

Model warna RGB merupakan model warna *hardware oriented*, dimana warna yang dihasilkan akan tergantung dengan *hardware* yang digunakan (T.S., 2019).

(2) HSV Color Model

Hue, *Saturation* dan *Value* merupakan warna yang memiliki model ruang warna yang berbeda dengan RGB. *Hue* digunakan untuk menyatakan warna sesungguhnya seperti merah, jingga, kuning dll. *Hue* digunakan untuk membedakan tingkat warna kemerahan, kehijauan, kekuningan, kebiruan, dll berdasarkan gelombang cahaya. Nilai dari *Hue* merupakan nilai dari 0 hingga 2π . Nilai ini menyatakan bahwa model *hue* ini berbentuk lingkaran. Sehingga apabila kita telusuri terus dari warna merah maka kita akan kembali lagi ke warna merah. *Saturation* digunakan untuk menyatakan tingkat keputihan dari setiap warna pada *hue*. Sehingga apabila bila warna merah memiliki tingkat *saturation* yang tinggi atau bernilai 1 maka tidak ada kandungan warna putih pada warna *hue*. Sedangkan apabila *saturation* memiliki nilai 0 maka kandungan warna tersebut tidak memiliki *hue* atau sepenuhnya berwarna putih. *Value* pada model HSV merupakan nilai kadar warna yang berada pada warna tersebut. Mirip seperti *saturation*, *value* ini memiliki rentang dari 0 hingga 1. Apabila nilai *value* ini sebesar 0, maka warna tersebut akan menjadi hitam. Sedangkan apabila nilai *value* bernilai 1 maka tingkat kandungan warna hitam pada warna tersebut menjadi hilang (T.S., 2019).

Untuk lebih jelasnya bisa dilihat pada **gambar 3**. Pada **gambar 3** dapat dilihat bahwa puncak nilai *value* merupakan seluruhnya warna yang tidak mengandung warna hitam. Sehingga yang mempengaruhi warna adalah *hue* dan *saturation*. Sangat berbeda dengan HSL yang apabila L bernilai 1 apapun nilai *hue* dan nilai *saturation* tidak dapat mengubah warna. Model warna HSV juga disebut sebagai model warna *user oriented*, karena warna yang dihasilkan merupakan *true color* dari citra (T.S., 2019).



Gambar 3. Visualisasi HSV Color Model
Sumber : <https://commons.wikimedia.org/>

c. Operasi Morfologi

Istilah morfologi berasal dari pembelajaran tentang bentuk atau struktur dari tumbuhan atau hewan. Proses morfologi menunjukkan operasi tertentu yang dikenakan pada sebuah objek dengan elemen terstruktur (Jain, 1986).

Beberapa definisi yang digunakan dalam operasi morfologi:

- Titik Objek, yaitu titik yang merupakan bagian dari objek. Biasanya direpresentasikan dengan nilai 1 atau warna putih.
- Titik Latar, yaitu titik yang merupakan bagian latar. Biasanya direpresentasikan dengan nilai 0 atau warna hitam.

- Titik Tetangga, adalah titik yang langsung bersebelahan dengan titik yang sedang ditinjau, yaitu titik yang berada di sebelah kiri, kanan, atas, dan bawah (Achmad, 2003).

Operasi morfologi adalah teknik pengolahan citra yang didasarkan pada bentuk segmen atau region dalam citra. Operasi ini antara lain meliputi pencarian batas/kontur, dilasi, erosi, penutupan (*closing*), pembukaan (*opening*), dan pengisian (*filling*) (Achmad, 2003).

(1) Pencarian Batas/Kontur.

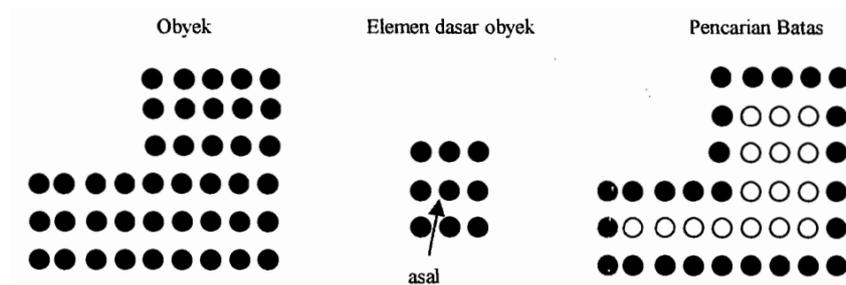
Operasi ini digunakan untuk menentukan batas/kontur dari segmen objek. Operasi ini dilakukan terhadap titik-titik objek, yaitu apabila titik objek tersebut bukan titik batas, maka titik tersebut dapat diubah menjadi titik latar, sedangkan titik-titik batas akan dipertahankan. Untuk mengetahui apakah suatu titik objek adalah titik batas atau tidak, dapat dilakukan dengan cara memeriksa semua titik tetangganya. Apabila ada salah satu dari semua tetangga tersebut merupakan titik latar, maka titik tersebut adalah titik biasa. Sedangkan apabila semua tetangganya adalah titik objek, maka titik tersebut bukanlah titik batas (Achmad, 2003).

Secara umum, algoritma untuk mencari batas/kontur objek adalah (Achmad, 2003) :

- Tetapkan bahwa citra hasil sama dengan citra asal
- Semua titik dalam citra asal, lakukan pencocokan apakah titik tersebut adalah titik dari objek

- Jika iya, maka lakukan pencocokan apakah titik tersebut adalah titik batas
 - Jika iya, maka titik tersebut adalah tetap titik objek.
 - Jika tidak, maka ubah titik tersebut menjadi titik latar
- Jika tidak, maka lanjutkan ke titik selanjutnya

Adapun visualisasi dari proses pencarian batas/kontur dapat dilihat pada **gambar 4**.



Gambar 4. Visualisasi proses pencarian batas/kontur
Sumber: (Achmad, 2003).

Contoh pada **gambar 4** di atas menunjukkan proses pencarian batas. Proses ini dilakukan dengan mengecek titik tetangga dan titik asal. Apabila ada salah satu dari titik tetangganya merupakan titik asal, maka titik asal merupakan titik batas yang harus dipertahankan (Achmad, 2003).

(2) Dilasi

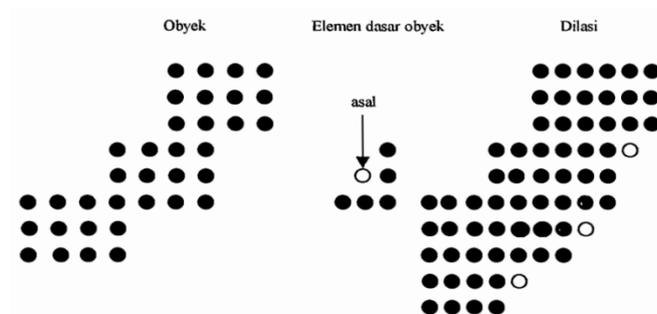
Operasi dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambah lapisan di sekeliling objek. Terdapat dua cara untuk melakukan operasi ini, yaitu dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek, atau lebih mudahnya tetapkan setiap titik yang tetangganya adalah titik objek menjadi titik objek.

Cara kedua, yaitu dengan mengubah semua titik di sekeliling titik batas menjadi titik objek, atau lebih mudahnya tetapkan semua titik tetangga sebuah titik objek menjadi titik objek (Achmad, 2003).

Secara umum, algoritma untuk mencari dilasi adalah (Achmad, 2003) :

- Untuk semua titik dalam citra, cek apakah titik tersebut adalah titik objek
 - Jika iya, maka ubah semua tetangganya menjadi titik objek
 - Jika tidak, maka lanjutkan ke titik selanjutnya

Adapun visualisasi dari operasi dilasi dapat dilihat pada **gambar 5**.



Gambar 5. Visualisasi operasi dilasi
Sumber: (Achmad, 2003).

Contoh pada **gambar 5** di atas menunjukkan proses operasi dilasi. Proses ini dilakukan dengan cara mengubah titik tetangga titik asal menjadi titik obyek (Achmad, 2003).

(3) Erosi

Operasi ini adalah kebalikan dari operasi dilasi. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Cara yang dapat dilakukan juga dua. Cara pertama, yaitu dengan mengubah semua

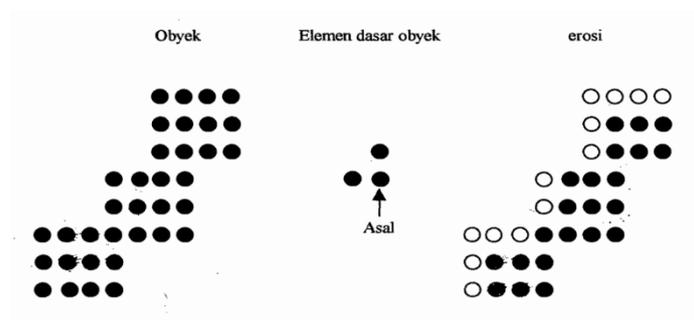
titik batas menjadi titik latar dan cara kedua, dengan mengeset semua titik di sekeliling titik latar menjadi titik latar (Achmad, 2003).

Secara umum, algoritma untuk operasi erosi adalah (Achmad, 2003)

:

- Untuk semua titik dalam citra, cek apakah titik tersebut adalah titik latar
 - Jika iya, maka ubah semua tetangganya menjadi titik latar
 - Jika tidak, maka lanjutkan ke titik berikutnya

Adapun visualisasi dari operasi dilasi dapat dilihat pada **gambar 6**.



Gambar 6. Visualisasi operasi erosi

Sumber: (Achmad, 2003).

Contoh pada **gambar 6** di atas menggambarkan operasi erosi. Operasi ini dilakukan dengan mengubah titik tetangga dari titik asal menjadi titik latar (Achmad, 2003).

(4) Penutupan (*closing*)

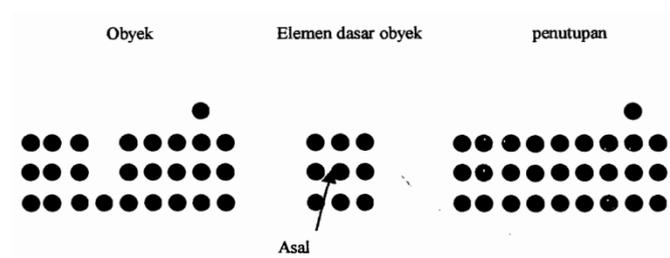
Operasi penutupan adalah kombinasi antara operasi dilasi dan erosi yang dilakukan secara beruntun. Citra asli didilasi terlebih dahulu, kemudian hasilnya dierosi. Operasi ini digunakan untuk menutup atau menghilangkan lubang-lubang kecil yang ada dalam segmen objek.

Operasi penutupan juga digunakan untuk menggabungkan dua segmen objek yang saling berdekatan (menutup sela antara dua objek yang sangat berdekatan) (Achmad, 2003).

Secara umum, algoritma untuk operasi penutupan adalah (Achmad, 2003) :

- Untuk semua titik dalam citra, cek apakah titik tersebut adalah titik objek
 - Jika iya, maka ubah semua tetangganya menjadi titik objek
 - Jika tidak, maka lanjutkan ke titik berikutnya
- Cek apakah titik tersebut adalah titik latar
 - Jika iya, maka ubah semua tetangganya menjadi titik latar
 - Jika tidak, maka lanjutkan ke titik selanjutnya

Adapun visualisasi dari operasi penutupan dapat dilihat pada **gambar 7**.



Gambar 7. Visualisasi operasi *closing*
Sumber: (Achmad, 2003).

Contoh pada **gambar 7** di atas menunjukkan operasi penutupan. Operasi ini dilakukan untuk menghilangkan celah titik yang terdapat pada titik objek (Achmad, 2003).

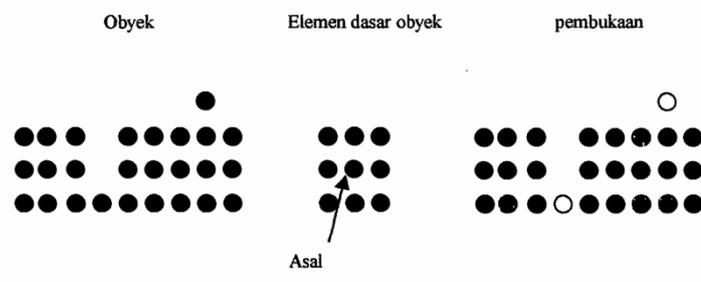
(5) Pembukaan (*opening*)

Operasi pembukaan juga merupakan kombinasi antara operasi erosi dan dilasi yang dilakukan secara berurutan, tetapi citra asli dierosi terlebih dahulu baru kemudian hasilnya didilasi. Operasi ini digunakan untuk memutus bagian-bagian dari objek yang hanya terhubung dengan satu atau dua buah titik saja (Achmad, 2003).

Secara umum, algoritma untuk operasi pembukaan adalah (Achmad, 2003) :

- Untuk semua titik dalam citra, cek apakah titik tersebut adalah titik latar
 - Jika iya, maka ubah semua tetangganya menjadi titik latar
 - Jika tidak, maka lanjutkan ke titik berikutnya
- Cek apakah titik tersebut adalah titik objek
 - Jika iya, maka ubah semua tetangganya menjadi titik objek
 - Jika tidak, maka lanjutkan ke titik berikutnya

Adapun visualisasi dari operasi pembukaan dapat dilihat pada **gambar 8**.



Gambar 8. Visualisasi operasi *opening*
Sumber: (Achmad, 2003).

Contoh pada **gambar 8** di atas menunjukkan operasi pembukaan. Operasi ini dilakukan dengan memutus bagian objek yang hanya terhubung dengan satu atau dua titik saja (Achmad, 2003).

(6) Pengisian (*filling*)

Operasi pengisian merupakan kebalikan dari operasi pencarian batas citra. Pada operasi ini, citra masukan adalah citra batas/kontur. Kemudian dilakukan pengisian sehingga diperoleh segmen objek yang pejal/solid. Prosesnya dimulai dengan menentukan titik awal pengisian yang terletak di dalam objek, kemudian bergerak ke arah titik-titik tetangganya. Operasi dilakukan secara rekursif dan berhenti jika sampai di batas objek. Perbedaan lokasi titik awal ini dapat mengakibatkan citra hasil yang diperoleh berbeda satu sama lain (Achmad, 2003).

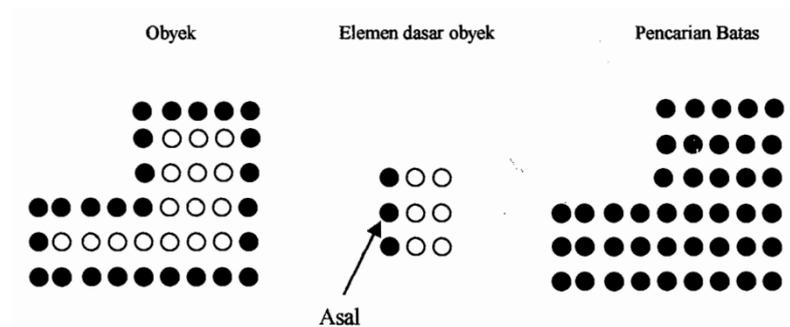
Secara umum, algoritma untuk operasi pengisian adalah (Achmad, 2003) :

- Tentukan titik tersebut menjadi titik objek
- Cek apakah titik tetangga atasnya adalah titik latar
 - Jika iya, maka tetapkan titik tersebut menjadi titik objek.
 - Jika tidak, maka lanjutkan ke titik berikutnya
- Cek apakah titik tetangga kanannya adalah titik latar
 - Jika iya, maka tetapkan titik tersebut menjadi titik objek.
 - Jika tidak, maka lanjutkan ke titik berikutnya
- Cek apakah titik tetangga bawahnya adalah titik latar
 - Jika iya, maka tetapkan titik tersebut menjadi titik objek.

- Jika tidak, maka lanjutkan ke titik berikutnya
- Cek apakah titik tetangga kirinya adalah titik latar
 - Jika iya, maka tetapkan titik tersebut menjadi titik objek.
 - Jika tidak, maka lanjutkan ke titik berikutnya

Adapun visualisasi dari operasi pengisian dapat dilihat pada **gambar**

9.



Gambar 9. Visualisasi operasi *filling*

Sumber: (Achmad, 2003).

Contoh pada **gambar 9** di atas menunjukkan operasi pengisian. Operasi ini dilakukan dengan mengubah titik tetangga dari titik asal menjadi titik objek dengan cara mengisi objek dengan arah ke atas, sampai menabrak batas objek, kemudian ke arah kanan sampai menabrak batas, lalu ke bawah dan akhirnya ke kiri. Urutan arah tersebut dapat diatur menurut kehendak dari pengguna (Achmad, 2003).

2.4. Pengolahan Video

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. Video juga bisa dikatakan sebagai gabungan gambar-gambar bergerak yang dibaca berurutan dalam suatu waktu

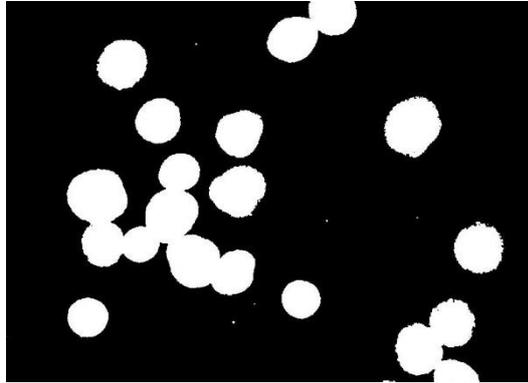
dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame rate*, dengan satu fps (*frame per second*). Karena dimainkan dalam kecepatan yang tinggi maka tercipta ilusi gerak yang halus, semakin besar nilai frame rate maka akan semakin halus pergerakan yang ditampilkan (Liana, 2012).

2.5. *Blob Analysis*

Blob atau *Binary Large Object* adalah tipe data yang dapat menyimpan objek atau data berbentuk biner. *Blob* biasanya digunakan dalam database untuk menyimpan data biner seperti gambar, file multimedia, dan kode perangkat lunak yang dapat dieksekusi (Techopedia, 2016). *Blob Analysis* adalah teknik dasar dari *machine vision* berdasarkan daerah citra yang konstan. Kelebihan utama dari teknik ini adalah memiliki fleksibilitas yang tinggi dan kemampuan yang sangat baik. Namun batas dari teknik ini adalah perbedaan *background* dan *foreground* yang harus jelas (Zebra Technologies Corp., 2022).

Blob Analysis terdiri dari *blob extraction* untuk mengisolasi *blobs* (objek) dalam citra biner, dimana *blob* terdiri dari kelompok piksel yang saling berhubungan seperti contoh **gambar 10**. Piksel yang berhubungan disebut *connectivity*, yang pikselnya dapat bertetangga atau tidak. *Blob analysis* termasuk dalam analisis terhadap luas area dan bentuk objek *blob* dari suatu citra yang menjadi fokus deteksi. Dalam menentukan nilai *blob*, ada beberapa hal yang harus diketahui untuk menghasilkan sebuah *blob* yang optimal. Penentuan luas *blob* pada setiap objek pada proses segmentasi *foreground* perlu dianalisis karena nilai *blob* pada tiap objek akan berbeda. Hal ini dipengaruhi oleh fitur objek seperti ukuran,

jenis, dan teknik pengambilan data video. Prosesnya dimulai dari penandaan area *foreground* yang dianggap objek, kemudian pengumpulan data area menjadi *blob*



Gambar 10. *Binary Large Object*

Sumber: <https://gis.stackexchange.com/>

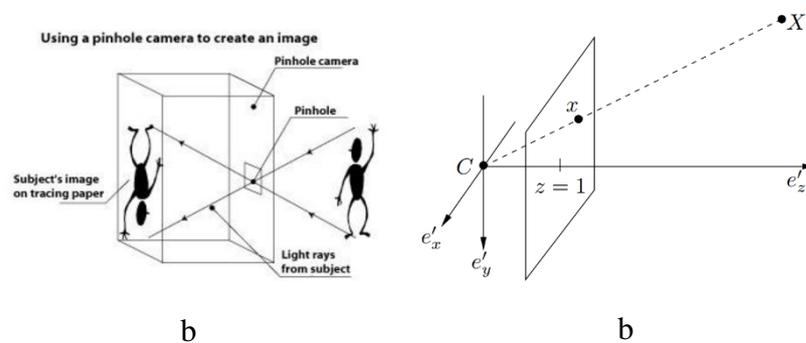
seperti posisi piksel awal, panjang terhadap sumbu x dan sumbu y dan luas area piksel terhadap area sebuah *blob* (Basri, 2015).

Deteksi api dengan metode *blob analysis* dilakukan dengan menganalisis citra biner dari objek yang telah disegmentasi. Analisa yang dilakukan berdasarkan fitur api yang mengalami perubahan besar area secara terus-menerus, dimana objek yang telah disegmentasi dilacak titik pusatnya kemudian dianalisis perubahan besar area dari masing-masing *blob* yang terdeteksi di tiap *frame* hingga 20 *frame* berikutnya. Setelah itu masing-masing dari *blob* yang terdeteksi akan ditotalkan dan dibandingkan dengan *threshold* perubahan area api, serta menghitung besar perpindahan titik pusat api antara *frame* pertama dan *frame* ke-20 dan dibandingkan dengan *threshold* perpindahan titik pusat api.

2.6. *Pinhole Model*

Pinhole Model atau *Pinhole Camera Model* adalah sebuah metode pengambilan gambar berdasarkan hubungan geometrik antara titik three dimensi dan titik 2 dimensi yang sesuai dengan proyeksi pada citra. Kamera *Pinhole* atau

biasa disebut kamera lubang jarum, merupakan sebuah kamera unik yang dapat memproyeksikan gambar dengan baik walaupun tidak menggunakan lensa. Prinsip kerja kamera *Pinhole* adalah seperti pada saat seseorang berdiri pada sebuah ruang kamar yang benar-benar tertutup rapat, namun di dalam ruangan tersebut terdapat sebuah lubang kecil di salah satu sisinya. Gelombang cahaya akan ‘bocor’ memasuki lubang ini, sehingga sebuah citra akan terbentuk pada sisi dinding yang berseberangan dengan lubang tersebut. Citra yang terbentuk akan menyerupai objek yang terletak di luar ruang kamar, hanya saja terproyeksikan secara terbalik (Apexia, 2018).



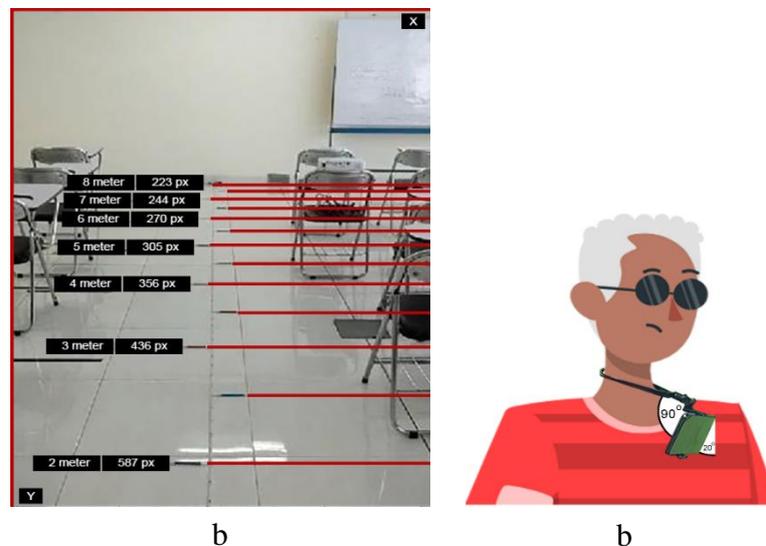
Gambar 11. (a) Pengambilan gambar dengan *Pinhole model*.
 (b) Model matematika dari pengambilan gambar *Pinhole model*

Sumber : (Apexia, 2018).

Ketika menggunakan *Pinhole model* pemetaan geometrik dari 3 dimensi ke 2 dimensi disebut dengan proyeksi perspektif. Titik tengah dari proyeksi perspektif dinyatakan sebagai *optical center* atau *camera center* dan garis yang tegak lurus dengan bidang citra yang melewati *optical center* dinyatakan sebagai *optical axis*, selanjutnya titik perpotongan dari bidang citra dengan *optical axis* dinyatakan sebagai *principal point* (Apexia, 2018). **Gambar 11(a)** menunjukkan contoh pengambilan gambar dengan metode *Pinhole model*, sedangkan **gambar 11(b)**

menunjukkan model matematika dari *Pinhole model* sehingga model matematika ini menjadi dasar untuk dapat mengetahui jarak dari sebuah objek (Lund University, 2020).

Estimasi jarak pada penelitian dilakukan dengan metode yang berdasarkan *Pinhole model*, metode ini bekerja dengan cara memberikan piksel ukuran jarak asli seperti pada **gambar 12(a)** dengan besar sudut pengambilan gambar seperti pada **gambar 12(b)**.



Gambar 12. (a) Ukuran jarak dalam piksel.
(b) Besar sudut pengambilan video

Sehingga pada jarak piksel 587 diinisialisasi dengan jarak 2 meter, piksel 498 diinisialisasi sebagai 2.5 meter, piksel 436 diinisialisasikan sebagai 3 meter, piksel 391 diinisialisasi sebagai 3.5 meter, piksel 356 diinisialisasikan sebagai 4 meter, piksel 328 diinisialisasi sebagai 4.5 meter, piksel 305 diinisialisasikan sebagai 5 meter, piksel 286 diinisialisasi sebagai 5.5 meter, piksel 270 diinisialisasikan sebagai 6 meter, piksel 256 diinisialisasi sebagai 6.5 meter, piksel

244 diinisialisasikan sebagai 7 meter, piksel 233 diinisialisasi sebagai 7.5 meter, dan piksel 223 diinisialisasikan sebagai 8 meter.

2.7. Penelitian Terkait

Penelitian mengenai deteksi api dan estimasi jarak objek telah banyak dilakukan sebelumnya, di antaranya:

a. *Blob Adaptation Through Frames Analysis for Dynamic Fire Detection*

Penelitian deteksi api dan jarak api yang dilakukan oleh Indrabayu dkk. menggunakan fitur warna dari api, dengan menerapkan segmentasi warna *HSV Color Model* dan *RGB Color Model*, kemudian hasil dari segmentasi warna tersebut dimasukkan ke dalam proses morfologi. Hasil dari proses morfologi tersebut lalu dilakukan analisis terhadap *blob* yang terbentuk, dimana karena sifat api yang luas areanya selalu berubah-ubah membuat metode *blob analysis* cocok untuk digunakan dengan akurasi mencapai 88.86%. Setelah *blob* tersebut terdeteksi sebagai api, lalu dilakukan pengukuran jarak api menggunakan *pinhole camera model* dan berhasil mencapai nilai MSE sebesar 0.0358. Deteksi dan pengukuran jarak api tersebut dilakukan terhadap beberapa objek api dan *non* api (Indrabayu, et al., 2020).

b. Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis *Sms Gateway* Pada Raspberry Pi

Selanjutnya penelitian yang dilakukan oleh Syahrul Yoga Pradana dkk. menggunakan RGB dan *HSV color model* untuk proses segmentasi gambarnya, dan mengubahnya menjadi *binary image* untuk dapat

mendeteksi titik api yang terpusat. Kemudian dengan memanfaatkan *sms gateway* pada Raspberry Pi, hasil dari pendeteksian api dapat diketahui. Penggunaan metode tersebut berhasil mencapai tingkat akurasi pendeteksian titik api sebesar 90% (Pradana & dkk, 2018).

c. *Image Processing Based Forest Fire Detection using Infrared Camera*

Penelitian yang dilakukan oleh Norsuzila Ya'cob dkk menggunakan segmentasi warna RGB dan YCbCr *color model* untuk mendeteksi keberadaan api pada kebakaran hutan kemudian melakukan filter terhadap citra yang ditangkap menggunakan MATLAB *Analyzer ti Process Images* untuk dapat mendapatkan titik pusat dari api pada kebakaran hutan. Pendeteksian ini dilakukan dengan memanfaatkan kamera inframerah. Penelitian tersebut berhasil membedakan api dan bukan api serta mencari titik pusat dari api pada kebakaran hutan (Ya'acob & dkk, 2020).

d. *A Real-Time Fire Detection Method from Video with Multifeatured Fusion*

Penelitian yang dilakukan oleh Faming Gong dkk untuk mendeteksi api secara *real time* menerapkan RGB dan HIS *color model* dalam segmentasi warnanya. Untuk dapat mendeteksi gerakan dari api, maka dilakukan metode *Improved Frame Difference*, yang kemudian dilakukan verifikasi hasil pendeteksian api menggunakan *Support Vector Machine* atau SVM. Dengan menerapkan metode tersebut, api berhasil dideteksi dengan tingkat keakuratan mencapai 95.29% (Gong & dkk, 2019).