

**USULAN TUGAS AKHIR**  
**DETEKSI API PADA VIDEO DENGAN GAUSSIAN MIXTURE**  
**MODEL UNTUK DETEKSI GERAKAN DAN SEGMENTASI**  
**WARNA API DALAM RUANG WARNA YCBCR**

Tugas akhir  
untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



Oleh :

**RISTIRIANTO ADI**

**F1D016078**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2020**

**USULAN TUGAS AKHIR**  
**DETEKSI API PADA VIDEO DENGAN GAUSSIAN MIXTURE MODEL**  
**UNTUK DETEKSI GERAKAN DAN SEGMENTASI WARNA API DALAM**  
**RUANG WARNA YCBCR**

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama

**Prof. Dr. Eng. I GP Suta Wijaya, S.T., M.T.**  
**NIP. 19731130 200003 1 001**

Tanggal: 20 April 2020

2. Pembimbing Pendamping

**Ramaditia Dwiyanaputra, ST., M.Eng.**  
**NIP. -**

Tanggal: 21 April 2020

Mengetahui

Ketua Program Studi Teknik Informatika

Fakultas Teknik

Universitas Mataram

**Prof. Dr. Eng. I GP Suta Wijaya, S.T., M.T.**

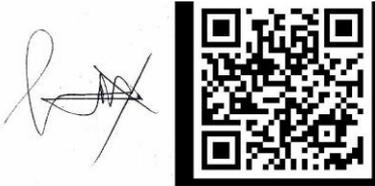
NIP. 19731130 200003 1 001

**USULAN TUGAS AKHIR**  
**DETEKSI API PADA VIDEO DENGAN GAUSSIAN MIXTURE MODEL**  
**UNTUK DETEKSI GERAKAN DAN SEGMENTASI WARNA API DALAM**  
**RUANG WARNA YCBCR**

Telah diperiksa dan disetujui oleh Tim Penguji

:

1. Penguji I



**Dr. Eng. I Gede Putu Wirarama Wedashwara W, S.T., M.T.** Tanggal: 24 Juni 2020

NIP: 19840919 201803 1 001

2. Penguji II



**Fitri Bimantoro, ST., M.Kom.**

Tanggal: 24 Juni 2020

NIP: 19860622 201504 1 002

3. Penguji III



**Gibran Satya Nugraha, S.Kom., M.Eng.**

Tanggal: 24 Juni 2020

NIP: 19920323 201903 1 012

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Teknik

Universitas Mataram



**Prof. Dr. Eng. I GP Suta Wijaya, S.T., M.T.**

NIP. 19731130 200003 1 001

# DAFTAR ISI

COVER .....	i
LEMBAR PENGESAHAN.....	ii
<b>DAFTAR ISI</b> .....	<b>iv</b>
<b>DAFTAR TABEL</b> .....	<b>vi</b>
<b>DAFTAR GAMBAR</b> .....	<b>vii</b>
<b>ABSTRAK</b> .....	<b>viii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI</b> .....	<b>5</b>
2.1 Tinjauan Pustaka.....	5
2.2 Teori Penunjang.....	7
2.2.1 Api .....	7
2.2.2 Kecerdasan Buatan .....	8
2.2.3 Citra dan Video.....	8
2.2.4 CCTV .....	9
2.2.5 Computer Vision .....	10
2.2.6 Warna .....	11
2.2.7 Segmentasi Warna .....	11
2.2.8 Ruang Warna YCbCr .....	11
2.2.9 Background Substraction dengan Gaussian Mixture Model .....	12

2.2.10	Klasifikasi Piksel Api berdasarkan Model Warna .....	14
<b>BAB III</b>	<b>METODE PENELITIAN .....</b>	<b>17</b>
3.1	Bahan dan Alat Penelitian.....	17
3.2	Rencana Penelitian.....	17
3.2.1	Pengumpulan Data.....	18
3.2.2	Studi Literatur.....	21
3.2.3	Perancangan Model Deteksi Api pada Video.....	21
3.2.4	Skenario Pengujian.....	28
3.3	Jadwal Kegiatan.....	30
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>31</b>
<b>LAMPIRAN</b>	<b>.....</b>	<b>34</b>

## **DAFTAR TABEL**

Tabel 3.1 Pembagian data video untuk latar siang dan malam.....	26
Tabel 3.2 Confusion matrix 2x2 .....	29
Tabel 3.3 Jadwal kegiatan perancangan sistem .....	30

## DAFTAR GAMBAR

Gambar 2.1 Contoh penerapan dari algoritma .....	13
Gambar 2.2 Perbandingan <i>channel</i> R, G, dan B .....	14
Gambar 2.3 Perbandingan <i>channel</i> Y, Cb, dan Cr.....	15
Gambar 3.1 Diagram Alir Rencana Penelitian .....	18
Gambar 3.2 Contoh data api pada siang dan malam hari .....	19
Gambar 3.3 Api kecil.....	19
Gambar 3.4 Api Sedang.....	20
Gambar 3.5 Api besar .....	20
Gambar 3.6 Objek yang menyerupai api .....	21
Gambar 3.7 Proses deteksi gerakan dan segmentasi api.....	21

## ABSTRAK

Api merupakan awal mula dari kebakaran, yang merupakan bencana alam yang dapat membahayakan jiwa dan menyebabkan kerugian harta-benda. Solusi untuk mendeteksi api yang umum digunakan saat ini adalah dengan menggunakan sensor, yang bekerja mendeteksi api dengan bergantung kepada karakteristik tertentu seperti asap, suhu, atau radiasi. Sensor api tidak cocok digunakan di ruang terbuka atau ruangan yang ukurannya sangat luas. Selain itu, apabila dipasang di lingkungan *outdoor*, sensor juga rentan terhadap kondisi cuaca pada saat itu. Apabila arah angin tidak menuju sensor sehingga asap diterbangkan berlawanan arah dengan posisi sensor, kemungkinan besar sensor tidak bisa mendeteksi terjadinya api dan asap. Untuk mengurangi dampak dari kekurangan-kekurangan tersebut, sensor api dapat digunakan bersama-sama dengan kamera pengawas (CCTV) yang sekarang banyak dipasang di gedung-gedung perkantoran. Metode pendeteksian secara visual ini memiliki beberapa kelebihan: dapat menjangkau wilayah yang lebih luas, dan dapat langsung menunjukkan lokasi persis di mana api berada lewat gambar yang ditangkap oleh kamera. Penelitian ini mencoba membangun model untuk mendeteksi api pada video dengan pendekatan pengolahan citra digital menggunakan Gaussian Mixture Model untuk deteksi gerakan dan segmentasi warna api pada ruang warna YCbCr. Model kemudian diuji dengan metrik akurasi, presisi, dan *recall*. Selain itu, diuji pula kecepatan komputasi yang mampu dicapai oleh model. *Dataset* yang digunakan berupa video dengan ukuran api kecil, sedang, besar, dan video yang hanya memiliki objek-objek serupa api. Hasil penelitian diharapkan dapat mendeteksi api dalam skala kecil, sedang, dan besar, dan mampu menghindari pendeteksian yang salah terhadap objek-objek yang serupa dengan api.

**Keywords:** Deteksi Api, GMM, YCbCr, Background Subtraction

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Api merupakan awal mula dari kebakaran, yang merupakan bencana alam yang dapat membahayakan jiwa dan menyebabkan kerugian harta-benda. Di Indonesia sendiri, kebakaran yang sering menjadi sorotan adalah kebakaran hutan, yang dianggap sebagai ancaman potensial bagi pembangunan berkelanjutan karena efeknya secara langsung bagi ekosistem, kontribusinya terhadap peningkatan emisi karbon, dan dampaknya bagi keanekaragaman hayati[1]. Kebakaran juga kerap terjadi di area padat penduduk, yang dipengaruhi oleh korslet listrik dan kebocoran gas di dapur[2].

Solusi untuk mendeteksi api yang umum digunakan saat ini adalah dengan menggunakan sensor, yang bekerja mendeteksi api dengan bergantung kepada karakteristik tertentu seperti asap, suhu, atau radiasi[3]. Akan tetapi, cara ini memiliki kekurangan, yaitu keterbatasan jarak yang dapat dijangkau oleh sensor. Sensor api tidak cocok digunakan di ruang terbuka atau ruangan yang ukurannya sangat luas. Selain itu, apabila dipasang di lingkungan *outdoor*, sensor juga rentan terhadap kondisi cuaca pada saat itu. Apabila arah angin tidak menuju sensor sehingga asap diterbangkan berlawanan arah dengan posisi sensor, kemungkinan besar sensor tidak bisa mendeteksi terjadinya api dan asap[4]. Untuk mengurangi dampak dari kekurangan-kekurangan tersebut, sensor api dapat digunakan bersama-sama dengan kamera pengawas (CCTV) yang sekarang banyak dipasang di gedung-gedung perkantoran. Metode pendeteksian secara visual ini memiliki beberapa kelebihan: dapat menjangkau wilayah yang lebih luas, dan dapat langsung menunjukkan lokasi persis di mana api berada lewat gambar yang ditangkap oleh kamera. Saat ini kamera pengawas umumnya hanya digunakan untuk menjaga keamanan dari pencurian, bukan pengawasan kebakaran[3].

Untuk memanfaatkan potensi CCTV yang telah dijelaskan sebelumnya, telah dilakukan beberapa penelitian mengenai deteksi api menggunakan pemrosesan citra. Penelitian-penelitian tersebut berusaha menciptakan model yang dapat mendeteksi api di dalam video. Penelitian dilakukan dengan mengumpulkan video kebakaran yang ditangkap oleh CCTV dari internet, kemudian model diuji terhadap video tersebut.

Beberapa penelitian menggunakan algoritma klasifikasi *machine learning* dalam melakukan deteksi api, antara lain metode Convolutional Neural Network (CNN), Artificial Neural Network (ANN), dan K-Nearest Neighbour (KNN). Fitur yang digunakan adalah warna dan tekstur. Beberapa penelitian juga mengikutsertakan deteksi gerak (*motion detection*) untuk meningkatkan akurasi. Model yang ideal adalah yang mampu mencapai akurasi yang tinggi dan waktu komputasi yang rendah.

Dua metode yang kiranya dapat memberikan hasil yang cukup baik dalam memenuhi syarat akurasi dan kecepatan komputasi adalah deteksi gerakan dan segmentasi warna. Algoritma deteksi gerakan yang banyak digunakan karena kecepatan komputasinya adalah *background subtraction* berbasis Gaussian Mixture Model. Selain memiliki kecepatan komputasi yang tinggi, *background subtraction* berbasis Gaussian Mixture Model juga *robust* terhadap *noise* sehingga tidak terlalu sensitif terhadap gerakan-gerakan yang kecil[5]. Untuk segmentasi warna, penelitian sebelumnya telah menunjukkan bahwa ruang warna YCbCr dan HSV dapat memisahkan komponen *luminance* (kecerahan) dan *chrominance* (warna) sehingga efektif dalam mengidentifikasi piksel api. Kedua ruang warna ini juga tidak sensitif terhadap perubahan pencahayaan sehingga dapat bekerja pada kondisi pencahayaan yang tidak seragam (*uneven illumination condition*). Akan tetapi, konversi RGB ke HSV biaya komputasinya lebih tinggi jika dibandingkan konversi RGB ke YCbCr sehingga menggunakan ruang warna YCbCr lebih menguntungkan[6].

Atas dasar tersebut, dalam penelitian ini akan dibangun sebuah model untuk mendeteksi api pada video dengan pendekatan pengolahan citra digital menggunakan Gaussian Mixture Model untuk deteksi gerakan dan segmentasi warna api pada ruang warna YCbCr. Penelitian ini akan mengumpulkan data video kebakaran dalam beragam skala, yaitu kecil, sedang, dan besar. Selain itu, dikumpulkan juga data video yang hanya memiliki objek-objek serupa api. Hasil penelitian diharapkan dapat mendeteksi api dalam skala kecil, sedang, dan besar, dan mampu menghindari pendeteksian yang salah terhadap objek-objek yang serupa dengan api.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka penulis merumuskan beberapa masalah yang akan dibahas yaitu sebagai berikut:

1. Bagaimana cara membangun model pendeteksian api pada video dengan pengolahan citra digital berdasarkan informasi gerakan dan warna?

2. Bagaimana melakukan deteksi api berdasarkan adanya gerakan (*motion*) pada setiap *frame* video?
3. Bagaimana melakukan segmentasi warna api di dalam ruang warna YCbCr?
4. Seberapa optimal metode Gaussian Mixture Model dalam mendeteksi gerakan pada video?
5. Seberapa optimal segmentasi warna api di dalam ruang warna YCbCr?

### **1.3 Batasan Masalah**

Berdasarkan perumusan masalah yang ada, terdapat batasan-batasan masalah sebagai berikut:

1. Data video direkam oleh kamera statis (*stationary*) yang memiliki channel R, G, B dengan panjang antara 10-20 detik dengan format MP4
2. Data video dikumpulkan dari berbagai sumber di internet, yaitu Youtube dan *dataset* penelitian yang dibagikan kepada publik
3. Deteksi api tidak mencakup asap, hanya kobaran lidah api

### **1.4 Tujuan Penelitian**

Berdasarkan latar belakang yang telah diuraikan di atas, maka penulis dapat menguraikan beberapa tujuan yaitu sebagai berikut:

1. Membangun model deteksi api pada video menggunakan pengolahan citra digital berdasarkan informasi gerakan dan warna
2. Melakukan deteksi gerakan dengan metode Gaussian Mixture Model
3. Melakukan segmentasi warna api di dalam ruang warna YCbCr
4. Mengetahui seberapa optimal metode Gaussian Mixture Model dalam mendeteksi gerakan pada video
5. Mengetahui seberapa optimal segmentasi warna api di dalam ruang warna YCbCr

### **1.5 Manfaat Penelitian**

Adapun manfaat yang dapat diperoleh dari penelitian ini yaitu sebagai berikut:

1. Bagi mahasiswa: dapat menerapkan ilmu yang didapat di perkuliahan dan menambah wawasan mengenai Pengolahan Citra Digital.

2. Bagi pembaca: menambah pengetahuan tentang pengolahan citra digital khususnya mengenai deteksi gerakan dan segmentasi warna, dan dapat menjadi referensi untuk penelitian-penelitian selanjutnya.

## **1.6 Sistematika Penulisan**

Laporan tugas akhir ini ditulis dengan sistematika penulisan yang meliputi: Bab I: PENDAHULUAN yang menjelaskan latar belakang penelitian mengenai deteksi api pada video. Pada bab ini juga terdapat rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir; Bab II: TINJAUAN PUSTAKA DAN TEORI berisi tentang tinjauan pustaka yang memuat penelitian-penelitian yang telah dilakukan sebelumnya yang menjadi acuan untuk melakukan penelitian mengenai deteksi api pada video. Bab ini juga membahas teori-teori yang berhubungan dengan penelitian, yang meliputi: api, Gaussian Mixture Model, dan ruang warna YCbCr; BAB III: METODOLOGI PENELITIAN mengandung pembahasan mengenai metodologi yang digunakan untuk mengembangkan model deteksi api pada video, yaitu terdiri atas: alat, bahan, jalannya perancangan dengan metode yang telah ditentukan dan cara pengujian; BAB IV: HASIL DAN PEMBAHASAN membahas tentang hasil dan pembahasan dari penelitian deteksi api pada video, yakni terhadap hasil pengujian dari akurasi model ini; BAB V: KESIMPULAN DAN SARAN berisi kesimpulan yang didapatkan dari uji performa dan kecepatan model deteksi api pada video yang telah dikembangkan, serta saran yang berguna sebagai bahan acuan untuk mengembangkan optimasi algoritma ini pada penelitian lainnya.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian mengenai deteksi api menggunakan pemrosesan citra telah banyak dilakukan sebelumnya. Penelitian [4] menggunakan Background Substraction untuk menentukan objek yang bergerak (api dianggap tidak pernah statis) kemudian ekstraksi fitur dilakukan dengan GLCM untuk masukan ANN. Pengujian pada 15 buah video menunjukkan *error rate* sebesar 7 persen. Akan tetapi, pada jurnal ini tidak dijelaskan bagaimana metode yang diusulkan membedakan benda bergerak yang menyerupai api, dan benda bergerak yang memang merupakan api. Data yang digunakan untuk *training* hanya terdiri dari 15 buah, yang kiranya tidak memadai untuk algoritma Artificial Neural Network. Data video yang digunakan mengandung adegan kebakaran dengan menggunakan bahan bakar berlainan jenis dan jarak pengambilan gambar yang berlainan, selain itu terdapat juga adegan video berupa objek yang menyerupai api.

Beberapa penelitian telah mencoba menggunakan Convolutional Neural Network. Penelitian [7] menggunakan Convolutional Neural Network, tanpa ekstraksi fitur, untuk mendeteksi api dan asap, yang memperoleh akurasi klasifikasi sebesar 97,9 persen terhadap *testing dataset* sebanyak 1427 gambar api dan 1758 gambar asap. Penelitian [8] melatih CNN untuk mengenali api pada gambar. Peneliti membuat dua model *classifier*, untuk mengenali api pada gambar secara global, dan mengenali posisi api secara spesifik di dalam gambar. Akurasi mencapai 97 persen untuk *training set* dan 90 persen untuk *testing set*. Metode CNN memberikan hasil yang baik, akan tetapi memiliki biaya komputasi yang tinggi sehingga kurang ideal untuk diterapkan pada sistem yang bersifat *real-time*.

Penelitian [9] dan [10] menggunakan ruang warna HSV dalam melakukan segmentasi warna api. Penelitian [9] menggunakan Adaptive Gaussian Mixture Model untuk mendeteksi gerakan dengan statistik warna api pada ruang warna HSV untuk mendeteksi kebakaran. Akurasi sistem terhadap video dengan latar siang hari sebesar 97,6 persen sedangkan pada video dengan latar malam hari sebesar 98,65 persen. Kesalahan terutama disebabkan oleh pantulan cahaya kebakaran pada dinding atau permukaan yang mengkilap. Hasil penelitian menunjukkan bahwa waktu komputasi cukup rendah sehingga bisa dilakukan secara *real-time*. Penelitian [9] melakukan pengujian terhadap 30 data video yang dibagi menjadi 15 video siang hari dan 15 video malam

hari. Lokasi yang diuji adalah hutan, bangunan, dan dalam ruangan. Data divariasikan agar mengetahui apakah model dapat mengatasi waktu dan lokasi kebakaran yang berbeda-beda. Di penelitian [10], segmentasi pada ruang warna HSV diterapkan pada robot pemadam api. Eksperimen menunjukkan bahwa algoritma yang digunakan tidak dapat mendeteksi api dikarenakan selalu salah mengidentifikasi lampu sebagai api. Penelitian [10] tidak menggunakan Background Substraction untuk deteksi gerakan.

Deteksi api juga telah dilakukan dengan ekstraksi tekstur. Penelitian [11] menggunakan metode Three-frame Differencing untuk mendeteksi gerakan, ekstraksi tekstur dengan gabungan GLCM dan LBP-three orthogonal, dan pengklasifikasian menggunakan KNN. Hasil pengujian keseluruhan sistem diatas dapat diambil rata-rata akurasi untuk video berisi objek api sebesar 76.70% dan untuk video berisi objek non-api sebesar 65.18%. Akurasi terkecil terjadi pada video dengan api yang sudah mulai padam.

Kekurangan dari algoritma-algoritma yang telah dicoba sebelumnya adalah waktu komputasi yang terlalu tinggi, atau munculnya *false positive* yang disebabkan oleh objek-objek yang menyerupai api. Algoritma yang menggunakan CNN menunjukkan akurasi yang sangat baik namun mahal dari segi komputasi. Algoritma dengan ekstraksi tekstur menggunakan GLCM atau LBP belum bisa mengklasifikasikan dengan benar objek-objek yang menyerupai api (contoh: lampu kendaraan yang berwarna merah). Algoritma dengan segmentasi warna pada ruang warna HSV telah mampu mencapai akurasi yang sangat tinggi, akan tetapi konversi RGB ke HSV membutuhkan biaya komputasi yang tinggi, sehingga dapat dipilih ruang warna lain serupa HSV yang mampu memisahkan *luminance* dan *chrominance* dengan konversi dari RGB yang biaya komputasinya lebih rendah[6].

Sebagai alternatif, segmentasi api bisa dilakukan pada ruang warna YCbCr. Ruang warna ini banyak digunakan dalam deteksi objek yang menggunakan fitur warna. Penelitian [12] melakukan deteksi kulit untuk pengenalan gambar eksplisit menggunakan ruang warna YCbCr. Metode ini dicoba pada seribu gambar dan dapat mengidentifikasi 88.8 % gambar eksplisit dengan nilai *false positive* 5%[12]. Penelitian [13] melakukan deteksi wajah dengan menggunakan segmentasi warna dan *template matching*. Hasil akurasi dari eksperimen yang dilakukan terhadap 300 gambar mencapai 92.3% persen[13].

Ruang warna YCbCr juga pernah digunakan dalam penelitian deteksi api dalam gambar. Penelitian [14] menggunakan metode deteksi berbasis rule menggunakan ruang warna YCbCr. Metode ini telah dicoba pada 400 gambar dan menghasilkan deteksi api 99 persen dengan 14 persen *false positive*. Operasi aritmatik yang dilakukan oleh model ini bersifat linear dengan ukuran gambar sehingga biaya komputasinya rendah. Pada penelitian [15], segmentasi api juga dilakukan pada ruang warna YCbCr, dengan hasil pengujian terhadap 200 gambar menunjukkan bahwa sistem mencapai deteksi api hingga 99.4 persen, dengan false positive sebesar 12 persen. Kedua jurnal ini menggunakan YCbCr dikarenakan ruang warna tersebut mampu memisahkan unsur *luminance* dan *chrominance* pada citra, sehingga hasilnya lebih baik dalam mensegmentasi api berdasarkan warnanya. Hasil deteksi bisa menjadi lebih baik apabila digabungkan dengan metode untuk mendeteksi gerakan pada video, sehingga objek-objek statis yang sewarna dengan api tidak akan dideteksi sebagai api.

Berdasarkan tinjauan di atas, pada penelitian ini akan digabungkan segmentasi api pada ruang warna YCbCr dengan deteksi gerakan pada video menggunakan Gaussian Mixture Model. Diharapkan penggabungan kedua metode ini dapat mengurangi *false positive* yang disebabkan objek-objek yang menyerupai api seperti lampu dan matahari, dengan biaya komputasi yang rendah sehingga dapat diterapkan secara *real-time*.

## **2.2 Teori Penunjang**

### **2.2.1 Api**

Api dalam Kamus Besar Bahasa Indonesia (KBBI) berarti panas dan cahaya yang berasal dari sesuatu yang terbakar[16]. Api merupakan hasil dari reaksi eksotermik kimia yang disebabkan oleh panas atau suatu katalis, sedangkan lidah api (*flame*) merupakan fase gas dari api. Lidah api tidak selalu muncul pada reaksi api[17].

Agar api bisa terjadi, diperlukan tiga hal: panas(*heat*), oksigen, dan bahan bakar. Apabila ada sumber panas, oksigen, dan bahan bakar yang cukup, maka api akan terus menyala, dan memproduksi panas. Panas selalu mengalir dari daerah bersuhu tinggi ke daerah bersuhu rendah. Perpindahan panas ini yang menyebabkan api dapat menyebar[18].

### 2.2.2 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia[19].

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik[19]

Demikian juga agar mesin bisa cerdas (bertindak seperti dan sebaik manusia) maka harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan :

1. Basis Pengetahuan (*Knowledge Base*), bersifat fakta-fakta, teori , pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor Inferensi (*Inference Engine*), kemampuan menarik kesimpulan berdasarkan pengetahuan dan pengalaman.

### 2.2.3 Citra dan Video

Citra adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus intensitas cahaya  $f(x,y)$ , dimana harga  $x$  dan  $y$  merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik  $(x,y)$  merupakan tingkat kecemerlangan citra pada titik tersebut; Pada citra digital, dilakukan diskritisasi koordinat spasial (*sampling*), dan diskritisasi tingkat kecemerlangan/keabuan(kwantisasi). Citra digital direpresentasikan dalam bentuk matriks, di mana indeks baris dan kolom menyatakan suatu titik pada citra dan elemen matriksnya(yang disebut sebagai pixel) menyatakan tingkat keabuan pada titik tersebut[20].

Video adalah teknologi penangkapan, perekaman, pengolahan, dan penyimpanan, pemindahan, dan perekonstruksian urutan gambar diam dengan menyajikan adegan-adegan dalam gerak secara elektronik[21]. Terdapat dua macam video yaitu video analog dan video digital. Video analog dibentuk dari deretan sinyal elektrik (gelombang analog) yang direkam oleh kamera dan dipancarluaskan melalui gelombang udara. Sedangkan videi digital dibentuk dari sederetan sinyal digital yang berbentuk diskrit, yang menggambarkan titik sebagai rangkaian nilai minimum atau maksimum, nilai minimum brarti 0 dan nilai maksimum berarti 1[22].

Sinyal video diciptakan dari hasil luaran sensor kamera yang menangkap citra dua dimensi dan mengubahnya menjadi sinyal elektrik satu dimensi. Gambar bergerak atau video merupakan kumpulan dari gambar-gambar individu, di mana setiap citra yang ditangkap oleh kamera merupakan satu *frame* dalam video. Sinyal ini merupakan sinyal analog[23]. Untuk menciptakan video digital, representasi digital dari sinyanya analog harus dibangun. Dengan kata lain, mereka ulang sinyal analog menjadi sinyal digital. Untuk melakukan ini, sinyal analog harus diukur pada waktu-waktu tertentu dan diberikan nilai numerik. Semakin tinggi frekuensi pengukuran ini, semakin akurat representasi digital dari suatu sinyal analog. Proses ini disebut sebagai *sampling*[24].

#### 2.2.4 CCTV

*Closed Circuit Television* adalah sebuah kamera video digital yang difungsikan untuk memantau dan mengirimkan sinyal video pada suatu ruang yang kemudian sinyal itu akan diteruskan ke sebuah layar monitor. Yang merupakan teknologi kamera pengawasan 24 jam non stop. Pada sistem konvensional dengan VCR (Video Cassette Recorder), awalnya gambar dari kamera CCTV hanya dikirim melalui kabel ke sebuah ruang monitor tertentu dan dibutuhkan pengawasan secara langsung oleh operator (petugas) keamanan dengan resolusi gambar yang masih rendah yaitu 1 image per 12,8 seconds. Namun seiring dengan perkembangan teknologi yang sangat pesat seperti saat ini, banyak kamera CCTV yang telah menggunakan sistem teknologi yang modern. Sistem kamera CCTV digital saat ini dapat dioperasikan maupun dikontrol melalui Personal Computer atau Telephone genggam, serta dapat dimonitor dari mana saja dan kapan saja selama ada komunikasi dengan internet maupun akses GPRS[25].

*Framerate* yang didukung oleh CCTV modern mencapai 30 fps. Framerate ini mampu menangkap gerakan dengan detail yang tajam, akan tetapi membutuhkan *bandwidth* yang besar.

Oleh sebab itu rata-rata *framerate* yang digunakan di industri *surveillance* adalah 15 fps untuk menghemat *bandwidth* tanpa mengurangi kualitas secara signifikan[26].

### 2.2.5 Computer Vision

Computer Vision adalah transformasi data dari suatu gambar diam atau gambar bergerak (video) menjadi sebuah keputusan atau representasi baru. Transformasi tersebut dilakukan untuk mencapai tujuan-tujuan tertentu. Transformasi yang menghasilkan keputusan, contohnya adalah mengenali wajah manusia di dalam gambar, atau membedakan tumor ganas dengan tumor jinak. Transformasi yang menghasilkan representasi baru, misalnya adalah mengubah suatu gambar berwarna menjadi gambar *grayscale*, atau menghilangkan gerakan-gerakan kamera (*camera motion*) yang tidak diinginkan dari suatu video.

Sekalipun hal-hal yang berhubungan dengan penglihatan mampu dilakukan manusia secara intuitif, penglihatan (*vision*) bagi computer terbukti sebagai suatu masalah yang sulit. Hal ini disebabkan karena permasalahan computer vision adalah sebuah *inverse problem*, yaitu mencari suatu parameter yang tidak diketahui menggunakan informasi yang tidak lengkap. Hal ini mustahil dipecahkan secara formal. Oleh sebab itu, sebagian besar solusi dalam bidang computer vision menggunakan model-model probabilistik untuk menemukan solusi-solusi potensial.

Saat ini teknologi *computer vision* telah cukup maju dan banyak diterapkan dalam berbagai permasalahan di dunia nyata, sebagai contoh:

1. Optical Character Recognition(OCR): membaca tulisan tangan manusia dan mengenali plat nomor mobil secara otomatis.
2. Inspeksi Mesin: Memastikan kualitas produk-produk atau perangkat pabrik menggunakan Teknik Stereo Vision.
3. *Retail*: kasir otomatis.
4. *Medical Imaging*: deteksi kanker dan tumor dalam *scan* X-ray.
5. Visual Effect: menggabungkan CGI (Computer Generated Imagery) dengan *live action footage* dengan cara mengikuti (*tracking*) titik-titik tertentu pada video sehingga bisa memperkirakan pergerakan kamera dan lokasi benda-benda di dalam ruang tiga dimensi dan memberikan ilusi objek digital yang terintegrasi dengan objek sungguhan.
6. Pengawasan (*surveillance*): mengawasi rumah dari pencurian, insiden tenggelam di kolam renang, dan kondisi lalu-lintas di jalan raya.

## 2.2.6 Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang paling rendah. Sensasi warna diperoleh dengan adanya interaksi antara warna dengan sistem saraf sensitif warna manusia[20].

## 2.2.7 Segmentasi Warna

Segmentasi Warna (Color filtering) dapat diartikan suatu teknik pengolahan citra yang digunakan untuk memanipulasi suatu citra berdasarkan warna spesifik. Segmentasi dilakukan dengan membandingkan nilai pixel dengan suatu nilai yang spesifik. Warna yang digunakan dalam segmentasi warna dapat direpresentasikan dalam berbagai ruang warna. Dalam penelitian ini, akan dikhususkan pada ruang warna YCbCr.[27]

## 2.2.8 Ruang Warna YCbCr

YCbCr dan RGB merupakan dua ruang warna yang paling banyak digunakan di dalam merepresentasikan citra digital[28]. YCbCr merepresentasikan citra ke dalam satu komponen kecerahan (*luminance*) dan dua komponen warna (kromatik), sedangkan RGB merepresentasikan citra ke dalam tiga komponen *Red*, *Green*, dan *Blue*[28]. Pada YCbCr, luminance disimpan di dalam komponen Y, dan warna disimpan di dalam komponen Cb dan Cr[12].

Ruang warna RGB sangat sensitif terhadap perbedaan intensitas pencahayaan, sehingga ruang warna lain dibutuhkan untuk meningkatkan efektifitas segmentasi warna[28]. YCbCr adalah salah satu ruang warna yang dapat digunakan sebagai alternatif. Beberapa alasan menggunakan ruang warna YCbCr antara lain: komponen Y independen dari komponen warna sehingga bisa mengurangi dampak perbedaan intensitas pencahayaan[28], dan konversi antara RGB dan YCbCr bersifat linear sehingga biaya komputasinya rendah[15]. Adapun konversi dari RGB ke YCbCr dapat dilakukan dengan Persamaan (2-1)[14].

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2-1)$$

## 2.2.9 Background Substraction dengan Gaussian Mixture Model

Background Substraction merupakan pendekatan yang umum digunakan untuk mendeteksi objek yang bergerak di dalam video dengan kamera yang statis (*stationary*). Ide dari pendekatan ini adalah mengidentifikasi objek yang bergerak dengan cara mencari perbedaan antara *frame* saat-ini (*current frame*) dan *frame* referensi, atau yang biasa disebut “*background model*”. *Background model* merupakan representasi dari video yang tidak mengandung objek yang bergerak, dan harus selalu diperbaharui untuk beradaptasi dengan perubahan pada *background* (contoh: perubahan kondisi pencahayaan)[29].

Salah satu metode *background subtraction* adalah dengan menggunakan Gaussian Mixture Model. Metode ini memodelkan *time series* yang terdiri dari nilai-nilai piksel ke dalam Gaussian Mixture Model (GMM). Nilai-nilai suatu piksel yang berubah seiring waktu disebut “*pixel process*”. “*Pixel process*” terdiri atas *time series* dari nilai-nilai suatu piksel, skalar untuk *grayvalues* dan vektor untuk gambar berwarna. Pada waktu  $t$ , diketahui nilai-nilai (*history*) suatu piksel pada posisi  $\{x_0, y_0\}$ , sebagai Persamaan (2-2).

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \quad (2-2)$$

Kita bisa menggunakan data *time series* tersebut untuk memperkirakan probabilitas dari munculnya suatu pixel. Dengan kata lain, kita bisa memperkirakan probabilitas kemunculan suatu pixel dengan membuat model GMM berdasarkan *time series* tersebut.

Probabilitas kemunculan suatu piksel dapat didefinisikan dengan Persamaan (2-3).

$$P(X_t) = \sum_{i=1}^K w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2-3)$$

dimana  $K$  adalah jumlah distribusi,  $w_{i,t}$  merupakan estimasi bobot dari komponen Gaussian ke- $i$  pada waktu  $t$ ,  $\Sigma_{i,t}$  adalah matriks kovarians dari dari komponen Gaussian ke- $i$  pada waktu  $t$ , dan  $\eta$  adalah *probability density function*, dengan Persamaan (2-4).

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (2-4)$$

Setiap piksel baru  $X_t$  dihitung jaraknya dari masing-masing komponen GMM. Apabila jaraknya di bawah 2.5 Standar Deviasi (SD) dari suatu komponen, maka nilai dari suatu piksel dianggap *match* dengan komponen tersebut. Jika tidak ada satupun dari  $K$  komponen yang *match*

dengan suatu piksel, maka komponen dengan bobot terendah dibuang dan digantikan dengan komponen yang nilai rata-ratanya merupakan nilai piksel baru, dengan varian tinggi dan bobot awal rendah.

Bobot untuk komponen  $k$  pada waktu  $t$ ,  $\omega_{k,t}$ , diperbaharui dengan Persamaan (2-5).

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \quad (2-5)$$

Nilai varian dan rata-rata dari komponen yang tidak 'match' tidak berubah. Untuk komponen yang *match*, nilai rata-rata dan varian diperbaharui dengan Persamaan (2-6) dan Persamaan (2-7).

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (2-6)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (2-7)$$

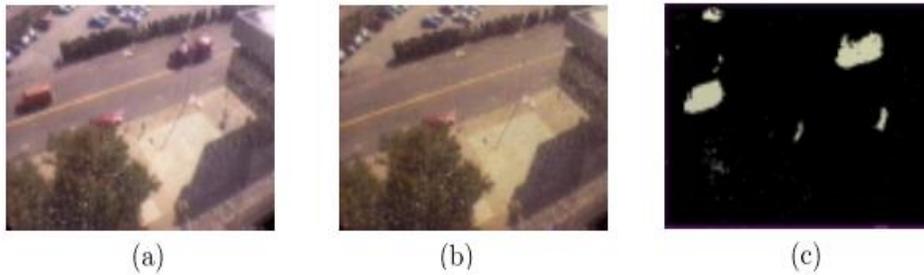
dimana

$$\rho = \alpha N(X_t | \mu_k, \sigma_k)$$

Komponen GMM kemudian diurutkan berdasarkan nilai  $\omega/\sigma$  dari besar ke kecil.  $B$  komponen pertama dipilih sebagai background model berdasarkan Persamaan (2-8).

$$B = \operatorname{argmin}_b (\sum_{k=1}^b \omega_k > T) \quad (2-8)$$

Untuk menentukan apakah suatu piksel merupakan *background* atau *foreground*, dilihat apakah pixel tersebut *match* dengan salah satu  $B$  komponen yang dipilih. Apabila iya, maka piksel dikategorikan sebagai *background*. Apabila tidak, maka piksel dikategorikan sebagai *foreground*[5].



Gambar 2.1 Contoh penerapan dari algoritma. (a) merupakan gambar asli dari frame ke-n, (b) merupakan gambar yang dihasilkan dari distribusi *background*, (c) merupakan segmentasi pixel *foreground*

### 2.2.10 Klasifikasi Piksel Api berdasarkan Model Warna

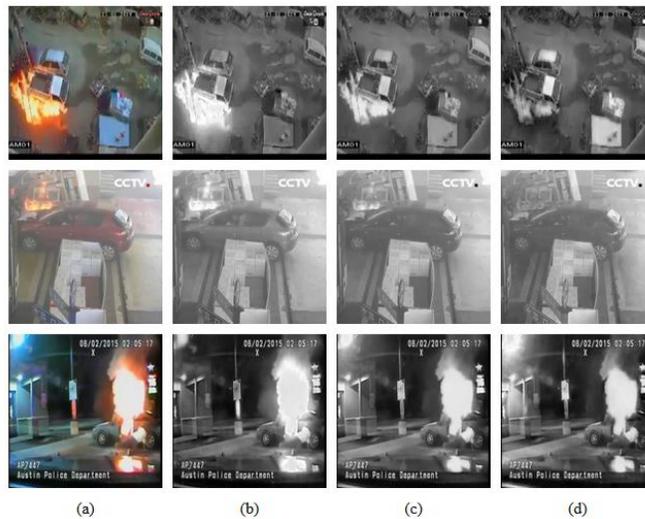
Gambar digital berwarna memiliki tiga *channel*: Red (merah), Green (hijau), dan Blue (biru). Kombinasi dari ketiga warna tersebut membuat suatu *device* mampu menampilkan warna dalam bentuk digital. Setiap *channel* dikuantisasi ke dalam level-level yang diskrit. Pada umumnya warna digital direpresentasikan dalam 8 bit untuk tiap-tiap *channel*. Sebagai contoh, warna putih direpresentasikan oleh  $(R,G,B) = (255,255,255)$  dan hitam direpresentasikan oleh  $(R,G,B) = (0,0,0)$ . Suatu gambar terdiri atas piksel, di mana setiap piksel direpresentasikan oleh suatu lokasi spasial  $(x,y)$  dan vektor warna  $(R(x,y),G(x,y),B(x,y))$  [14].

Klasifikasi piksel api dilakukan menggunakan suatu model warna yang berbasis rules, sebagaimana yang telah dilakukan oleh penelitian [14]. Dalam penelitian tersebut, digunakan tujuh buah *rules* yang menggabungkan informasi dari ruang warna RGB dan YCbCr. Ketujuh *rules* tersebut akan dijelaskan sebagai berikut:

#### 1. Rule 1

Penelitian [14] mendapati bahwa dalam piksel yang merupakan api, intensitas komponen R lebih tinggi dari intensitas G, dan intensitas G lebih tinggi dari intensitas B. *Rule 1* didefinisikan dalam Persamaan (2-9).

$$R_1(x, y) = \begin{cases} 1, & \text{jika } R(x, y) > G(x, y) > B(x, y) \\ 0 & \end{cases} \quad (2-9)$$



Gambar 2.2 Kolom (a) merupakan gambar asli dalam ruang RGB, kolom (b) merupakan channel R, kolom (c) merupakan channel G, kolom (d) merupakan channel B

2. *Rule 2*

Penelitian [14] melakukan analisis histogram terhadap area yang terdapat api dalam gambar, sehingga memperoleh nilai *threshold* dalam ruang warna RGB yang mendeskripsikan piksel yang merupakan api. Berdasarkan analisis histogram yang telah dilakukan oleh [14] terhadap ratusan gambar, didapatkan aturan berikut yang didefinisikan dalam Persamaan (2-10).

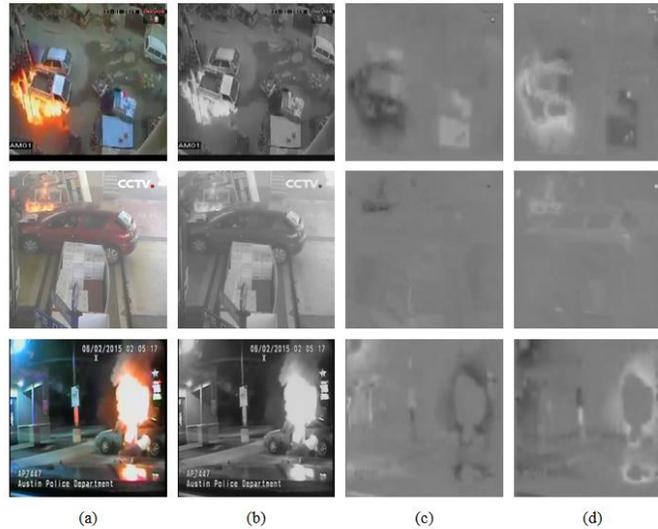
$$R_2(x, y) = \begin{cases} 1, & \text{jika } R(x, y) > 190 \cap G(x, y) > 100 \cap B(x, y) < 140 \\ 0 & \end{cases} \quad (2-10)$$

3. *Rule 3 dan Rule 4*

Untuk *rule 3* dan selanjutnya, klasifikasi piksel dilakukan di dalam ruang warna YCbCr, oleh karena itu diperlukan konversi dari RGB ke YCbCr menggunakan Persamaan (2-1). Setelah dilakukan konversi, maka pada piksel dapat diterapkan Persamaan (2-11) dan Persamaan (2-12).

$$R_3(x, y) = \begin{cases} 1, & \text{jika } Y(x, y) \geq Cb(x, y) \\ 0 & \end{cases} \quad (2-11)$$

$$R_4(x, y) = \begin{cases} 1, & \text{jika } Cr(x, y) \geq Cb(x, y) \\ 0 & \end{cases} \quad (2-12)$$



Gambar 2.3 Kolom (a) merupakan gambar asli dalam ruang RGB, kolom (b), (c), dan (d) secara berturut-turut merupakan *channel* Y,Cb, dan Cr

4. *Rule 5*

Area yang terdapat api pada umumnya merupakan area yang memiliki tingkat kecerahan paling tinggi di dalam suatu gambar. Berdasarkan analisis yang telah dilakukan oleh penelitian [14], didapati bahwa pada suatu piksel yang merupakan api, intensitas Y lebih tinggi dari  $Y_{mean}$ , intensitas Cb lebih tinggi dari  $Cb_{mean}$ , dan intensitas Cr lebih tinggi dari  $Cr_{mean}$ . Pengamatan ini didefinisikan di dalam Persamaan (2-13).

$$R_5(x, y) = \begin{cases} 1, & \text{jika } Y(x, y) \geq Y_{mean} \cap Cb(x, y) \leq Cb_{mean} \cap Cr(x, y) \geq Cr_{mean} \\ 0 & \end{cases} \quad (2-13)$$

5. *Rule 6*

Penelitian [14] mendapati bahwa terdapat perbedaan yang signifikan antara intensitas Cb dan Cr pada piksel api. Piksel api mengandung intensitas Cb yang rendah, sedangkan intensitas Cr tinggi. Pengamatan ini dirangkum dalam Persamaan (2-14).

$$R_6(x, y) = \begin{cases} 1, & \text{jika } |Cb(x, y) - Cr(x, y)| \geq Th \\ 0 & \end{cases} \quad (2-14)$$

Nilai Th yang tepat telah ditentukan oleh Penelitian [16] melalui analisis ROC, yaitu  $Th=70$ .

6. *Rule 7*

Penelitian [14] berhasil menemukan nilai *threshold* untuk komponen Cb dan Cr dari piksel api melalui analisis histogram. Penelitian [14] tidak mempertimbangkan komponen Y sebab intensitasnya dipengaruhi oleh kondisi pencahayaan (*illuminance condition*) pada saat gambar diambil sehingga bisa berubah-ubah. Hasil analisis ini dirangkum dalam Persamaan (2-15).

$$R_7(x, y) = \begin{cases} 1, & \text{jika } Cb(x, y) \leq 120 \text{ dan } Cr(x, y) \geq 150 \\ 0 & \end{cases} \quad (2-15)$$

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Bahan dan Alat Penelitian**

Pada penelitian ini, dibutuhkan bahan dan alat penelitian sebagai sarana pelaksanaan. Bahan-bahan yang digunakan pada penelitian ini adalah citra api dan non api yang disajikan dalam bentuk video sebanyak 60 video dalam format MP4 dengan *framerate* yang divariasikan menjadi 10, 20, dan 30 fps. Lebih lanjut lagi, *dataset* juga akan divariasikan resolusinya menjadi 320x240, 640x480, dan 800x600. *Dataset* dibagi menjadi dua yaitu *dataset* api dan non api, masing-masing sebanyak 30 video. Alat yang digunakan dalam proses penelitian ini terbagi menjadi dua bagian yaitu:

1. Perangkat keras

Perangkat keras yang digunakan dalam penelitian ini adalah *laptop* dengan spesifikasi sebagai berikut:

- a. Prosesor Intel® Core™ i5
- b. Memori RAM 4 GB

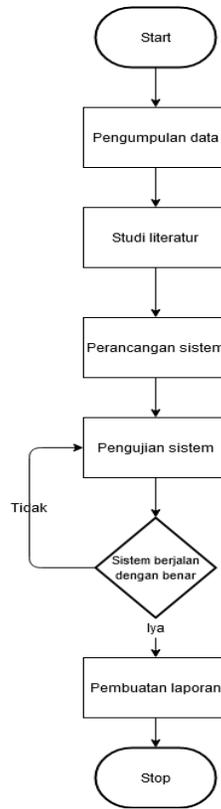
2. Perangkat lunak

Perangkat lunak yang digunakan dalam penelitian ini, yaitu:

- a. Sistem operasi Windows
- b. Python 3

#### **3.2 Rencana Penelitian**

Penelitian dilakukan dalam beberapa tahap secara sistematis, dimulai dari pengumpulan data hingga pembuatan laporan. Tahapan-tahapan tersebut terdiri atas Pengumpulan Data, Studi Literatur, Perancangan Model, dan Pengujian Model. Tahapan-tahapan dalam proses penelitian dapat dilihat pada diagram alir Gambar 3.1.



Gambar 3.1 Diagram Alir Rencana Penelitian

### 3.2.1 Pengumpulan Data

Sebanyak 60 video berdurasi 10-20 detik dikumpulkan dari berbagai sumber di internet. Sumber yang dimaksud seperti Youtube, dan berbagai *repository dataset* yang dibagikan kepada publik untuk keperluan penelitian. Video terdiri atas video yang terdapat api, dan video yang tidak terdapat api, masing-masing sebanyak 30 video.

Video yang berasal dari Youtube didapat dengan kata kunci “api”, “kebakaran”, “kebakaran cctv”, “fire”, “fire cctv”, dan “fire surveillance”. *Repository dataset* yang digunakan adalah [30] dan [31]. Data video yang terdiri atas video yang terdapat api dan tidak terdapat api, masing-masing berjumlah 30, dibagi lagi ke dalam video yang berlatar siang dan malam hari. Hal ini bertujuan untuk menguji apakah model mampu menangani perbedaan pencahayaan antara kondisi malam dan siang. Pembagian data video untuk dapat dilihat dalam Tabel 3.1. Lebih lanjut lagi, data video api diperinci menurut skalanya, yaitu api kecil, api sedang, dan api besar.

Tabel 3.1 Pembagian data video untuk latar siang dan malam

	Siang	Malam	Total
Ada Api	15	15	30
Tidak Ada Api	15	15	30
Total	30	30	60



Gambar 3.2 Contoh data api pada siang dan malam hari

1. Api Kecil

Video api yang berukuran kecil dikarakterisasikan oleh ukuran api yang relatif kecil dibandingkan ukuran api di video lain dalam dataset, namun api masih jelas terlihat oleh mata manusia. Jarak antara api dan kamera bervariasi, dengan jarak antara 5 meter hingga kurang dari 1 meter.



Gambar 3.3 Api kecil

## 2. Api Sedang

Video api yang berukuran sedang dikarakterisasikan oleh ukuran api yang tidak besar, namun juga tidak kecil. Objek yang terbakar bervariasi mulai dari sesemakan hingga barang elektronik. Api berukuran sedang terlihat jelas oleh mata manusia.



Gambar 3.4 Api Sedang

## 3. Api Besar

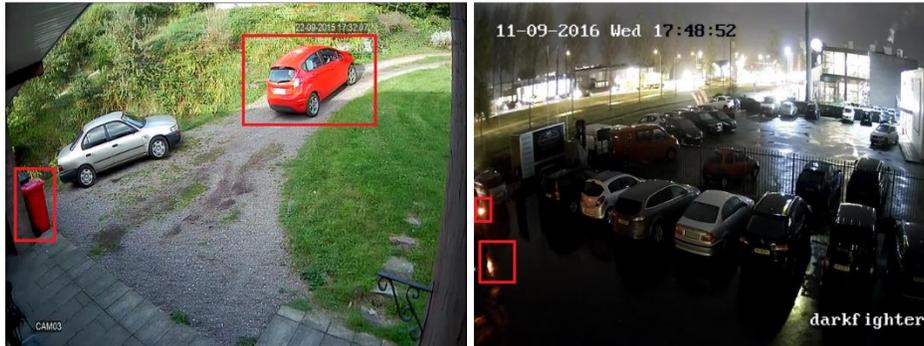
Video api yang berukuran besar dikarakterisasikan oleh ukuran api yang relatif besar dibandingkan objek api di video lain dalam dataset. Video api berukuran besar termasuk kebakaran hutan, kebakaran pom bensin, dan ledakan kendaraan.



Gambar 3.5 Api besar

Video yang tidak mengandung api dicari menggunakan kata kunci “cctv”, “cctv at night”, “cctv night”, dan “cctv malam”. Untuk data video yang tidak mengandung api, diusahakan di dalam video tersebut terdapat setidaknya satu buah objek yang berwarna dengan api, baik yang statis maupun dinamis. Objek-objek yang dianggap menyerupai api antara lain matahari, bunga,

dan objek-objek yang berwarna kemerahan, semisal kendaraan, lampu, pakaian, dan tembok bata[14].



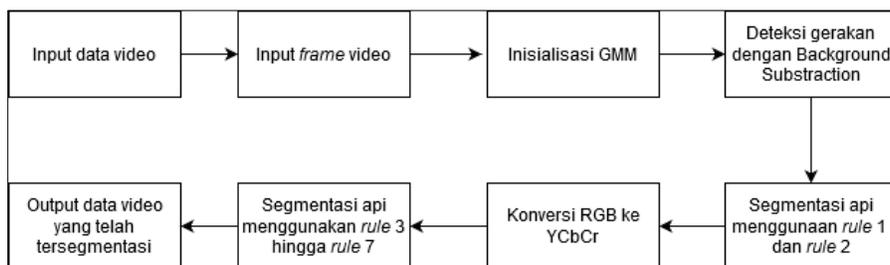
Gambar 3.6 Objek yang menyerupai api

### 3.2.2 Studi Literatur

Guna mendukung berjalannya penelitian, studi literatur dilakukan dengan mempelajari buku-buku, jurnal penelitian serta sumber lain yang berkaitan dengan permasalahan yang diangkat. Adapun materi yang dipelajari dalam studi literatur berkaitan dengan segmentasi warna, ruang warna YCbCr, background subtraction dengan Gaussian Mixture Model serta materi lain yang berkaitan dengan penelitian yang dilakukan.

### 3.2.3 Perancangan Model Deteksi Api pada Video

Secara umum, model deteksi api menggunakan Gaussian Mixture Model dan segmentasi warna api terdiri dari pembacaan *frame* video, inialisasi model untuk Gaussian Mixture Model, deteksi gerakan dengan *background subtraction* berbasis Gaussian Mixture Model, konversi RGB ke YCbCr, dan segmentasi api berdasarkan warna seperti yang ditunjukkan pada Gambar 3.7.



Gambar 3.7 Proses deteksi gerakan dan segmentasi api

Tahapan-tahapan dalam proses deteksi api pada video dijelaskan di dalam subbab berikut.

### 3.2.3.1 Inisialisasi model untuk Gaussian Mixture Model

Gaussian Mixture Model menggunakan beberapa konstanta yang ditentukan pada tahap inisialisasi. Konstanta yang dimaksud adalah *learning rate* ( $\alpha$ ), jumlah distribusi Gaussian ( $K$ ), dan *threshold* ( $T$ ) porsi data yang dianggap sebagai *background*. Pada contoh ini, konstanta ditetapkan sebagai berikut:

1.  $\alpha = 0.05$
2.  $K = 3$
3.  $T = 0.7$

Untuk masing-masing distribusi Gaussian, ditetapkan parameter  $\mu = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ ,  $\sigma^2 = 1.0$  (untuk menghindari *division by zero* ketika mengurutkan distribusi berdasarkan  $\omega/\sigma$ ), dan  $\omega = 0$  (untuk menandakan bahwa tiap distribusi belum merepresentasikan data), sehingga didapatkan parameter tiga distribusi Gaussian sebagai berikut:

1.  $\omega_1 = 0$ ,  $\mu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , dan  $\sigma^2_1 = 1.0$
2.  $\omega_2 = 0$ ,  $\mu_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , dan  $\sigma^2_2 = 1.0$
3.  $\omega_3 = 0$ ,  $\mu_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , dan  $\sigma^2_3 = 36$

### 3.2.3.2 Deteksi Gerakan dengan Background Substraction

Berikut ini diberikan contoh *time series* yang merupakan nilai-nilai piksel dari suatu sekuens gambar, yang masing-masing elemennya berupa vektor untuk merepresentasikan piksel RGB.  $X$  menggambarkan nilai-nilai piksel pada posisi (0,0) dari *frame* 1 hingga *frame* 6.

$$X = \left\{ \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix}, \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix}, \begin{bmatrix} 100 \\ 50 \\ 40 \end{bmatrix}, \begin{bmatrix} 20 \\ 50 \\ 70 \end{bmatrix}, \begin{bmatrix} 20 \\ 40 \\ 70 \end{bmatrix}, \begin{bmatrix} 40 \\ 40 \\ 60 \end{bmatrix} \right\}$$

Pada iterasi pertama, dibuat sebuah distribusi Gaussian baru dengan  $\mu = X_1$ ,  $\sigma^2 = 36$ , dan  $\omega = 0.03$ . Distribusi ini menggantikan distribusi pertama yang telah diinisialisasi pada langkah sebelumnya, sehingga didapatkan tiga distribusi Gaussian sebagai berikut:

$$1. \quad \omega_1 = 0.03, \mu_1 = \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix}, \text{ dan } \sigma^2_1 = 36.0$$

$$2. \quad \omega_2 = 0, \mu_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ dan } \sigma^2_2 = 1.0$$

$$3. \quad \omega_3 = 0, \mu_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ dan } \sigma^2_3 = 36$$

Data  $X_1$  sebagai data pertama dianggap sebagai *foreground*. Ketiga bobot dari distribusi Gaussian dinormalisasi sehingga jumlahnya sama dengan satu, sehingga didapatkan parameter model sebagai berikut:

$$1. \quad \omega_1 = 1, \mu_1 = \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix}, \text{ dan } \sigma^2_1 = 36.0$$

$$2. \quad \omega_2 = 0, \mu_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ dan } \sigma^2_2 = 1.0$$

$$3. \quad \omega_3 = 0, \mu_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ dan } \sigma^2_3 = 36$$

Untuk data selanjutnya, algoritma *background subtraction* melalui langkah-langkah sebagai berikut:

1. Mengurutkan distribusi Gaussian berdasarkan nilai  $\omega/\sigma$ .

Distribusi diurutkan dari besar ke kecil. Oleh karena  $\omega_2$  dan  $\omega_3$  adalah 0, maka jelas bahwa  $\omega_1/\sigma_1 > \omega_2/\sigma_2 \geq \omega_3/\sigma_3$

2. Memilih B distribusi pertama yang merepresentasikan porsi data yang dianggap sebagai *background*, di mana  $B = \operatorname{argmin}_b \sum_{k=1}^b \omega_k > T$

Berdasarkan hasil pengurutan dari langkah 1, maka dipilih distribusi satu sebagai *background*, sebab  $\omega_1 = 1 > T=0.7$ , sehingga didapatkan B (distribusi *background*) dan F (distribusi *foreground*) sebagai berikut:

$$B = \{(\omega_1, \mu_1, \sigma_1)\}$$

$$F = \{(\omega_2, \mu_2, \sigma_2), (\omega_3, \mu_3, \sigma_3)\}$$

3. Menentukan kecocokan nilai piksel  $X_t$  dengan suatu distribusi

Kecocokan piksel  $X_t$  ditentukan dengan menghitung jarak mahalanobis piksel  $X_t$  ke suatu distribusi menggunakan Persamaan (2-12). Apabila jaraknya kurang dari 2.5 SD, maka  $X_t$  dianggap *match* dengan distribusi tersebut, dan variabel  $M_t$  diberi nilai 1. Contoh perhitungan untuk piksel  $X_2$  adalah sebagai berikut:

$K=1$

$$D(X_2) = \sqrt{(X_2 - \mu_1)^T \Sigma^{-1} (X_2 - \mu_1)}$$

$$D(X_2) = \sqrt{\frac{\left( \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix} - \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix} \right)^T \left( \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix} - \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix} \right)}{36.0}}$$

$$D(X_2) = 2.35$$

Sebab  $2.35 < 2.5$ , maka  $X_2$  match dengan distribusi satu, dan  $M_1 = 1$ . Distribusi-distribusi yang lain tidak perlu diperiksa karena secara otomatis dianggap tidak *match*, sehingga nilai  $M_2=0$ , dan  $M_3=0$ .  $X_2$  dianggap sebagai *background* karena *match* dengan salah satu distribusi B.

4. Memperbaharui parameter model

Bobot dari  $K$  distribusi pada waktu  $t$ , wkt, diperbaharui dengan Persamaan (2-5):

$$w_{1,2} = (1 - 0.05) * 1 + 0.05 * 1 = 1$$

$$w_{2,2} = (1 - 0.05) * 0 + 0.05 * 0 = 0$$

$$w_{3,2} = (1 - 0.05) * 0 + 0.05 * 0 = 0$$

Parameter  $\mu$  dan  $\sigma$  dari Distribusi yang *match* dengan data, yaitu distribusi satu, diperbaharui secara berturut-turut dengan Persamaan (2-6) dan Persamaan (2-7). Sebelumnya perlu dihitung terlebih dahulu nilai  $\rho$ , sebagai berikut:

$$\rho = \alpha n(X_2 | \mu_1, \sigma_1)$$

$$\rho = \alpha \left( \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_2 - \mu_1)^T \Sigma^{-1} (X_2 - \mu_1)} \right)$$

$$\rho = 0.05 \left( \frac{1}{(2 * 3.14)^{3/2} 36^{3/2}} e^{-\frac{1}{2} \left( \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix} - \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix} \right)^T (1/36) \left( \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix} - \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix} \right)} \right)$$

$$\rho = 9.1 * 10^{-8}$$

Pembaharuan parameter  $\mu$  dengan Persamaan (2-6) sebagai berikut:

$$\mu_{12} = 1 - 9.1 * 10^{-8} * \begin{bmatrix} 20 \\ 40 \\ 60 \end{bmatrix} + 9.1 * 10^{-8} * \begin{bmatrix} 30 \\ 40 \\ 70 \end{bmatrix} = \begin{bmatrix} 19.99 \\ 39.99 \\ 59.99 \end{bmatrix}$$

Pembarahuan parameter varian dengan Persamaan (2-7):

$$\sigma_{12} = (1 - 9.1 * 10^{-8}) * 36 + 9.1 * 10^{-8} * \begin{bmatrix} 10 & 0 & 10 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \\ 10 \end{bmatrix} = 36$$

Menggunakan langkah-langkah yang sama untuk data berikutnya, maka didapatkan hasil sebagai berikut:

$$X = [F, B, F, B, B, F], \text{ dimana } F = \text{foreground dan } B = \text{background}$$

### 3.2.3.3 Konversi RGB ke YCbCr

Konversi RGB ke YCbCr dilakukan dengan menggunakan Persamaan (2-1). Sebagai contoh, diberikan sebuah vektor  $X_{RGB} = [160 \ 200 \ 160]^T$  yang merepresentasikan nilai suatu piksel dari sebuah gambar berwarna. Vektor ini diubah dari RGB ke YCbCr menggunakan Persamaan (2-1), sehingga didapatkan:

$$X_{YCbCr} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} 160 \\ 200 \\ 160 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} = \begin{bmatrix} 174 \\ 116 \\ 113 \end{bmatrix}$$

### 3.2.3.4 Segmentasi Api Berdasarkan Warna

Diberikan sebuah matriks M berdimensi 3x3 yang merepresentasikan sebuah gambar digital dalam ruang warna RGB:

$$M = \begin{bmatrix} (239,121,68) & (152,93,35) & (143,119,95) \\ (93,188,210) & (193,138,73) & (110,86,74) \\ (128,110,90) & (137,72,18) & (73,54,41) \end{bmatrix}$$

Tiap piksel di dalam matriks M diklasifikasikan sebagai piksel api atau bukan menggunakan Persamaan (2-9) hingga Persamaan (2-15). Apabila suatu piksel memenuhi ketujuh syarat yang didefinisikan di dalam persamaan-persamaan tersebut, maka piksel dianggap piksel api. Apabila

salah satu dari ketujuh syarat tidak terpenuhi, maka piksel dianggap bukan piksel api. Berikut ini diberikan contoh perhitungan untuk piksel  $M(0,0) = (239,121,68)$ .

1. *Rule 1*

Persamaan (2-9) menyatakan bahwa piksel api memiliki intensitas R lebih besar dari intensitas G, dan intensitas G lebih besar dari intensitas B. Pada piksel  $M(0,0)$ ,  $R = 239$ ,  $G=121$ , dan  $B=68$ . Perhitungan untuk Persamaan (2-9) dijabarkan sebagai berikut:

$$R(0,0) > G(0,0) > B(0,0) \text{ maka } R_1(0,0) = 1$$

2. *Rule 2*

*Rule 2* mendefinisikan nilai threshold piksel api untuk komponen R,G, dan B. Nilai threshold ini ditetapkan dalam Persamaan (2-10). Perhitungan untuk piksel  $M(0,0)$ :

$$R_2(x,y) = \begin{cases} 1, & \text{jika } R(0,0) > 190 \cap G(0,0) > 100 \cap B(0,0) < 140 \\ 0 & \end{cases}$$

$$R(0,0) = 239 > 190 \cap G(0,0) = 121 > 100 \cap B(0,0) = 68 < 140, \text{ maka } R_2(0,0) = 1$$

3. *Rule 3*

*Rule 3* dan selanjutnya membutuhkan informasi piksel dalam ruang warna YCbCr, sehingga diperlukan konversi dari RGB ke YCbCr menggunakan Persamaan (2-1). Berikut diberikan contoh konversi RGB ke YCbCr pada piksel  $M(0,0)$ :

$$M(0,0)_{YCbCr} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -18.214 \end{bmatrix} \cdot M(0,0)_{RGB} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

$$M(0,0)_{YCbCr} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -18.214 \end{bmatrix} \cdot \begin{bmatrix} 239 \\ 121 \\ 68 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

$$M(0,0)_{YCbCr} = \begin{bmatrix} 145 \\ 87 \\ 184 \end{bmatrix}$$

Setelah menerapkan Persamaan (2-1) untuk piksel-piksel yang lain di dalam matriks M, maka didapatkan matriks M dalam ruang warna YCbCr:

$$M = \begin{bmatrix} (145,87,184) & (105,94,158) & (122,114,140) \\ (155,152,85) & (142,91,157) & (95,119,139) \\ (113,117,137) & (89,95,160) & (66,119,137) \end{bmatrix}$$

*Rule 3* membandingkan nilai intensitas  $Y(x,y)$  dan  $Cb(x,y)$  yang terangkum dalam Persamaan (2-11). Perhitungan untuk piksel  $M(0,0)$ :

$$R_3(0,0) = \begin{cases} 1, & \text{jika } Y(0,0) \geq Cb(0,0) \\ 0 & \end{cases}$$

$$Y(0,0) = 145 \geq Cb(0,0) = 87, \text{ maka } R_3(0,0) = 1$$

4. *Rule 4*

*Rule 4* membandingkan nilai intensitas Cr dan Cb sebagaimana terangkum dalam Persamaan (2-12).

$$R_4(0,0) = \begin{cases} 1, & \text{jika } Cr(0,0) \geq Cb(0,0) \\ 0 & \end{cases}$$

$$Cr(0,0) = 184 \geq Cb(0,0) = 87, \text{ maka } R_4(0,0) = 1$$

5. *Rule 5*

*Rule 5* membandingkan nilai Y, Cb, dan Cr dari suatu piksel dengan nilai  $Y_{mean}$ ,  $Cb_{mean}$ , dan  $Cr_{mean}$  dari gambar secara keseluruhan sebagaimana terangkum dalam Persamaan (2-13). Sebelumnya perlu dihitung terlebih dahulu nilai  $Y_{mean}$ ,  $Cb_{mean}$ , dan  $Cr_{mean}$  sebagai berikut:

$$Y_{mean} = \frac{\sum_{x=0}^8 \sum_{y=0}^8 Y(x, y)}{9} = 114.66$$

$$Cb_{mean} = \frac{\sum_{x=0}^8 \sum_{y=0}^8 Cb(x, y)}{9} = 109.77$$

$$Cr_{mean} = \frac{\sum_{x=0}^8 \sum_{y=0}^8 Cr(x, y)}{9} = 144.11$$

Berdasarkan Persamaan (2-13), dilakukan perhitungan terhadap piksel M(0,0):

$$R_5(0,0) = \begin{cases} 1, & \text{jika } Y(0,0) \geq Y_{mean} \cap Cb(0,0) \leq Cb_{mean} \cap Cr(0,0) \geq Cr_{mean} \\ 0 & \end{cases}$$

$$Y(0,0) = 145 \geq Y_{mean} = 114.66 \cap Cb(0,0) = 87 \leq Cb_{mean} = 109.77 \cap Cr(0,0) = 184 \geq Cr_{mean} = 144.11, \text{ maka } R_5(0,0) = 1$$

6. *Rule 6*

*Rule 6* mengklasifikasikan piksel berdasarkan selisih antara Cb dan Cr, dan membandingkannya dengan suatu nilai *threshold*. *Rule 6* terangkum dalam Persamaan (2-14).

$$R_6(0,0) = \begin{cases} 1, & \text{jika } |Cb(0,0) - Cr(0,0)| \geq 70 \\ 0 & \end{cases}$$

$$|87 - 184| = 97 \geq 70, \text{ maka } R_6(0,0) = 1$$

7. *Rule 7*

*Rule 7* mengklasifikasikan piksel berdasarkan *threshold* Cb dan Cr sebagaimana terangkum dalam Persamaan (2-15). Contoh perhitungan terhadap piksel M(0,0):

$$R_7(0,0) = \begin{cases} 1, & \text{jika } Cb(0,0) \leq 120 \text{ dan } Cr(0,0) \geq 150 \\ 0 & \end{cases}$$

$$Cb(0,0) = 87 \leq 120 \text{ dan } Cr(0,0) = 184 \geq 150, \text{ maka } R_7(0,0) = 1$$

Setelah menguji piksel terhadap ketujuh *rules*, maka didapatkan:

$$R_1(0,0) \cap R_2(0,0) \cap R_3(0,0) \cap R_4(0,0) \cap R_5(0,0) \cap R_6(0,0) \cap R_7(0,0) = 1 \cap 1 \cap 1 \cap 1 \cap 1 \cap 1 \cap 1 \\ = 1$$

Dapat disimpulkan bahwa piksel M(0,0) adalah piksel api karena memenuhi ketujuh *rules*.

### 3.2.4 Skenario Pengujian

Tahapan pengujian merupakan tahapan yang ditujukan sebagai bahan evaluasi untuk mengetahui apakah model yang dibuat telah berhasil bekerja sesuai dengan ekspektasi. Adapun yang menjadi tolak ukur keberhasilan adalah performa model yang diukur dengan akurasi, *precision*, dan *recall*, serta kecepatan komputasi. Hasil pengujian dapat menjadi bahan analisis untuk pengembangan selanjutnya.

Pengujian dilakukan terhadap 60 data video, yang terbagi dua menjadi video terdapat api dan yang tidak terdapat api. Masing-masing video berdurasi 10-20 detik. Adapun penjelasan mengenai jenis-jenis pengujian yang akan dilakukan, dijabarkan dalam subbab berikut.

#### 3.2.4.1 Pengujian Akurasi Deteksi Api

Pada tahap ini dicoba untuk menemukan parameter model yang terbaik. Dikarenakan *rules* segmentasi YCbCr telah menggunakan *threshold* yang optimal dari penelitian sebelumnya, maka parameter yang akan diubah adalah pada Gaussian Mixture Model.

##### 1. Jumlah distribusi di dalam GMM (K)

Jumlah distribusi di dalam GMM menentukan berapa banyak distribusi yang digunakan untuk memodelkan sebuah *pixel process*. Nilai yang disarankan untuk K adalah 3,4, atau 5[5].

##### 2. Rasio data yang dianggap *background* (T)

Parameter ini menentukan bagian data yang dianggap sebagai *background*. Dikarenakan pada sebuah video diasumsikan bahwa rasio *background* lebih banyak daripada *foreground*, maka nilai T yang akan diuji adalah 0.7, 0.8, dan 0.9.

3. *Learning rate* ( $\alpha$ )

Parameter ini menentukan seberapa cepat model memperbaharui dirinya atas masukan data baru. Nilai yang sudah pernah diuji dan memberikan hasil yang baik dalam kasus deteksi api adalah 0.005[9].

Selain itu, efek dari *framerate* video juga akan diuji terhadap model. *Framerate* video boleh jadi akan memengaruhi cara GMM dalam mensegmentasi gerakan. Ini berkaitan dengan *learning rate* dan seberapa cepat model memperbaharui dirinya atas masukan data baru. *Framerate* yang rendah berarti masukan data yang lebih sedikit bagi GMM per detik video, yang akan mempengaruhi proses pembaharuan model. *Framerate* yang akan diuji adalah 10, 20, dan 30 fps.

Performa dari model yang telah dikembangkan dinilai dengan tiga buah parameter yaitu akurasi, *precision*, dan *recall*. Nilai ketiga parameter tersebut menjadi tolak ukur performa model dalam mendeteksi api. Performa dari model menjadi gambaran seberapa optimal penggunaan Gaussian Mixture Model dan ruang warna YCbCr dalam mendeteksi api pada video.

Tabel 3.2 Confusion matrix 2x2

		Predict class	
		A	B
Actual class	A	AA	AB
	B	BA	BB

Tabel 3.2 akan menjadi acuan untuk melakukan perhitungan terhadap performa dari model yang dihasilkan. Kelas A dan B secara berturut-turut merepresentasikan *frame* yang terdapat api dan *frame* yang tidak terdapat api. Nilai akurasi, *precision*, dan *recall* dihitung dengan persamaan-persamaan berikut:

$$\text{Akurasi} = \frac{TP+TN}{(TP+FP+FN+TN)} * 100 \% \quad (3-1)$$

$$\text{Precision} = \frac{TP}{(TP+FP)} * 100 \% \quad (3-2)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} * 100 \% \quad (3-3)$$

### 3.2.4.2 Pengujian Kecepatan Algoritma Deteksi Api

Pengujian dilakukan dengan mengukur kecepatan perhitungan algoritma pada setiap *frame* video, yang ditentukan oleh jumlah *frame* yang dapat dieksekusi per detik (*frame per second* atau *fps*). Video diubah ke dalam kategori 320x240, 640x480, dan 800x600. Hal ini dilakukan untuk mengetahui apakah model mampu memproses beragam resolusi video secara *real-time*. Apabila jumlah *frame* yang mampu dieksekusi model lebih besar daripada *frame rate* video, maka model dianggap mampu bekerja secara *real-time*.

### 3.3 Jadwal Kegiatan.

Waktu yang digunakan dalam proses pembuatan model deteksi api yaitu selama lima bulan. Jadwal kegiatan pembuatan model deteksi api dapat dilihat di Tabel 3.3.

Tabel 3.3 Jadwal kegiatan perancangan sistem

No	Kegiatan	Waktu (Bulan)					Keterangan
		I	II	III	IV	V	
1	Analisa	■					Analisa kebutuhan
2	Perancangan		■	■			Perancangan sistem
3	<i>Coding</i>			■	■	■	Pengkodean sistem
4	<i>Testing</i>				■	■	Pengujian sistem
5	Dokumentasi	■	■	■	■	■	Dokumentasi sistem

## DAFTAR PUSTAKA

- [1] L. Tacconi, “Kebakaran hutan di Indonesia: penyebab, biaya dan implikasi kebijakan,” *CIFOR Occas. Pap.*, vol. 38, no. 38, 2003, doi: 10.17528/cifor/001200.
- [2] S. S. Dewi, D. Satria, E. Yusibani, and D. Sugiyanto, “Prototipe Sistem Informasi Monitoring Kebakaran Bangunan Berbasis Google Maps dan Modul GSM,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 1, no. 1, p. 33, 2017, doi: 10.35870/jtik.v1i1.31.
- [3] H. A. Muzakkiy, “Deteksi Api Berbasis Sensor Visual Menggunakan Metode Support Vector Machines,” Institut Teknologi Sepuluh November, 2016.
- [4] Andi, V. Suhartono, and R. A. Pramunendar, “Deteksi Api Menggunakan Background Substraction Dan Artificial Neural Network Untuk Real Time Monitoring,” *J. Teknol. Inf.*, vol. 12, no. April, pp. 15–24, 2016.
- [5] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 246–252, 1999, doi: 10.1109/cvpr.1999.784637.
- [6] K. B. Shaik, P. Ganesan, V. Kalist, B. S. Sathish, and J. M. M. Jenitha, “Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space,” *Procedia Comput. Sci.*, vol. 57, pp. 41–48, 2015, doi: 10.1016/j.procs.2015.07.362.
- [7] S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, and F. Fnaiech, “Convolutional neural network for video fire and smoke detection,” *IECON 2016 - 42nd Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 877–882, 2016, doi: 10.1109/IECON.2016.7793196.
- [8] Q. Zhang, J. Xu, L. Xu, and H. Guo, “Deep Convolutional Neural Networks for Forest Fire Detection,” *Int. Forum Manag. Educ. Inf. Technol. Appl.*, pp. 568–575, 2016, doi: 10.2991/ifmeita-16.2016.105.
- [9] A. Prahara, “Deteksi Kebakaran pada Video Berbasis Pengolahan Citra dengan Dukungan GPU,” *Semin. Nas. Apl. Teknol. Inf.*, p. 6, 2015.
- [10] R. R. Suryadi, I. Wijayanto, A. Rusdinar, F. T. Elektro, and C. Detection, “Perancangan dan implementasi sistem pendeteksi api pada robot pemadam api dengan menggunakan sensor api dan kamera,” *e-Proceeding Eng.*, vol. 4, no. 3, pp. 3611–3624, 2017.
- [11] G. L. Khamdani, T. A. B. Wirayuda, and F. Sthevanie, “Sistem Deteksi Api Berbasis Visual menggunakan Metode Local Binary Patterns-Three Orthogonal Planes dan Grey-level Co-occurrence Matrix,” 2015.
- [12] J. A. M. Basilio, G. A. Torres, G. S. Pérez, L. K. T. Medina, and H. M. P. Meana, “Explicit image detection using YCbCr space color model as skin detection,” *Appl. Math. Comput. Eng. - Am. Conf. Appl. Math. Am. 5th WSEAS Int. Conf. Comput. Eng. Appl. CEA’11*, no. December, pp. 123–128, 2011.
- [13] A. Chen, L. Pan, Y. Tong, and N. Ning, “Face detection technology based on skin color segmentation and template matching,” *2nd Int. Work. Educ. Technol. Comput. Sci. ETCS 2010*, vol. 2, pp. 708–711, 2010, doi: 10.1109/ETCS.2010.465.

- [14] V. Vipin, "Image Processing Based Forest Fire Detection," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 2, pp. 87–95, 2012.
- [15] V. V K and S. C. Sajjan, "Fire Detection using YCbCr Color Model," *IJRSET*, vol. 2, no. 2, pp. 698–701, 2016.
- [16] Pusat Bahasa, *Kamus Besar Bahasa Indonesia*, vol. 66. 2008.
- [17] J. G. Quintiere, *Fundamentals of Fire Phenomena*. New Jersey: John Wiley & Sons, LTD, 2006.
- [18] P. J. Fardell and C. Lukas, "Understanding fire," *Chemistry in Britain*, 1987. [https://www.crookston.mn.us/Fire/SafetyTips/Understanding Fire.htm](https://www.crookston.mn.us/Fire/SafetyTips/Understanding%20Fire.htm) (accessed Nov. 25, 2019).
- [19] Muhammad Dahria, "Kecerdasan buatan ( Artificial Intelligence )," *Artif. Intell.*, vol. 1, no. 2, pp. 1–10, 2014.
- [20] H. D. Hutahaean, "Teknik Penajaman Citra Digital Dengan Menggunakan Metode Contrast Stretching," *Pelita Inform. Budi Darma*, vol. III, pp. 35–44, 2013.
- [21] M. Fadhli, "Pengembangan Media Pembelajaran Berbasis Video Kelas Iv Sekolah Dasar," *J. Dimens. Pendidik. dan Pembelajaran*, vol. 3, no. 1, pp. 24–29, 2015.
- [22] C. Prihantini and G. K. Nugroho, "Pembuatan Video Profil Sekolah Menengah Pertama Negeri 4 Polokarto Kabupaten Sukoharjo Dengan Menggunakan Komputer Multimedia," *J. Teknol. dan Sist. Komput.*, vol. 5, no. 1, pp. 1–5, 2013, doi: 10.9744/jmk.16.1.83.
- [23] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, vol. 18, no. 1. 2003.
- [24] M. Weise, *How Video Works, Second Edition: From Analog to High Definition*. 2007.
- [25] V. Dwisantri, "Rancang Bangun Alat Sistem Kendali Monitoring CCTV Menggunakan Mikrokontroler pada Laboratorium Jurusan Teknik Komputer," 2019.
- [26] J. Honovich, "Frame Rate Guide for Video Surveillance," *IPVM*, 2014. <https://ipvm.com/reports/frame-rate-surveillance-guide> (accessed Apr. 27, 2020).
- [27] D. Hardiyanto and D. A. Sartika, "Ekstraksi Fitur Citra Api Berbasis Ekstraksi Warna pada Ruang Warna HSV dan RGB," *FAHMA (Jurnal Teknol. Inf. Dan Ilmu Komputer)*, vol. 16, no. 03, 2018.
- [28] C. Lin, "Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network," *Pattern Recognit. Lett.*, vol. 28, no. 16, pp. 2190–2200, 2007, doi: 10.1016/j.patrec.2007.07.003.
- [29] M. Piccardi, "Background subtraction techniques: A review," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 4, pp. 3099–3104, 2004, doi: 10.1109/ICSMC.2004.1400815.
- [30] "Index of /VisiFire/Demo/FireClips." <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/FireClips/> (accessed May 16, 2020).
- [31] "GitHub - sulenn/fire-dataset: various of fire pictures and videos in the repository."

<https://github.com/sulenn/fire-dataset> (accessed May 16, 2020).

# LAMPIRAN

## Lembar Asistensi

**Notifications** 

### [ta] Editor Decision

2020-03-27 01:49 PM

Ristirianto Adi:

We have reached a decision regarding your submission to Publikasi Tugas Akhir S-1 PSTI FT-UNRAM, "Deteksi Api pada Video dengan Gaussian Mixture Model untuk Deteksi Gerakan dan Segmentasi Warna Api dalam Ruang Warna YCbCr".

Our decision is to: Accept Submission

I Gede Pasek Suta Wijaya  
gpsutawijaya@unram.ac.id

Note:

**Proposal TA sudah disetujui, gunakan notifikasi ini untuk melakukan pendaftaran Seminar TA1.**

---

[Publikasi Tugas Akhir S-1 PSTI FT-UNRAM](#)