

**USULAN TUGAS AKHIR**

**DETEKSI API PADA VIDEO MENGGUNAKAN METODE ARTIFICIAL  
NEURAL NETRWORK**



**Oleh :**

**BUDIMAN RABBANI  
F1D 016 018**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MATARAM  
JANUARI 2020**

**Usulan Tugas Akhir**  
**DETEKSI API PADA VIDEO MENGGUNAKAN METODE ARTIFICIAL**  
**NEURAL NETRWORK**

Oleh:

**BUDIMAN RABBANI**  
**F1D 016 018**

Telah diperiksa dan disetujui oleh Tim Pembimbing:

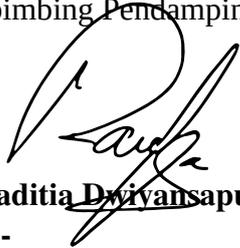
1. Pembimbing Utama



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

Tanggal: \_\_\_\_\_

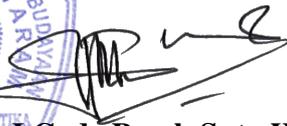
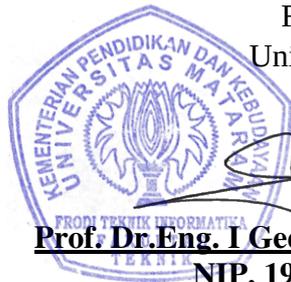
2. Pembimbing Pendamping



**Ramaditia Dwiyan saputra, ST., M.Eng.**  
**NIP. -**

Tanggal: 2 April 20

Mengetahui  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



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

**Usulan Tugas Akhir**  
**DETEKSI API MENGGUNAKAN METODE ARTIFICIAL NEURAL**  
**NETRWORK**

Oleh:  
**BUDIMAN RABBANI**  
**FID 016 018**

Telah diperiksa dan disetujui oleh Tim Penguji:

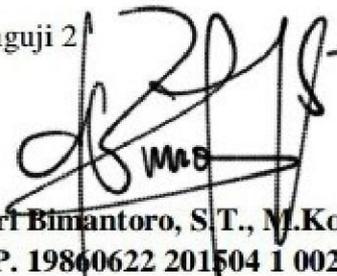
1. Penguji 1



**Arik Aranta, S.Kom., M.Kom.**  
**NIP. 19940220 201903 1 004**

Tanggal: 2 april 2020

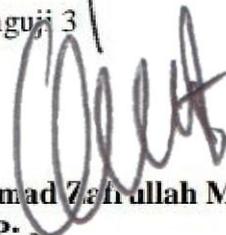
2. Penguji 2



**Fitri Bimantoro, S.T., M.Kom.**  
**NIP. 19860622 201504 1 002**

Tanggal: 2 april 2020

3. Penguji 3



**Ahmad Zahidullah Mardiansyah S.T., M. Eng.**  
**NIP: -**

Tanggal: 2 april 2020

Mengetahui  
Ketua Program Studi Teknik Informatika  
Fakultas Teknik  
Universitas Mataram



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

## Daftar Isi

<b>HALAMAN JUDUL</b>	
<b>HALAMAN PENGESAHAN</b>	
<b>PRAKATA</b> .....	i
<b>DAFTAR ISI</b> .....	iii
<b>DAFTAR GAMBAR</b> .....	iv
<b>DAFTAR TABEL</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>INTISARI</b> .....	vii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
1.6 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI</b> .....	5
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori .....	7
3.3.1 Api .....	7
3.3.2 Kecerdasan Buatan.....	8
3.3.3 <i>Gray-Level Co-Occurrence Matrix</i> .....	9
3.3.4 <i>Local Binary Pattern</i> .....	11
3.3.5 Konsep Dasar Jaringan Saraf Tiruan .....	12
3.3.6 Arsitektur Jaringan Syaraf Tiruan .....	13
3.3.7 Fungsi Aktivasi <i>Backpropegation</i> .....	14
3.3.8 Pelatihan Standar <i>Backpropegation</i> .....	15
<b>BAB III METODE PENELITIAN</b> .....	17
3.1 Bahan dan Alat Penelitian .....	17
3.2 Rencana Penelitian .....	18
3.3 Rancangan Sistem Deteksi Api .....	19
3.3.9 Tahap Preprocessing .....	19
3.3.10 <i>Extraction Feature</i> .....	21

3.3.11	Klasifikasi .....	25
3.4	Teknik Pengujian Sistem .....	31
3.4.1	Skenario Pengujian Sistem.....	31
3.5	Jadwal Kegiatan.....	33
<b>DAFTAR PUSTAKA .....</b>		<b>35</b>

## Daftar Gambar

Gambar 2. 1 contoh gambar adanya objek api .....	8
Gambar 2. 2 Penerapan Konsep Kecerdasan Buatan di Komputer[20] .....	9
Gambar 2. 3 Arsitektur Lapisan Tunggal .....	13
Gambar 2. 4 Arsitektur Lapisan Multilayer .....	14
Gambar 2. 5 Fungsi Aktivasi Sigmoid Biner .....	14
Gambar 2. 6 Fungsi Aktivasi Sigmoid Bipolar .....	15
Gambar 3. 1 Diagram alir pembuatan sistem .....	18
Gambar 3. 2 Proses training (pelatihan), testing (pengujian) dan klasifikasi. ....	19
Gambar 3. 3 Citra awal .....	20
Gambar 3. 4 Hasil segmentasi .....	20
Gambar 3. 5 Proses Cropping .....	20
Gambar 3. 6 Citra 4x4 .....	21
Gambar 3. 7 (a) penentuan titik pusat index(1,1) (b) hasil threshold .....	21
Gambar 3. 8 (a) weights (b)hasil perubahan nilai sesuai weights(b) hasil akhir komputasi LBP pixel(1,1) .....	22
Gambar 3. 9 Hasil akhir perhitungan	22
Gambar 3. 10 Contoh matriks citra grayscale .....	22
Gambar 3. 11 Matriks GLCM 3x3 .....	23
Gambar 3. 12 Pembentukan matriks GLCM sudut 0° .....	23
Gambar 3. 13 Arah matriks GLCM a. Sudut 45°. (b) Sudut 90°. (c) Sudut 135° ...	24
Gambar 3. 14 Hasil pembentukan matriks GLCM (a) $\theta = 0^\circ d = 1$ . (b) $\theta = 45^\circ d =$ 1. (c) $\theta = 90^\circ d = 1$ . (d) $\theta = 135^\circ d = 1$ .....	24
Gambar 3. 15 Matriks GLCM yang telah dinormalisasi .....	24
Gambar 3. 16 Arsitektur backpropagation .....	25

## Daftar Tabel

Tabel 2.1 Tabel fitur tekstur GLCM. ....	12
Tabel 3. 1 Contoh data frame video .....	17
Tabel 3. 2 Data dummy .....	25
Tabel 3. 3 Bias dan bobot akhir dari <i>input layer</i> ke <i>hidden layer</i> pertama .....	30
Tabel 3. 4 Bias dan bobot akhir dari <i>hidden layer</i> pertama ke <i>output layer</i> .....	30
Tabel 3. 5 <i>Output</i> data latih .....	31
Tabel 3. 6 Confusion matrix 3x3.....	32
Tabel 3. 7 Hasil deteksi .....	33
Tabel 3. 8 Jadwal kegiatan perancangan sistem.....	34

## **ABSTRACT**

*Fire is the result of a chemical reaction between fuel and oxygen. Fire is often used by people for cooking, lighting the room and so forth. However, fire can pose a danger if the fire cannot be controlled. Therefore, many tools have been developed to support sensors such as fire. However, this research used camera catches to promote fire. Therefore, using the capture camera will see the best model of backpropagation neural network by combining the local binary pattern (LBP) feature extraction method and the Gray Level Co-occurrence Matrix (GLCM) to access the fire. Then to evaluate the performance of the model created using three parameters that contain accuracy, recall, precision.*

**Keyword: Fire Detection, GLCM, LBP, ANN Backpropagation.**

## INTISARI

Api merupakan sebuah hasil dari reaksi kimia antara bahan bakar dan oksigen. Api sering kali dimanfaatkan oleh masyarakat untuk memasak, penerangan ruangan dan lain sebagainya. Namun api juga berpotensi menimbulkan bahaya jika api tidak dapat dikontrol dengan baik sehingga dapat menyebabkan kebakaran. Oleh karena itu, sudah banyak alat yang dikembangkan untuk mendeteksi kebakaran seperti sensor. Namun pada penelitian ini digunakan hasil tangkapan kamera untuk mendeteksi api. Oleh karena itu, dengan memanfaatkan tangkapan kamera akan dilihat model yang terbaik *artificial neural network backpropagation* dengan penggabungan metode ekstraksi fitur gabungan *local binary patern* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM) untuk mendeteksi api. Kemudian untuk mengevaluasi kinerja dari model yang dibuat digunakan tiga buah parameter yakni akurasi, *recall*, *precision*.

**Kata kunci:** Deteksi Api, GLCM, LBP, ANN *Backpropagation*.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Api merupakan sebuah hasil dari reaksi kimia antara bahan bakar dan oksigen. Api sering kali dimanfaatkan oleh masyarakat untuk memasak, penerangan ruangan dan lain sebagainya. Namun api juga berpotensi menimbulkan bahaya jika api tidak dapat dikontrol dengan baik sehingga dapat menyebabkan kebakaran.

Kebakaran hutan merupakan hal yang selalu menjadi sorotan di Indonesia, pasalnya dari tahun ke tahun kebakaran selalu terjadi di beberapa wilayah Indonesia. Tercatat pada website resmi BNPB (Badan Nasional Penanggulangan Bencana) bahwa jumlah kasus kebakaran cukup banyak disetiap tahunnya. Selama 3 tahun ini tercatat kasus tertinggi terjadi pada tahun 2018. Pada tahun 2017 tercatat ada 96 kasus, 2018 tercatat ada 527 kasus dan 2019 tercatat ada 150 kasus kebakaran hutan dan lahan[1]. Kebakaran juga kerap terjadi di daerah-daerah padat penduduk yang umumnya terjadi akibat hubungan singkat arus listrik, kebocoran gas LPG atau kelalaian yang dilakukan manusia seperti lupa mematikan kompor, lupa mematikan lilin dan sebab lain yang dapat menimbulkan terjadinya kebakaran. Banyak kasus kebakaran yang terlambat dideteksi dari awal sehingga menimbulkan korban jiwa, kerugian materil, serta kematian flora dan fauna.

Kecerdasan buatan merupakan salah satu bentuk maju nya teknologi, dimana dengan kecerdasan buatan teknologi dapat yang meniru kecerdasan manusia yang dapat mengenali objek-objek seperti api, jenis kelamin dan lain sebagainya. Kecerdasan juga dapat meniru kecerdasan manusia untuk menyelesaikan masalah-masalah dengan cepat seperti contoh pendeteksian api yang dapat menyebabkan terjadinya kebakaran. Saat ini telah dikembangkan sensor-sensor untuk mendeteksi kebakaran seperti sensor yang bekerja untuk mendeteksi api. Dimana sensor bekerja untuk mendeteksi api bergantung pada karakteristik tertentu seperti suhu dan asap. Akan tetapi kemampuan sensor-sensor tersebut memiliki kelemahan seperti sensor api kurang cukup menjangkau area yang luas dan sensor asap yang bergantung pada arah mata angin. Hal tersebutlah yang menyebabkan sensor tidak ideal untuk mendeteksi kebakaran dini [2].

Kamera merupakan suatu alat yang memiliki banyak fungsi. Kamera saat ini banyak digunakan karena kemampuannya yang dapat menangkap gambar sebagaimana fungsi indra penglihatan. Dengan hasil tangkapan kamera ini pernah dilakukan penelitian untuk mendeteksi api. Dimana pada penelitian [3] dan [4] yang menggunakan ruang warna HSV dalam melakukan segmentasi warna api. Pada penelitian [3] kesalahan terutama disebabkan oleh pantulan cahaya kebakaran pada dinding atau permukaan yang mengkilap. Lalu pada penelitian [4], eksperimen menunjukkan bahwa algoritma yang digunakan tidak dapat mendeteksi api dikarenakan selalu salah mengidentifikasi lampu sebagai api. Oleh karena itu proses yang hanya menggunakan proses segmentasi belum optimal untuk mendeteksi api.

Lalu selanjutnya penelitian yang tidak sampai hanya pada proses segmentasi warna saja, yakni melakukan proses ekstraksi fitur terhadap citra. Dimana pada penelitian [5] dilakukan proses ekstraksi fitur menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) dan klasifikasi menggunakan metode *Artificial Neural Network* (ANN) *backpropagation*. Pada penelitian ini didapatkan *error* yang cukup rendah yakni sebesar 7%. Namun peneliti menyarankan agar dilakukannya proses kombinasi metode ekstraksi fitur dengan metode *Local Binary Patern* (LBP).

Oleh sebab itu, pada penelitian kali ini dengan memanfaatkan hasil tangkapan gambar dari kamera akan dibangun sebuah sistem yang dapat mendeteksi kebakaran dengan menggunakan pendekatan pengolahan citra digital menggunakan penggabungan metode ekstraksi fitur *Local Binary Patern* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM) serta klasifikasi *Artificial Neural Network* (ANN) *backpropagation*. Dari penelitian ini, diharapkan hasilnya dapat dimanfaatkan untuk pendeteksi api.

## **1.2 Rumusan Masalah**

Berdasarkan pada latar belakang yang telah diuraikan, perumusan masalah yang didapat pada tugas akhir ini yaitu Bagaimana cara membangun model ANN *backpropagation* terbaik dengan metode ekstraksi fitur LBP dan GLCM yang dapat mendeteksi api?

## **1.3 Batasan Masalah**

Dari permasalahan yang telah diuraikan di atas terdapat batasan-batasan masalah dalam pembangunan aplikasi ini yaitu:

1. Klasifikasi dibagi menjadi dua kelas yaitu terdeteksi dan tidak terdeteksi api
2. Objek yang dideteksi hanya api
3. Data video memiliki *channel* R,G,B dengan panjang antara 10-60 detik
4. Jarak titik api ke kamera pada video tidak ditentukan
5. Formad video yang digunakan .mp4, .avi
6. Resolusi video maksimal 640x640

#### **1.4 Tujuan**

Adapun tujuan yang akan dicapai dari tugas akhir ini yaitu membangun model ANN *backpropagation* terbaik dengan metode ekstraksi fitur gabungan LBP dan GLCM yang dapat mendeteksi api.

#### **1.5 Manfaat**

Manfaat yang diperoleh dari tugas akhir ini adalah:

1. Bagi penyusun
  - a. Menerapkan ilmu yang telah diperoleh selama proses perkuliahan di Teknik Informatika Universitas Mataram.
  - b. Menambah wawasan dalam bidang pengolahan citra digital (pengenalan pola).
2. Bagi pembaca
  - a. Menambah ilmu pengetahuan terutama mengenai teknologi pengenalan pola api dalam pengolahan citra digital.
  - b. Menerapkan hasil dari skripsi ini pada permasalahan yang berhubungan dengan pengenalan pola api.
  - c. Dapat Mengetahui cara mendeteksi kebakaran menggunakan metode ANN.

#### **1.6 Sistematika Penulisan**

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. Bab I Pendahuluan  
Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.
2. Bab II Tinjauan Pustaka dan Dasar Teori

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, meliputi Pengenalan Pola, LBP, GLCM dan ANN.

3. Bab III Metodologi Penelitian

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

4. BAB IV: HASIL DAN PEMBAHASAN

Bab ini menjabarkan hasil yang diperoleh dari pengembangan dan penggunaan sistem pakar diagnosis kerusakan bangunan disertai dengan pembahasan hasil tersebut. Bab ini juga membahas mengenai perhitungan yang digunakan dalam pembuatan sistem

5. BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari sistem yang telah dibangun dan digunakan serta saran yang diberikan sebagai bahan acuan dan evaluasi agar sistem mampu dikembangkan lebih baik lagi.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian dengan penggunaan *Local Binary Pattern* (LBP) sebagai ekstraksinya telah dilakukan oleh beberapa peneliti diantaranya ekstraksi ciri pada telapak tangan dimana prosesnya dimulai dengan proses *pre-processing* yaitu tahap persiapan citra berwarna yang akan dirubah menjadi citra keabuan kemudian dilanjutkan dengan proses *regioning* atau proses pembagian citra menjadi beberapa region, sehingga didapatkan akurasi sebesar 92,31% [6], pengenalan pola huruf hiragana dan katakana pada *smartphone* dengan 460 data uji sehingga didapatkan tingkat akurasi sebesar 81,1% yang merupakan hasil yang cukup besar namun metode ini kurang mampu mengenali perbedaan pola huruf yang kurang signifikan [7], serta deteksi keaslian mata uang rupiah dengan metode k-NN sebagai metode klasifikasi dengan jumlah total 120 data uji dengan masing – masing 30 set data pecahan 50k asli, 30 set data pecahan 50k palsu dan 30 set data pecahan 100k asli dan 30 set data pecahan 100k palsu sehingga didapatkan tingkat akurasi sebesar 95% [8].

Penelitian dengan menggunakan metode *Gray Level Co-occurrence Matrix* (GLCM) sebagai metode ekstraksinya telah dilakukan oleh beberapa peneliti diantaranya klasifikasi mutu pepaya dengan metode ANN *backpropagation* sebagai metode klasifikasi dengan jumlah data yang digunakan sebanyak 192 data citra dengan rincian 156 data latih yang terdiri dari 52 citra pepaya kelas super, 52 citra pepaya kelas A, dan 52 citra pepaya kelas B, dan data pengujian sebanyak 36 data citra pepaya dan didapatkan sebesar 86,11% [9], Klasifikasi jenis daging sapi, kambing dan babi dengan metode *k-Nearest Neighbour* (k-NN) sebagai metode klasifikasi dengan jumlah citra yang digunakan adalah 25 untuk data latih dan 5 citra data uji dari masing-masing jenis daging, sehingga total keseluruhan dataset adalah sebanyak 90 citra dengan tingkat akurasi sebesar 73,3% [10], klasifikasi mutu buah jeruk keprok dengan metode *support vector machine* (SVM) sebagai metode klasifikasi dengan jumlah data sebanyak 100, 60 sebagai data latih dan 40 sebagai data uji dan akurasi sebesar 82,5% [11] dan klasifikasi kayu jati dan mahoni dengan metode jarak Euclidean sebagai metode klasifikasi dengan tingkat akurasi sebesar 82,5% [12].

Penelitian dengan menggunakan penggabungan metode LBP dan GLCM pernah dilakukan oleh beberapa peneliti antara lain pengenalan ekspresi mulut pembelajar untuk mengidentifikasi status ekspresi wajah pembelajar saat menggunakan *e-learning* dengan pengolahan citra digital dengan metode *multiclass SVM* sehingga didapatkan akurasi sebesar 95% [13] dan pengenalan ekspresi wajah pengguna *e-learning* dengan metode *backpropagation neural network* sebagai metode klasifikasi nya sehingga didapatkan tingkat akurasi sebesar 88,89% [14].

Selanjutnya penelitian yang menggunakan metode *Artificial Neural Network (ANN) backpropagation* sebagai metode pengklasifikasian pernah digunakan oleh beberapa peneliti antara lain klasifikasi kardiograf dengan jumlah data sebanyak 36 dan diklasifikasi menjadi 3 kelas yaitu kelas normal, kelas *suspect* dan kelas *pathologic* dengan record sebanyak 2126 sehingga didapatkan akurasi sebesar 99,15% [15], klasifikasi daun tanaman *theobroma cacao* dengan jumlah data sebanyak 90 citra dan didapatkan akurasi sebesar 83% daun rusak, 96% daun rusak sedang dan 86% daun sehat [16],

Penelitian mengenai deteksi api menggunakan ruang warna sebagai segmentasi untuk mendeteksi api telah banyak dilakukan sebelumnya. Penelitian yang dilakukan oleh [3], [4] dan [17] yang menggunakan ruang warna dalam melakukan segmentasi warna api. Penelitian [3] 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 biaya komputasi cukup rendah sehingga bisa dilakukan secara *real-time*. Di penelitian [4], segmentasi pada ruang warna HSV diterapkan pada robot pemadam api dengan hasil akurasi sebesar 80%. Namun eksperimen menunjukkan bahwa algoritma yang digunakan tidak dapat mendeteksi api dikarenakan selalu salah mengidentifikasi lampu sebagai api. Penelitian ini tidak menggunakan *Background Subtraction* untuk deteksi gerakan. Selanjutnya di penelitian [17], dimana pada penelitian ini deteksi api dilakukan berbasis *spatial temporal* dengan metode ekstraksi kontur dan *area movement analysis* menghasilkan akurasi tertinggi yaitu 88%. Namun pada penelitian ini dilihat

dari hasilnya, metode ini tidak dapat mendeteksi api jika api berada di area yang banyak cahaya. Oleh karena itu untuk mendeteksi api proses yang hanya menggunakan segmentasi saja tidaklah optimal.

Penelitian mengenai deteksi api yang prosesnya tidak sampai hanya menggunakan metode segmentasi telah banyak dilakukan. Penelitian yang dilakukan oleh [18] dan [5] menggunakan proses ekstraksi fitur dan klasifikasi setelah dilakukannya proses segmentasi. Kedua peneliti ini menggunakan klasifikasi yang sama yakni ANN *backpropagation* namun dengan metode ekstraksi fitur yang berbeda. Pada peneliti [18] menggunakan metode ruang warna HSI sebagai metode ekstraksi fitur yang menghasilkan akurasi sebesar 96,47%. Lalu oleh [5] menggunakan metode ekstraksi fitur GLCM menghasilkan akurasi sebesar 93%. Namun pada jurnal ini menyarankan untuk dilakukannya kombinasi pada proses ekstraksi fitur menggunakan metode GLCM dan LBP.

Berdasarkan beberapa penelitian di atas, maka dapat dilakukan penelitian dengan penggabungan antara metode LBP dan GLCM sebagai ekstraksi citra api dan metode ANN *backpropagation* sebagai metode pengklasifikasiannya.

## 2.2 Dasar Teori

### 2.2.1 Api

Api dalam Kamus Besar Bahasa Indonesia (KBBI) berarti panas dan cahaya yang berasal dari sesuatu yang terbakar. Api adalah sumber energi yang berpotensi dapat menyebabkan kerusakan baik ekonomis dan ekologis [5]. Kebakaran adalah kejadian yang menimbulkan terjadinya api yang tidak terkendali yang dapat membahayakan jiwa maupun harta benda[19]. Api memiliki aturan warna untuk api gelap dan api terang [17]. Nilai piksel api dapat dilihat pada persamaan 2-1 dan 2-2.

$$\begin{aligned} & 179 < R \\ \text{Piksel Api Gelap} & \{ 170 < G < 159 \\ & B < 119 \end{aligned} \quad (2-1)$$

$$\begin{aligned} & 210 < R \\ \text{Piksel Api Terang} & \{ 123 < G \\ & B < 176 \end{aligned} \quad (2-2)$$

Gambar 2.1 merupakan gambar yang terdapat objek api.



Gambar 2. 1 contoh gambar adanya objek api

### 2.2.2 Kecerdasan Buatan

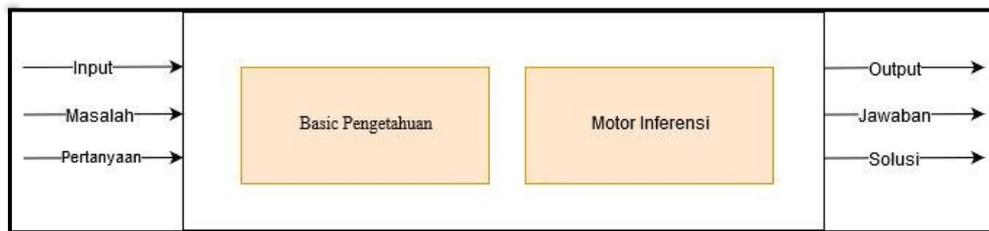
Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia[20].

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[20].

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.

Penerapan Konsep Kecerdasan Buatan pada Komputer dapat dilihat pada Gambar 2.1 berikut :



Gambar 2. 2 Penerapan Konsep Kecerdasan Buatan di Komputer[20]

### 2.2.3 Gray-Level Co-Occurrence Matrix

Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [11].

Koordinat pasangan piksel memiliki jarak  $d$  dan orientasi sudut  $\Theta$ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah sudut yaitu,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ , dan jarak antar piksel sebesar 1 piksel [11].

Penelitian yang dilakukan oleh Haralick yang berjudul “*Textural Features for Image Classification*” mengusulkan fitur tekstur yang mengandung informasi tentang karakteristik tekstur[21]. Pada Tabel 2.1 menjelaskan rumus perhitungan fitur tersebut.

Tabel 2.1 Tabel fitur tekstur GLCM.

No.	Properti	Rumus	Persamaan
1	<i>Angular Second Moment : Energy/Uniformity</i>	$f_1 = \sum_i \sum_j \{p(i, j)\}^2$	(2-3)
2	<i>Contrast</i>	$f_2 = \sum_{n=0}^{Ng-1} n^2 \left\{ \sum_{i=1}^{Ng} \sum_{\substack{j=1 \\  i-j =n}}^{Ng} p(i, j) \right\}$	(2-4)

3	Correlation	$f_3 = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$ <p>Dimana <math>\mu</math> merupakan means dan <math>\sigma</math> merupakan standar deviasi dari p. <math>N_g</math> merupakan jumlah kolom/baris. <math>n</math> jumlah pixel. <math>i</math> merupakan baris ke-<math>i</math>, <math>j</math> merupakan kolom ke-<math>j</math>.</p>	(2-5)
4	Sum of Squares : Variance	$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j)$	(2-6)
5	Inverse Difference Moment : Homogeneity	$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j)$	(2-7)
6	Sum Average	$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$	(2-8)
7	Sum Entropy	$f_7 = -\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$	(2-9)
8	Sum Variance	$f_8 = \sum_{i=2}^{2N_g} (i - \bar{x})^2 p_{(x+y)}(i)$	(2-10)
9	Entropy	$f_9 = -\sum_i \sum_j p(i, j) \log(p(i, j))$	(2-11)
10	Difference Variance	$f_{10} = \text{variance of } p_{x+y}$	(2-12)
11	Difference Entropy	$f_{11} = -\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$	(2-13)
12		$f_{12} = \frac{HXY - HXY1}{\max\{H, X, Y\}}$	(2-14)

	Information Measures of Correlation	$f_{13} = (1 - \exp[-2.0(HXY^2 - HXY)])^{\frac{1}{2}}$ $HXY = -\sum_i \sum_j p(i, j) \log(p(i, j))$ <p>Dimana HX dan HY adalah <i>entropy</i> dari <math>p_x</math> dan <math>p_y</math>, dan</p> $HXY1 = -\sum_i \sum_j p(i, j) \log\{p_x(i)p_y(j)\}$ $HXY2 = -\sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\}$	(2-15)
13	Maximal Correlation Coefficient	$f_{14} = (\text{Second largest eigenvalue of } Q)^{\frac{1}{2}}$ <p>dimana</p> $Q_{(i,j)} = \sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)}$	(2-16)

Pada jurnal internasional dengan judul “*Efficient analysis of satellite image denoising and resolution enhancement for improving classification accuracy*”, terdapat 5 fitur yang paling baik untuk digunakan yaitu *Energy, Contrast, Correlation, Homogeneity* dan *Entropy* [22].

#### 2.2.4 Local Binary Pattern

Local Binary Pattern merupakan metode yang digunakan sebagai ukuran tekstur *grayscale* yang terbukti efektif dan *invariant* terhadap pencahayaan yang berbeda. Metode ini teruji ampuh untuk mendeskripsikan tekstur, karena memiliki daya pembeda yang akurat, serta mempunyai toleransi terhadap perubahan *grayscale* yang *monotatic*. LBP dimanfaatkan untuk deskripsi tekstur dan didukung oleh komposisi pola mikro yang dapat dijelaskan oleh sebuah operator[6].

Operator tersebut bekerja dengan cara memberikan label pada piksel dengan melakukan *thresholding* pada setiap piksel tetangga sebagai nilai tengah dan mengubah hasilnya menjadi nilai 0 atau 1 (biner). Oleh sebab itu Ojala, et al menyebut

bahwa fundamental *pattern* ini sebagai *uniform pattern* karena mengandung dua *bitwise transition* dari 0 ke 1 dan sebaliknya. Apabila piksel bernilai kurang dari nilai tengah (piksel yang diolah) maka akan diberi nilai 0, sedangkan piksel yang memiliki nilai lebih dari nilai tengah maka akan diberi nilai 1 [6].

Secara matematis *thresholding* ditunjukkan oleh Persamaan 3, dan Persamaan 3 untuk menghitung nilai biner dari hasil *thresholding* menjadi angka desimal [6].

$$(x) \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2 - 17)$$

$$(x)_{LBP} = \sum_{p=0}^{p=1} S(gp - gc)2^p \quad (2 - 18)$$

Dimana :

P : banyaknya piksel tetangga

R : nilai jarak/radius

gc : nilai dari piksel x dan y

gp : nilai piksel tetangga

xc , yc : koordinat pusat

Untuk mengikut sertakan *micro-pattern* dari telapak tangan, ekstraksi LBP dilakukan dengan cara membagi citra sama rata menjadi *regionregion* R0,R1,R2,.....Rm. Pada setiap *region* dilakukan proses LBP yang kemudian menghasilkan histogram. Hasil histogram dari tiap *region* kemudian di-*concat* menjadi histogram ciri [6].

Histogram ciri yang terbentuk menggambarkan tekstur lokal dan bentuk global dari telapak tangan. Beberapa parameter yang dioptimalkan untuk menghasilkan ekstraksi ciri yang lebih baik. Parameter tersebut adalah operator dari LBP dan jumlah pembagian citra (*region*)[6].

### 2.2.5 Konsep Dasar Jaringan Saraf Tiruan

Setiap pola-pola informasi *input* dan *output* yang diberikan ke dalam JST diproses dalam neuron. Neuron-neuron tersebut terkumpul di dalam lapisan-lapisan yang disebut neuron *layers*. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3[23], yaitu :

1. Lapisan *input*, unit-unit di dalam lapisan *input* disebut unit-unit *input*. Unit-unit *input* tersebut menerima pola data dari luar yang menggambarkan suatu permasalahan.

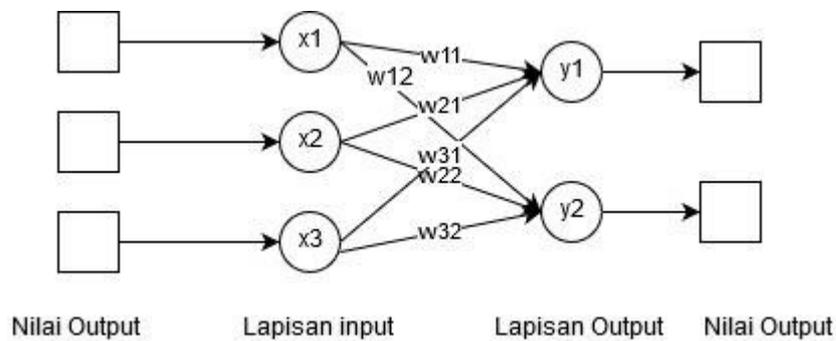
2. Lapisan tersembunyi, unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di mana *output*-nya tidak dapat secara langsung diamati.
3. Lapisan *Output*, unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

### 2.2.6 Arsitektur Jaringan Syaraf Tiruan

JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Hermawan A., 2006)[23]:

#### 1. Jaringan Lapisan Tunggal (*Single Layer Network*)

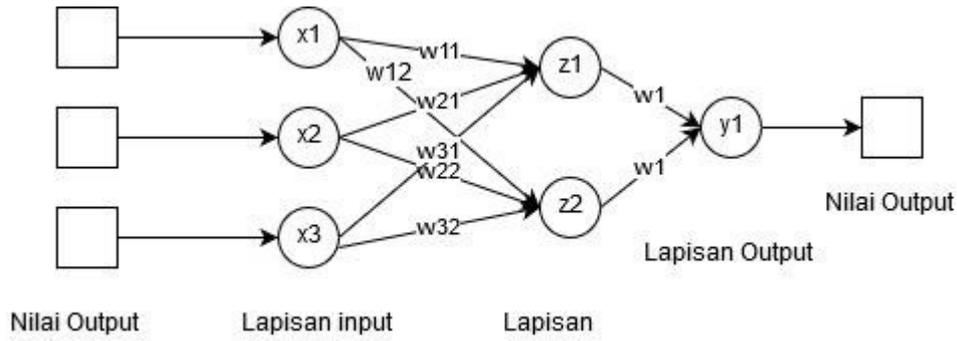
Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setiap neuron yang terdapat di dalam lapisan *input* selalu terhubung dengan setiap neuron yang terdapat pada lapisan *output*. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.



Gambar 2. 3 Arsitektur Lapisan Tunggal

#### 2. Jaringan Banyak Lapisan (*Multilayer net*)

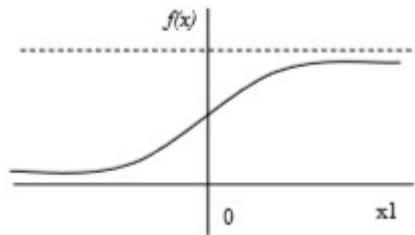
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis lapisan yakni lapisan *input*, lapisan *output*, dan lapisan tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama.



Gambar 2. 4 Arsitektur Lapisan Multilayer

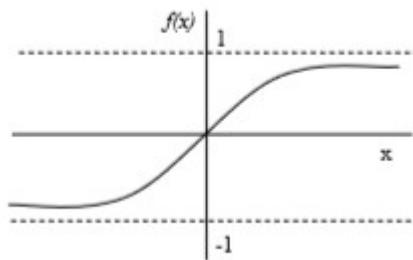
### 2.2.7 Fungsi Aktivasi *Backpropagation*

Pada *Artificial Neural Network backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi *sigmoid* biner yang memiliki range (0,1). Diberikan  $F(x) = \frac{1}{1+e^{-x}}$  dengan turunan  $f'(x) = f(x)(1 - f(x))$ . Grafik fungsinya tampak pada Gambar 5.



Gambar 2. 5 Fungsi Aktivasi Sigmoid Biner

Fungsi lain yang sering dipakai adalah fungsi *sigmoid bipolar* yang bentuk fungsinya mirip dengan fungsi *sigmoid* biner, tapi dengan range (-1,1). Diberikan  $F(x) = \frac{2}{1+e^{-x}} - 1$  dengan turunan  $f'(x) = \frac{(1+f(x))(1-f(x))}{2}$ . Grafik fungsinya tampak pada Gambar 2.7.



Gambar 2. 6 Fungsi Aktivasi Sigmoid Bipolar

Fungsi *sigmoid* memiliki nilai maksimum = 1. Maka untuk pola yang targetnya lebih dari 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi *sigmoid* yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi *sigmoid* hanya pada lapisan yang bukan lapisan keluaran. Pada lapisan keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas[23] :  $f(x) = x$

### 2.2.8 Pelatihan Standar *Backpropagation*

Pelatihan *backpropagation* meliputi 3 Tahap yaitu tahap maju, propagasi mundur, dan perubahan bobot. Algoritma pelatihan untuk jaringan dengan satu lapisan tersembunyi (dengan fungsi aktivasi *sigmoid* biner) adalah sebagai berikut : Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-8

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

#### Tahap I : Propagasi Maju

Langkah 3 :Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi  $z_j (j = 1, 2, \dots, p)$

$$z_j = v_{j0} + \sum_{i=0}^n netx_i v_{ji} \quad (2-19)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (2-20)$$

Langkah 5 : Hitung semua keluaran jaringan di unit  $y_k (y = 1, 2, \dots, m)$

$$y_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (2-21)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (2-22)$$

## Tahap II : Propagasi mundur

Langkah 6 : Hitung faktor unit keluaran berdasarkan *error* di setiap unit keluaran

$y_k (k = 1, 2, \dots, m)$

$$\delta_k = (t_k - y_k) f'(y_{net\ k}) = (t_k - y_k) y_k (1 - y_k) \quad (2-23)$$

$\delta_k$  merupakan unit eror yang akan dipakai dalam perubahan bobot lapisan di bawahnya (langkah 7) Hitung suku perubahan bobot  $w_{kj}$  (yang akan dipakai nanti untuk merubah bobot  $w_{kj}$ ) dengan laju percepatan  $\alpha$

$$\Delta w_{kj} = \alpha \delta_k z_j \quad ; \quad k = 1, 2, \dots, m \quad ; \quad j = 0, 1, 2, \dots, p \quad (2-24)$$

Langkah 7 : Hitung faktor  $\delta$  unit tersembunyi berdasarkan eror di setiap unit tersembunyi  $z_j (j = 1, 2, \dots, p)$

$$\delta_j = \sum_{k=1}^m \delta_k w_{kj} \quad (2-25)$$

Faktor  $\delta$  unit tersembunyi:

$$\delta_j = \delta_{net\ j} f'(\delta_{net\ j}) = \delta_{net\ j} z_j (1 - z_j) \quad (2-26)$$

Hitung suku perubahan bobot  $v_{ji}$  ( yang akan dipakai nanti untuk merubah bobot  $v_{ji}$ )

$$\Delta v_{ji} = \alpha \delta_j x_i \quad ; \quad j = 1, 2, \dots, p \quad ; \quad i = 0, 1, 2, \dots, m \quad (2-27)$$

## Tahap III : Perubahan bobot

Langkah 8 : Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran :

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (k = 1, 2, \dots, m \quad ; \quad j = 0, 1, 2, \dots, p) \quad (2-28)$$

Perubahan bobot garis yang menuju ke unit tersembunyi :

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (j = 1, 2, \dots, p \quad ; \quad i = 0, 1, 2, \dots, m) \quad (2-29)$$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju ( langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan *sigmoid* biner, maka langkah 4 dan 5 harus diselesaikan. Demikian juga turunannya pada langkah 6 dan 7

Langkah 9: Uji kondisi berhenti (akhir iterasi)[23].

## BAB III METODE PENELITIAN

### 3.1 Bahan dan Alat Penelitian

Bahan-bahan yang digunakan pada penelitian ini adalah citra api dan non api yang disajikan dalam bentuk video sebanyak 100 video yang diambil menggunakan kamera atau mengambil dari internet. Terdapat 2 pengkategorian video yaitu api dan non api. Tabel 3. 1 menunjukkan contoh *frame* dari video dengan kategori api dan non api.

Tabel 3. 1 Contoh data frame video

Api	Non Api
	

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™ i3
- b. Memori RAM 4 GB

2. Perangkat lunak

Perangkat lunak yang digunakan dalam penelitian ini, yaitu:

- a. Sistem operasi *Linux* dan *Windows*

Sistem operasi *linux* digunakan untuk pembuatan sistem pendeteksi api sedangkan *windows* digunakan untuk pembuatan laporan dan pembuatan data.

- b. *Jupyterlab*

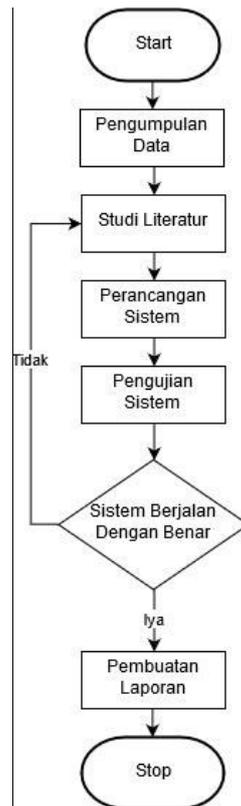
*Jupyterlab* adalah sebuah *software* yang digunakan untuk membuat sistem pendeteksi api

c. Filmora

Filmora adalah salah satu *software* yang digunakan untuk proses pengeditan *video* yang melingkupi pemotongan *video* agar durasinya menjadi 10-60 detik

### 3.2 Rencana Penelitian

Pada tahap proses penelitian dilakukan beberapa tahap secara sistematis, dimulai dari tahap pengumpulan data hingga tahap penarikan kesimpulan dari sistem yang telah dibuat dan diuji. Tahap proses penelitian ini dapat dilihat pada diagram alir Gambar 3. 1



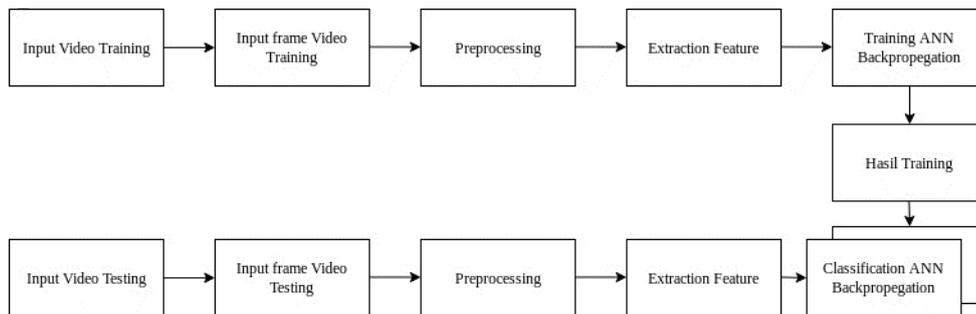
Gambar 3. 1 Diagram alir pembuatan sistem

Langkah pertama dalam pembuatan sistem ini adalah pengumpulan citra api dan non api dalam bentuk video. Citra tersebut di ambil dari berbagai sumber seperti internet ataupun citra yang dibuat sendiri. Langkah kedua yakni studi literatur dilakukan dengan cara mempelajari buku-buku, jurnal-jurnal penelitian serta sumber lain yang berhubungan dengan permasalahan yang sama. Adapun materi yang dipelajari dalam studi literatur yakni materi yang berkaitan dengan ekstraksi fitur menggunakan metode LBP dan GLCM, serta metode klasifikasi menggunakan metode ANN *backpropagation* serta materi yang lain yang berkaitan dengan penelitian yang

dilakukan. Selanjutnya adalah tahap pembangunan sistem sesuai yang telah direncanakan. Tahap pengujian merupakan tahap untuk menguji apakah sistem berjalan sesuai dengan tujuan, jika tidak sesuai maka kembali ke tahap studi literatur jika sudah sesuai maka akan dilanjutkan ke proses pembuatan laporan.

### 3.3 Rancangan Sistem Deteksi Api

Sistem ini secara garis besar terdapat tiga proses utama yaitu *training* (pelatihan), *testing* (pengujian) dan klasifikasi, seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Proses training (pelatihan), testing (pengujian) dan klasifikasi

#### 2.2.9 Tahap Preprocessing

Tahap *preprocessing* yang terdiri atas proses *segmentation* dan *cropping*. Setiap citra latih dan citra uji akan melewati tahapan awal tersebut untuk menghilangkan dan mengurangi *error* yang dapat berdampak pada akurasi akhir pada saat masuk ke tahap klasifikasi.

##### 1. *Segmentation*

Pada tahapan ini, akan dilakukan proses segmentasi terhadap citra dengan tujuan untuk mengetahui posisi dari api. Dimana untuk mengetahui posisi api digunakan aturan nilai piksel api, yaitu aturan untuk api gelap dan api terang [17]. Nilai piksel api dapat dilihat pada persamaan (3-1) dan (3-2).

$$\begin{aligned}
 & 179 < R \\
 \text{Piksel Api Gelap} & \{ 10 < G < 159 \\
 & B < 119
 \end{aligned} \tag{3-1}$$

$$\begin{aligned}
 & 210 < R \\
 \text{Piksel Api Terang} & \{ 123 < G \\
 & B < 176
 \end{aligned} \tag{3-2}$$

Pada Gambar 3.3 dan Gambar 3.4 merupakan tampilan dari tahap segmentasi



Gambar 3. 3 Citra awal



Gambar 3. 4 Hasil segmentasi

## 2. *Cropping*

Pada tahap ini, dilakukan proses *cropping* terhadap citra setelah diketahui posisi api dengan tujuan untuk pengolahan citra hanya difokuskan terhadap api saja. Lalu untuk Penentuan bagian yang di *crop* digunakan metode *connected components* untuk mendeteksi objek-objek yang sudah di segmentasi dalam 1 frame kemudian dipilih bagian dengan lokasi api terbanyak. Pada Gambar 3. 5 merupakan tampilan dari tahap *cropping*.



Gambar 3. 5 Proses *Cropping*

### 2.2.10 Extraction Feature

Tahap selanjutnya yaitu *extraction feature* yang merupakan bagian dari teknik pengenalan pola (*pattern recognition*) yang bertujuan untuk mendapatkan ciri dari citra. Pada penelitian ini digunakan penggabungan Antara metode ekstraksi *Local Binary Patern* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM). Proses ekstraksi dibagi menjadi 2 tahap Antara lain

#### 1. *Local Binary Patern*

*Local Binary Patern* adalah LBP adalah metode analisis tekstur yang menggunakan model statistika dan struktur. Operator LBP menggunakan perbandingan nilai keabuan dari piksel-piksel ketetanggaan. Operator dasar LBP berukuran 3 x 3 menggunakan 8 piksel ketetanggaan dari sebuah piksel tengah[7]. Berikut pada Gambar 3.6 merupakan contoh komputasi LBP pada citra 4x4 pixel

5	2	4	2
2	4	6	10
1	7	8	11
2	3	6	5

Gambar 3. 6 Citra 4x4

Berikut tahap-tahap Komputasi LBP:

- Lakukan *threshold* berpusat pada titik tengah. Pixel yang memiliki nilai sama atau lebih dibandingkan dengan titik tengah diberi nilai 1 selain itu diberi nilai 0. Sebagai contoh akan dilakukannya *threshold* dengan posisi titik pusat berada pada pixel (1,1). Kemudian titik pusat akan dibandingkan dengan pixel-pixel tetangga dari titik pusat. Gambar 3. 7 menyajikan hasil dari proses *threshold*

A =	<table border="1"> <tbody> <tr> <td>5</td> <td>2</td> <td>4</td> <td>2</td> </tr> <tr> <td>2</td> <td style="background-color: #f4a460;">4</td> <td>6</td> <td>10</td> </tr> <tr> <td>1</td> <td>7</td> <td>8</td> <td>11</td> </tr> <tr> <td>2</td> <td>3</td> <td>6</td> <td>5</td> </tr> </tbody> </table>	5	2	4	2	2	4	6	10	1	7	8	11	2	3	6	5
5	2	4	2														
2	4	6	10														
1	7	8	11														
2	3	6	5														

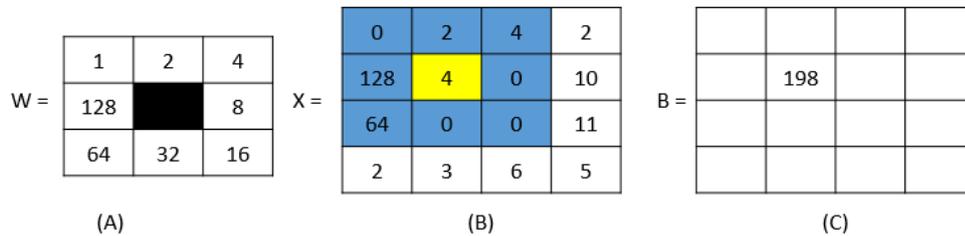
X =	<table border="1"> <tbody> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td style="background-color: #ffff00;">4</td> <td>0</td> <td>10</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>11</td> </tr> <tr> <td>2</td> <td>3</td> <td>6</td> <td>5</td> </tr> </tbody> </table>	0	1	1	2	1	4	0	10	1	0	0	11	2	3	6	5
0	1	1	2														
1	4	0	10														
1	0	0	11														
2	3	6	5														

(A)
(B)

Gambar 3. 7 (a) penentuan titik pusat index(1,1) (b) hasil *threshold*

- Ganti nilai 1 dari hasil *threshold* yang ada pada setiap *index* sesuai dengan posisi *weights*. Lalu jumlahkan seluruh nilai pixel tetangga dari pixel pusat

kemudian hasilnya ditempatkan pada matriks B dengan posisi pixel sama dengan posisi pixel titik pusat Gambar 3.8 menunjukkan proses penempatan hasil akhir dari proses komputasi pada posisi pixel (1,1)



Gambar 3. 8 (a) weights (b)hasil perubahan nilai sesuai weights(b) hasil akhir komputasi LBP pixel(1,1)

- c. Lakukan proses tersebut disetiap pixel citra sampai semua pixel pada matriks B terpenuhi.
- d. Sehingga hasil akhirnya dapat dilihat pada Gambar 3.9.

255	71	207	31
255	198	135	223
193	247	243	255
243	241	248	124

Gambar 3. 9 Hasil akhir perhitungan

## 2. Gray Level Co-occurrence Matrix

*Gray Level Co-occurrence Matrix* adalah matriks yang menggambarkan frekuensi munculnya pasangan dua *pixel* dengan intensitas tertentu dalam jarak (d) dan orientasi arah dengan sudut ( $\theta$ ) tertentu dalam citra [24]. Proses ekstraksi fitur dengan metode GLCM dibagi menjadi dua tahap, yaitu pembentukan matriks GLCM dan perhitungan nilai fitur GLCM.

- a. Pembentukan matriks GLCM

Misalkan terdapat citra *grayscale* dengan matriks seperti pada Gambar 3.10.

A=	0	0	1
	0	0	1
	2	2	2

Gambar 3. 10 Contoh matriks citra *grayscale*

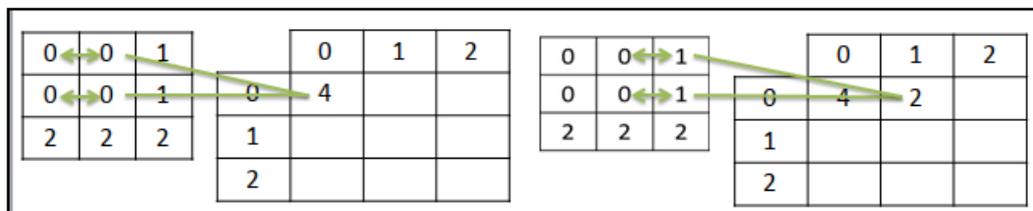
Dari matriks di atas dibentuk matriks GLCM dengan orientasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ . Adapun langkah-langkah pembentukan matriks GLCM sebagai berikut:

- 1) Langkah pertama yaitu membuat matriks dengan ordo  $n \times n$  dimana  $n$  merupakan nilai elemen terbesar dari matriks *grayscale* A. pada matriks A nilai elemen terbesar adalah 2 sehingga ukuran matriks yang akan dibuat adalah  $2 \times 2$  dengan nilai indeks dimulai dari 0 seperti yang terlihat pada Gambar 3.11 berikut.

$$B = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & & & \\ \hline 1 & & & \\ \hline 2 & & & \\ \hline \end{array}$$

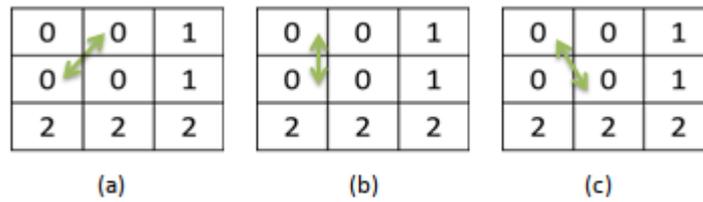
Gambar 3. 11 Matriks GLCM  $3 \times 3$

- 2) Penentuan nilai matriks B dimulai dari elemen (0,0). Dimana untuk mengisi setiap elemen dari matriks B, dilakukan iterasi terhadap 2 blok pada matriks A untuk mengecek nilai elemennya. Sebagai contoh, membuat matriks GLCM dengan sudut  $0^\circ$ , maka arah iterasi yang dilakukan yaitu horizontal.
- 3) Iterasi pertama dilakukan untuk mengecek jumlah matriks ketetanggaan yang bernilai (0,0). Iterasi dimulai dari elemen (0,0) dan (0,1) pada matriks A. Pengecekan berlaku dua arah, artinya elemen (0,0) dan (0,1) dicek sebanyak dua kali dengan arah berlawanan. Jumlah matriks ketetanggaan dengan nilai (0,0) ditulis pada matriks B elemen (0,0). Selanjutnya geser satu blok ke kanan untuk mengecek elemen matriks (0,1) dan (0,2). Lakukan pengecekan hingga elemen (0,2) dan (0,3) kemudian pindah ke baris selanjutnya. Gambar 3. 12 menunjukkan proses pembentukan matriks GLCM pada sudut  $0^\circ$ .



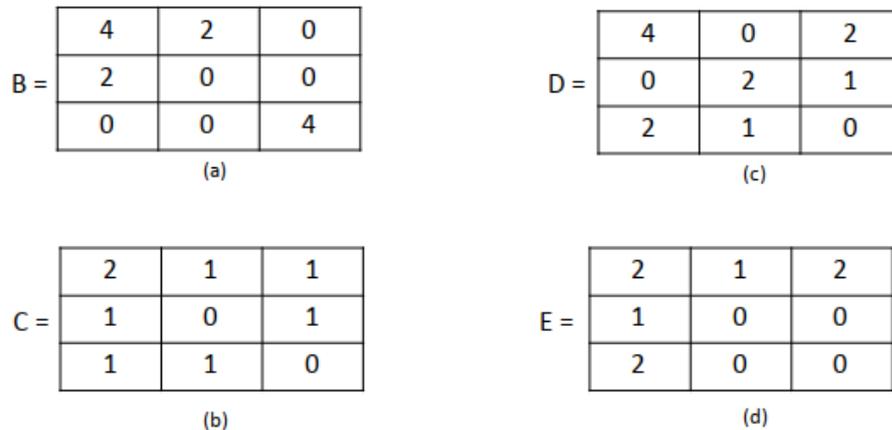
Gambar 3. 12 Pembentukan matriks GLCM sudut  $0^\circ$

- 4) Iterasi selanjutnya adalah mengecek jumlah matriks ketetanggaan yang bernilai (0,1) untuk di isi di matriks B di elemen (0,1), begitu seterusnya sampai elemen terakhir pada matriks B.
- 5) Hal yang sama dilakukan untuk membuat matriks GLCM dengan sudut  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$  namun dengan arah yang berbeda. Matriks GLCM sudut  $45^\circ$  dibuat dengan arah diagonal-kanan (*right-diagonal*), sudut  $90^\circ$  dengan arah vertikal, sedangkan sudut  $135^\circ$  dibuat dengan arah diagonal-kiri (*left-diagonal*) seperti yang terlihat pada Gambar 3.13.



Gambar 3. 13 Arah matriks GLCM a. Sudut  $45^\circ$ . (b) Sudut  $90^\circ$ . (c) Sudut  $135^\circ$

- 6) Didapatkan empat buah matriks GLCM seperti pada Gambar 3.14.



Gambar 3. 14 Hasil pembentukan matriks GLCM (a)  $\theta = 0^\circ$   $d = 1$ . (b)  $\theta = 45^\circ$   $d = 1$ . (c)  $\theta = 90^\circ$   $d = 1$ . (d)  $\theta = 135^\circ$   $d = 1$

- 7) Matriks GLCM kemudian dinormalisasi agar jumlah seluruh elemennya sama dengan satu. Gambar 3.15 merupakan hasil normalisasi terhadap matriks B.

$$F = \begin{bmatrix} 0.333 & 0.1666 & 0.0 \\ 1.666 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.333 \end{bmatrix}$$

Gambar 3. 15 Matriks GLCM yang telah dinormalisasi

Proses normalisasi menghasilkan 4 matriks GLCM normal yang kemudian akan digunakan pada tahap selanjutnya.

b. Perhitungan nilai fitur GLCM

Matriks GLCM yang telah dihasilkan pada tahap sebelumnya digunakan untuk menghitung 5 fitur GLCM yang terdiri atas ASM (*Energy*), *Contrast*, IDM (*Homogeneity*), *Entropy* dan *Correlation*.

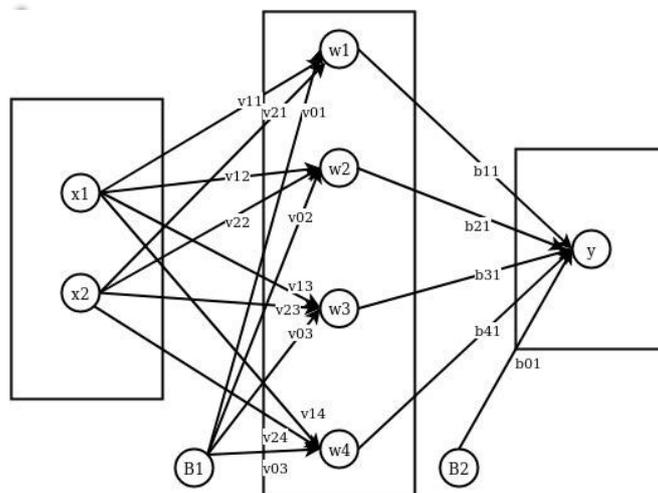
**2.2.11 Klasifikasi**

Proses klasifikasi yang dilakukan dalam penelitian ini adalah menggunakan metode ANN *backpropagation*. Berikut ini diberikan contoh *data training* menggunakan metode *backpropagation* pada pendeteksian api dengan data pelatihan *dummy* yang tertera pada Tabel 3.2. Data *dummy* tersebut merupakan data pelatihan yang terdiri dari 4 data yang terbagi ke dalam 2 kelas berdasarkan target ada atau tidak adanya api, yaitu 1 (ada) dan 0 (tidak ada).

Tabel 3. 2 Data *dummy*

No.	Input 1	Input 2	Target
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

Dalam pelatihan data ini, arsitektur jaringan syaraf tiruan *backpropagation* yang digunakan dapat dilihat pada Gambar 3.16.



. Gambar 3. 16 Arsitektur *backpropagation*

Arsitektur *backpropagation* yang digunakan dalam contoh perhitungan ini adalah *single layer* yang terdiri dari 1 *input*, 1 *hidden layer* dan 1 *output layer*. Di dalam *input layer* terdapat 2 buah *neuron*, 4 buah *neuron* dalam *hidden layer* dan 1 buah *neuron* pada layer *output*.

Proses pelatihan data dengan menggunakan metode *artificial neural network backpropagation* ini menggunakan beberapa ketentuan, berikut merupakan ketentuan-ketentuan untuk contoh pelatihan menggunakan metode ini.

1. Batas *error* = 0.0001
2. Batas *epoch* = 2000
3. *Learning rate*=0.01
4. Menggunakan fungsi aktivasi *sigmoid biner*

Alogaritma metode *backpropagation* terdiri dari 10 langkah termasuk dalam tahap inisialisasi yang terbagi menjadi 3 fase.

Langkah 0 : Inisialisai bobot awal dari *input* awal ke *hidden layer* pertama yang disajikan dalam Tabel 3.3 berikut.

Tabel 3.3 Bias dan bobot awal dari *input* layer ke *hidden layer* pertama.

Dari- Ke-	Bias (B1)	X1	X2
W1	0.28183	0.1377	0.0045
W2	0.7558	0.1031	-0.0380
W3	0.61836	-0.0317	0.1135
W4	0.2505	-0.0964	-0.0786

Bias dan bobot awal dari *input layer* ke *hidden layer* pertama dinyatakan dengan  $V_{01}$  sampai  $V_{24}$ , dapat dilihat pada Tabel 3.3. Selanjutnya untuk bias dan bobot awal dari *hidden layer* pertama ke *output layer* kedua dinyatakan dengan  $b_{01}$  sampai  $b_{41}$  yang disajikan dalam Table 3.4:

Tabel 3.4 Bias dan bobot awal dari *hidden layer* pertama ke *output layer* kedua

Dari- Ke-	Bias (B2)	W1	W2	W3	W4
Y	0.9097	-0.0936	0.3335	1.6324	0.0187

Langkah 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai 9.

Kondisi penghentian terpenuhi jika *error* < 0.0001 atau *epoch* > 2000.

Langkah 2: Untuk setiap data pelatihan, lakukan langkah 3 sampai 8

### Fase I: Feedforward

Langkah 3: Tiap unit masukan ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan meneruskannya ke unit selanjutnya, yaitu lapisan tersembunyi. Yang digunakan pada contoh ini adalah data pertama dengan  $X_1 = 1, X_2 = 0$ .

Langkah 4 : Hitung semua keluaran pada lapisan tersembunyi ( $Z_j, j = 1, 2, \dots, p$ ) menggunakan Persamaan (2-3)

$$\begin{aligned} W_{\text{net}1} &= 0.28183 + (1 * 0.1377) + (0 * 0.0045) \\ &= 0.41953 \end{aligned}$$

$$\begin{aligned} W_{\text{net}2} &= 0.7558 + (1 * 0.1031) + (0 * -0.0380) \\ &= 0.8589 \end{aligned}$$

$$\begin{aligned} W_{\text{net}3} &= 0.61836 + (1 * -0.0317) + (0 * 0.1135) \\ &= 0.58666 \end{aligned}$$

$$\begin{aligned} W_{\text{net}4} &= 0.2505 + (1 * -0.0964) + (0 * -0.0786) \\ &= 0.012542 \end{aligned}$$

Gunakan fungsi aktivasi untuk menghitung sinyal keluaran tiap *neuron hidden layer* pertama. Fungsi aktivasi yang digunakan adalah *sigmoid biner* dan menggunakan Persamaan (2-1).

$$W_1 = \frac{1}{1+e^{-0.41953}} - 1 = 0.60333$$

$$W_2 = \frac{1}{1+e^{-0.8589}} - 1 = 0.7024$$

$$W_3 = \frac{1}{1+e^{-0.58666}} - 1 = 0.6425$$

$$W_4 = \frac{1}{1+e^{-0.012542}} - 1 = 0.5031$$

Sinyal  $W_1, W_2,$  dan  $W_3$  akan diteruskan ke *output layer*.

Langkah 5 : Hitung semua keluaran jaringan di lapisan *output* ( $Y_k, k = 1, 2, \dots, m$ ) menggunakan Persamaan (2-5)

$$\begin{aligned} Y_{\text{net}1} &= 0.9097 + (0.60333 * -0.0936) + (0.7024 * 0.3335) + (0.7024 * 1.6324) + \\ &\quad (0.5031 * 0.0187) \\ &= 2.24348725 \end{aligned}$$

Gunakan fungsi aktivasi *sigmoid biner* untuk menghitung sinyal *output*-nya:

$$Y_1 = \frac{1}{1+e^{-0.16276}} - 1 = 0.904087276$$

## Fase II: *Backpropagation*

Langkah 6: Hitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran ( $y_k, k = 1, 2, \dots, m$ ) menggunakan Persamaan (2-7).

$$\begin{aligned}\delta_1 &= (t_1 - y_1) f'(y_{net1}) \\ &= (1 - 0.904087276) f'(2.24348725) \\ &= 0.008316925\end{aligned}$$

$\delta$  merupakan unit kesalahan yang akan digunakan dalam perubahan bobot *layer* di bawahnya (langkah 7). Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki  $W_{jk}$ ) dengan *learning rate* ( $\alpha$ ), menggunakan Persamaan (2-8)

$$\begin{aligned}\Delta b_{11} &= 0.01 * 0.008316925 * 0.60333 = 0.000050179 \\ \Delta b_{21} &= 0.01 * 0.008316925 * 0.7024 = 0.000058418 \\ \Delta b_{31} &= 0.01 * 0.008316925 * 0.6425 = 0.000053436 \\ \Delta b_{41} &= 0.01 * 0.008316925 * 0.5031 = 0.000041842\end{aligned}$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{0k}$ ) menggunakan Persamaan (2-9)

$$\Delta b_{01} = 0.01 * 0.008316925 = 0.00008316925$$

Langkah 7: Hitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ( $Z_j, j = 1, 2, \dots, p$ ) menggunakan Persamaan (2-21)

$$\begin{aligned}\delta_{net1} &= 0.008316925 * -0.0936 = -0.000778 \\ \delta_{net2} &= 0.008316925 * 0.3335 = 0.00277369 \\ \delta_{net4} &= 0.008316925 * 1.6324 = 0.01357654837 \\ \delta_{net4} &= 0.008316925 * 0.0187 = 0.000155526\end{aligned}$$

Faktor  $\delta$  unit tersembunyi *hidden layer* pertama dihitung menggunakan Persamaan (2-22):

$$\begin{aligned}\delta_1 &= \delta_{net7} f'(Z_{net7}) \\ &= -0.000778 * f'(0.41953) \\ &= -0.000186667 \\ \delta_2 &= 0.00277369 * f'(0.8589) \\ &= 0.00057976161 \\ \delta_3 &= 0.01357654837 * f'(0.58666) \\ &= 0.003118067325\end{aligned}$$

$$\delta_4 = 0.000155526 * f' (0.012542)$$

$$= 0.00003887997$$

Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ) dari *input layer* ke *hidden layer* pertama menggunakan Persamaan (2-23) dengan  $x_i = Z_i$

$$\Delta V_{11} = 0.01 * -0.000186667 * 1 = -0.00000186667$$

$$\Delta V_{12} = 0.01 * 0.00057976161 * 0 = 0$$

$$\Delta V_{13} = 0.01 * 0.003118067325 * 1 = 0.00003118067325$$

$$\Delta V_{14} = 0.01 * 0.00003887997 * 0 = 0$$

$$\Delta V_{21} = 0.01 * -0.000186667 * 1 = -0.00000186667$$

$$\Delta V_{22} = 0.01 * 0.00057976161 * 0 = 0$$

$$\Delta V_{23} = 0.01 * 0.003118067325 * 1 = 0.00003118067325$$

$$\Delta V_{24} = 0.01 * 0.00003887997 * 0 = 0$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $V_{0j}$ ) menggunakan Persamaan (2-23)

$$\Delta V_{01} = 0.01 * -0.000186667 = -0.00000186667$$

$$\Delta V_{02} = 0.01 * 0.00057976161 = 0.0000057976161$$

$$\Delta V_{03} = 0.01 * 0.003118067325 = 0.00003118067325$$

$$\Delta V_{04} = 0.01 * 0.00003887997 = 0.0000003887997$$

### Fase III: Perubahan Bobot

Langkah 8: Tiap-tiap unit *output* ( $Y_k$ ,  $k = 1, 2, \dots, m$ ) memperbaiki bias dan bobotnya ( $j = 0, 1, 2, \dots, p$ ) menggunakan Persamaan (2-14)

$$W_{jk} (\text{baru}) = W_{jk} (\text{lama}) + \Delta W_{jk}$$

$$b_{01} (\text{baru}) = 0.9097 + 0.00008316925 = 0.90978316925$$

$$b_{11} (\text{baru}) = -0.0936 + 0.000050179 = -0.093549821$$

$$b_{21} (\text{baru}) = 0.3335 + 0.000058418 = 0.333558418$$

$$b_{31} (\text{baru}) = 1.6324 + 0.000053436 = 1.632453436$$

$$b_{41} (\text{baru}) = 0.0187 + 0.000041842 = 0.018741842$$

Tiap-tiap unit tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) memperbaiki bias dan bobotnya ( $j = 0, 1, 2, 3, \dots, n$ ) menggunakan Persamaan (2-15)

$$V_{ij} (\text{baru}) = V_{ij} (\text{lama}) + \Delta V_{ij}$$

$$V_{01} (\text{baru}) = 0.28183 + (-0.00000186667) = 0.281828134$$

$$V_{11} (\text{baru}) = 0.1377 + -0.000001866 = 0.0040000207$$

$$V_{12} (\text{baru}) = 0.0045 + 0 = 0.0045$$

Demikian juga untuk  $V_{13} (\text{baru})$  sampai  $V_{24} (\text{baru})$  dihitung dengan cara yang sama untuk memperoleh semua  $V_{ij} (\text{baru})$

Langkah 9: Kondisi pelatihan berhenti, jika  $error \leq 0.0001$  atau jumlah  $epoch$  mencapai 100.  $Error$  dihitung dengan *Mean Square Error* menggunakan persamaan:

$$Error = 0.5 * \{(t_1 - y_1)^2 + (t_2 - y_2)^2 + \dots + (t_m - y_m)^2\} \quad (3-1)$$

1)

Dimana  $m$  adalah jumlah data yang dilatih.

*Software Jupyterlab* digunakan untuk pembuatan jaringan *backpropagation* yang sesuai dengan inisialisasi awal dan parameter dari pelatihan manual yang telah dilakukan. Sehingga diperoleh bias dan bobot akhir yang dapat dilihat pada Tabel 3.3, Tabel 3.5.

Tabel 3. 3 Bias dan bobot akhir dari *input layer* ke *hidden layer* pertama

Dari-Ke-	Bias (B1)	X1	X2
W1	0.41622322099	-0.18717812	-0.115188427727
W2	2.37629377222	-1.656265855	-1.4001292658
W3	0.346785132537	0.959095285	-0.6163883398
W4	0.412857015836	-0.20146158	-0.122667397

Tabel 3. 4 Bias dan bobot akhir dari *hidden layer* pertama ke *output layer*

Dari-Ke-	Bias (B2)	W1	W2	W3	W4
Y	1.50274692507	0.557285153	-2.184492	0.67748887558	0.54299331

Berdasarkan jaringan *backpropagation* yang telah dibuat, 9 data pelatihan *dummy* yang telah disiapkan kemudian dimasukkan ke dalam jaringan sehingga diperoleh hasil yang dapat dilihat pada Tabel 3.5. *Threshold* pada *output* data latih dilakukan dengan membulatkan *output* ke bilangan bulat (*integer*) terdekat, dengan kata lain menggunakan  $\theta = 0.5$ .

Tabel 3. 5 *Output* data latih

Data ke-	<i>Output</i>	Hasil <i>threshold</i>	Target
1	0.023894362639337377	0	0
2	-0.000870907443	0	0
3	0.9814555682475549	1	1
4	0.013271482008336536	0	0

Untuk mendapatkan tingkat akurasi dari hasil pelatihan, jumlah hasil *threshold* yang sesuai target dibagi dengan jumlah data. Tingkat akurasi =  $(4/4) * 100 \% = 100 \%$ .

### 3.4 Teknik Pengujian Sistem

Tahapan pengujian sistem merupakan tahapan yang ditujukan sebagai bahan evaluasi sistem untuk mengetahui sistem yang dibuat telah sesuai dengan perancangan ataupun sistem diharapkan. Dalam melakukan pengujian pada sistem ada beberapa metode pengujian sistem yang dapat digunakan.

#### 3.4.1 Skenario Pengujian Sistem

Pada sistem deteksi api menggunakan metode *backpropagation* ini, dilakukan pengujian terhadap pengaruh jumlah sampling terhadap unjuk kerja dan waktu komputasinya. Dimana rasio data *training* dan *testing* adalah 70:30, 80:20, dan 90:10. Setiap rasio data *training* memiliki ciri citra yang telah disimpan di dalam *database* masing-masing yang digunakan sebagai *training* untuk mendeteksi api dan non api. Sebagai tolak ukur penentuan target pada setiap rasio data video *testing*, dilakukan dengan mengambil ukuran sampel pada video dengan ukuran sampel ditentukan dengan cara perhitungan menggunakan metode sampling yakni metode solvin. Berikut cara menghitungnya

$$n = \frac{N}{1 + Ne^2}$$

Keterangan :

n = ukuran sampel

N = ukuran populasi

e = toleransi

Setelah didapatkan ukuran sampel, maka dilakukan proses klasifikasi terhadap semua sampel. Kemudian untuk menentukan hasil klasifikasi dari sampel tersebut akan dihitung tingkat pendeteksian api dengan rumus sebagai berikut

$$T = \frac{\text{jumlah api yang terdeteksi}}{\text{jumlah frame yang di ambil}} 100\%$$

Pada setiap rasio, untuk menentukan nilai T, maka nilai T akan dibandingkan dengan nilai pembanding (25%, 50%, 75%). Jika nilai T yang didapatkan lebih besar sama dengan nilai pembanding maka video tersebut terindikasi api, jika T lebih kecil dari nilai pemabnding maka video tersebut terindikasi non api.

Pengujian dilakukan untuk menentukan model terbaik, dimana terdapat beberapa tahap yang akan dilakukan untuk mendapatkan model yang terbaik, berikut merupakan tahap pengujian.

1. Penentuan rasio, pembanding, menentukan variabel *artificial neural network* terbaik yang terdiri dari variasi *learning rate* (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), variasi *error* (0.001, 0.0001, 0.00001), serta variasi *epoch* dari rentang nilai 100-500 dengan kelipatan 100. Adapun untuk *input*-an digunakan 4 buah fitur dari GLCM dengan 4 buah sudut yaitu 0, 45, 90, 135 derajat, jumlah *hidden layer* 1 [5] dan bias dengan nilai acak dengan rentang 0-1.
2. Setelah itu, maka dilanjutkan dengan penentuan variasi jumlah *hidden layer* yang baik (1,2,3)

Sebagai tolak ukur penentuan model terbaik maka digunakan tiga buah parameter yaitu akurasi, *precision*, *recall*.

Lalu untuk menentukan ketiga parameter tersebut maka dilakukan perhitungan dari hasil pendeteksian api dan non api. Hal tersebut dilakukan agar terdapat kesesuaian antara target dibandingkan dengan hasil dari sistem yang dibuat. Dalam melakukan *confusion matrix* seperti yang tertera Tabel 3. 6.

Tabel 3. 6 *Confusion matrix* 3x3.

	<b>Actual Class</b>	
<b>Predict Class</b>	TP	FP
	FN	TN

Keterangan:

TP= *True Positif*

TN= *True Negatif*

FP= *False Positif*

FN= *False Negatif*

Misalnya, dari pengkajian 10 video terdapat 5 video api dan 5 video non api, Hasil pendeteksian disajikan dalam Tabel 3.7

Tabel 3. 7 Hasil deteksi

NO	Status Sebenarnya	Predict
1	Non Api	Non Api
2	Non Api	Non Api
3	Non Api	Non Api
4	Non Api	Non Api
5	Non Api	Api
6	Api	Api
7	Api	Api
8	Api	Api
9	Api	Api
10	Api	Api

Berdasarkan kasus tersebut, dari 10 video yang dikaji didapatkan nilai akurasi, *percision*, *recall* sebesar:

$$\text{Akurasi} = \frac{TP+TN}{(TP+FP+FN+TN)} * 100 \% = \frac{5+4}{(5+1+0+4)} * 100\% = 90\%$$

$$\text{Percision} = \frac{TP}{(TP+FP)} * 100 \% = \frac{5}{(5+1)} * 100 \% = 83 \%$$

$$\text{Recall} = \frac{TP}{(TP + FN)} * 100 \% = \frac{5}{(5 + 0)} * 100 \% = 100 \%$$

### 3.5 Jadwal Kegiatan

Waktu yang digunakan dalam proses pengembangan sistem pendeteksi api yaitu selama enam bulan. Jadwal kegiatan pengembangan sistem pendeteksi api seperti pada Tabel 3.8

Tabel 3. 8 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] BNPB, “Data Informasi Bencana Indonesia.” [Online]. Available: <http://dibi.bnpb.go.id/>. [Accessed: 04-Oct-2019].
- [2] Sasmoko Dani. 2017. “Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis Iot Dan Sms Gateway Menggunakan Arduino”.Jurnal SIMETRIS.ISSN: 2252-4983.
- [3] Prahara Adhi. 2015. “Deteksi Kebakaran pada Video Berbasis Pengolahan Citra dengan Dukungan GPU”. Seminar Nasional Aplikasi Teknologi Informasi(SNATI). ISSN : 1907-5022.
- [4] R. R. Suryadi, I. Wijayanto, A. Rusdinar. 2017. “Perancangan Dan Implementasi Sistem Pendeteksi Api Pada Robot Pemadam Api Dengan Menggunakan Sensor Api dan Kamera ”. E-proceeding of Engineering. vol. 4, no. 3, pp. 3611–3624, ISSN : 2355-9365.
- [5] A. Kamaruddin, V. Suhartono, R. A. Pramunendar. “Deteksi Api Menggunakan Background Substraction Dan Artificial Neural Network Untuk Real Time Monitoring”.Jurnal Teknologi Informasi. vol. 12, no, ISSN: 1907-3380.
- [6] D. Retnoningrum, A. W. Widodo, M. A. Rahman. 2019. “Ekstraksi Ciri Pada Telapak Tangan Dengan Metode Local Binary Pattern ( LBP )”. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Vol. 3, No. 3, pp. 2611–2618, e-ISSN: 2548-964X.
- [7] R. Amat, J. Y. Sari, I. P. Ningrum. 2017. “Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada Smartphone”. Jurnal Ilmiah Teknologi Informasi (JUTI). Vol. 15, No. 2, p. 162-172.
- [8] Miladiah, R. Umar, I. Riadi . 2019.“Implementasi Local Binary Pattern untuk Deteksi Keaslian Mata Uang Rupiah”. Jurnal Edukasi dan Penelitian Informatika (JEPIN). Vol. 5, No. 2, pp. 197–201, e-ISSN: 2548-9364,ISSN: 2460-0741.
- [9] F. Wibowo, A. Harjoko. 2017. “Klasifikasi Mutu Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Jaringan Saraf Tiruan”. Khazanah Informatika. Vol. 3, No. 2, pp. 100–104.
- [10] N. Lihayati, R. E. Pawening, M. Furqan. 2016."Klasifikasi Jenis Daging Berdasarkan Tekstur Menggunakan Metode Gray Level Coocurent Matrix". Prosiding SENTIA. Vol. 8, pp. 305–310, ISSN: 2085-2347.
- [11] R. Widodo, A. W. Widodo, A. Supriyanto. 2018 “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix ( GLCM ) Citra Buah Jeruk Keprok ( Citrus reticulata Blanco ) untuk Klasifikasi Mutu”. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Vol. 2, No. 11, pp. 5769–5776, e-ISSN: 2548-964X.
- [12] Saifudin, A. Fadlil. 2015. “Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Coocurrence Matrix ( GLCM )”. SINERGI. Vol. 19,

No. 3, pp. 181–186, ISSN:1410-233.

- [13] Azwar. 2017. “Integrasi Ekstraksi Fitur Local Binary Pattern Dan Gray-Level Cooccurrence Metrix Untuk Pengenalan Ekspresi Mulut Pembelajar”. *ILKOM Jurnal Ilmiah*. Vol. 9, No. 1, pp. 17-24, e-ISSN:2548-7779, ISSN: 2087-1716.
- [14] Husdi. 2016. “Pengenalan Ekspresi Wajah Pengguna Elearning Menggunakan Artificial Neural Network Dengan Fitur Ekstraksi Local Binary Pattern Dan Gray Level Co-Occurrence Matrix”. *Jurnal Ilmiah ILKOM*. Vol. 8, No. 3, pp. 212-219, ISSN: 2087-1716.
- [15] Y. Ramdhani, S. Susanti, M. F. Adiwisastro, S. Topiq. 2018. “Penerapan Algoritma Neural Network Untuk Klasifikasi Kardiotokografi”. *Jurnal Informatika*. Vol. 5, No. 1, pp. 43–49, E-ISSN: 2528-2247, ISSN: 2355-6579.
- [16] E. Permata, A. Suherman, A. Maulana. 2014. “Klasifikasi Daun Tanaman Theobroma Cacao Menggunakan Metode Neural Network”. *Sentika*, pp. 160–165, 2014.
- [17] B. A. Prabowo, T. A. Budi W, F. Sthevanie. “Sistem Deteksi Api Berbasis Spatial-Temporal Dengan Metode Ekstraksi Kontur dan *Area Movement Analysis*”. Fakultas Informatika, Telkom University.
- [18] W. B. Horng and J. W. Peng. 2015. “Image-based fire detection using neural networks,”. *Department of Computer Science and Information Engineering*.
- [19] M. Nasir, Atthariq, M. Arhami. 2018. “Sistem Pendeteksi Dini Kebakaran Menggunakan Colour Image Processing dan Raspberry Pi”. *Proceeding Seminar Nasional Politeknik Negeri Lhokseumawe* . Vol. 2, No. 1, pp. 226–231, ISSN: 2598-3954.
- [20] Muhammad Dahria. 2014. “Kecerdasan buatan ( Artificial Intelligence )”. *Artif. Intell.* vol. 1, no. 2, pp. 1–10.
- [21] R. M. Haralick, K. Shanmugam. 1973. “Textural Features for Image Classification,”. *IEEE Trans. Syst. Man. Cybern.* Vol. SMC-3, No. 6, pp. 610–621.
- [22] T. Sree Sharmila. 2012. “*Efficient analysis of satellite image denoising and resolution enhancement for improving classification accuracy*”. Faculty Of Information And Communication Engineering Anna University Chennai. pp. 1–7.
- [23] Y. A. Lesnussa, S. Latuconsina, and E. R. Persulesy. 2015. “Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus: Prediksi Prestasi Siswa SMAN 4 Ambon)”. *Jurnal Matematika Integratif*. Vol. 11, No. 2, pp. 149-160, ISSN 1412-6184
- [24] Noordama. 2014. “Identifikasi varietas durio zibethinus berdasarkan sebaran trikoma daun menggunakan glcm dan knn noordama,” *Institut Pertanian. BOGOR*.