

# DETEKSI API PADA VIDEO MENGGUNAKAN METODE ARTIFICIAL NEURAL NETWORK

*(Fire Detection In Video Using Artificial neural network Method)*

Budiman Rabbani, Ramaditia Dwiyanaputra  
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA  
Email: manbudi590@gmail.com, rama@unram.ac.id

## **Abstract**

*Cameras are tools used for image capture. It is often used as a security tool for various places such as homes, roads, and others. In this study, fire object detection in the image captured by the camera was carried out. Fire is one of the causes of fire accidents if it cannot be controlled. Therefore, by utilizing the images, the best artificial neural network backpropagation model will be obtained by combining two feature extraction methods, namely local binary pattern (LBP) and Gray Level Co-occurrence Matrix (GLCM). Then, three parameters are used to evaluate the model's performance, namely: accuracy, recall, and precision. The dataset in this study is a video with a variety of fire and non-fire videos. Based on the final results of the study, the best accuracy, recall, and precision were 96%, 97%, 97%, respectively. Then, 30 videos with a variety of 15 fire videos and 15 non-fire videos were used for the validation process so that an accuracy of 86.6% was obtained with an average time of 6.029 minutes.*

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

*\*Penulis Korespondensi*

## **1. PENDAHULUAN**

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.

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]. Oleh karena itu, telah banyak dilakukannya penelitian untuk mendeteksi kebakaran dini untuk mencari alternatif lain yang dapat menutupi kekurangan sensor tersebut.

Salah satu penelitian yang berhubungan dengan deteksi api dengan memanfaatkan tangkapan kamera dilakukan 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.

Selanjutnya penelitian tidak hanya melakukan proses segmentasi warna untuk mendeteksi api, tetapi 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%. Penelitian ini menyarankan supaya 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 model arsitektur *Artificial Neural Network* (ANN) *backpropagation* dengan ekstraksi fitur gabungan antara *Local Binary Patern* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM) yang dapat mendeteksi api. Dari penelitian ini, diharapkan hasilnya dapat dimanfaatkan untuk pendeteksi api pada hasil rekaman kamera.

## 2. TINJAUAN PUSTAKA

Penelitian tentang penggunaan *Local Binary Pattern* (LBP) sebagai ekstraksi fitur telah dilakukan oleh beberapa peneliti sebelumnya. Pertama 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 di dapatkan akurasi sebesar 92,31% [6]. Kedua 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]. Ketiga 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 tentang penggunaan 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 papaya 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 tentang penggunaan 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 tentang penggunaan metode *Artificial neural network* (ANN) *backpropagation* sebagai metode pengklasifikasian pernah digunakan oleh beberapa peneliti antara lain klasifikasi kardiografi 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.

## 2.1 Sampel

Sugiyono (2011 : 81) menyatakan bahwa sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut. Populasi yang dijadikan sampel pada penelitian yaitu konsumen yang melakukan pembelian dalam kurun waktu di tahun 2014. Karena populasi dari yang menggunakan sepatu *Customade* diketahui maka teknik sampel dalam penelitian ini adalah probability sampling dengan menggunakan sampling acak sederhana (Simple Random Sampling). Berikut pengambiolan sampel dalam pengambilan jumlah sampel penulis menggunakan Slovin [19]:

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

Dimana :

n = ukuran sampel

N = ukuran populasi

e = persen kelonggaran ketidak telitian karena kesalahan pengambilan sampel yang masih dapat ditolerir

## 2.2 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[20]. Api memiliki aturan warna untuk api gelap dan api terang [17]. Nilai piksel api dapat dilihat pada persamaan 2 dan 3.

$$\text{Piksel Api Gelap} \begin{cases} 179 < R \\ 70 < G < 159 \\ B < 119 \end{cases} \quad (2)$$

$$\text{Piksel Api Terang} \begin{cases} 210 < R \\ 123 < G \\ B < 176 \end{cases} \quad (3)$$

Gambar 1 merupakan contoh gambar yang terdapat objek api.



Gambar 1 contoh gambar adanya objek api

## 3. METODE PENELITIAN

### 3.1 Diagram Alir 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.

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.

### 3.2 Pengambilan Dataset

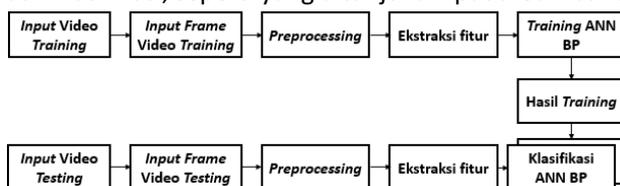
Pada penelitian ini digunakan dataset terdiri atas 100 video yang didapatkan dari internet dan membuat sendiri melalui kamera telpon seluler dengan mengabaikan kondisi terang atau gelap dari lingkungan pengambilan video, dengan perbandingan 50 video jenis api dan 50 video jenis *non*-api. Durasi video 10 detik dan kualitas video 30 *frame per second* (fps) dengan ukuran resolusi video maksimal sebesar 640x640. Dari 100 video akan diambil framenya dalam bentuk file .jpg, dimana per-video akan diambil sampel dengan menggunakan metode sampling yakni *slovin* [19] dengan toleransi sebesar 8%. Cara perhitungan pengambilan jumlah sampling dengan metode *slovin* sebagai berikut.

$$\text{jumlah\_sampel} = \frac{300}{1 + 300 * 0.08^2} = 103$$

Dataset tersebut digunakan untuk melakukan proses training dan *testing* untuk mendapatkan model *artificial neural network* yang baik. Dari hasil model yang telah didapatkan akan dilanjutkan proses validasi pada 30 data video yang terdiri dari 15 video jenis api dan 15 video jenis *non*-api. Pengujian ini dilakukan untuk mengetahui nilai ambang (*threshold*) yang baik untuk mendeteksi api

### 3.3 Perancangan Algoritma

Sistem ini secara garis besar terdapat tiga proses utama yaitu *training* (pelatihan), *testing* (pengujian) dan klasifikasi, seperti yang ditunjukkan pada Gambar 3



Gambar 3. Blok diagram sistem

Dimana:

ANN = *Artificial Neural Network*

BP = *Backpropagation*

### 3.4 Preprocessing

Tahap *pre-processing* yang terdiri atas proses segmentasi 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.

a. Segmentasi

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].

Pada Gambar 4 dan Gambar 5 merupakan tampilan dari tahap segmentasi



Gambar 4 Citra *input-an*



Gambar 5 Hasil segmentasi

b. *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 disegmentasi dalam 1 frame kemudian dipilih bagian dengan lokasi api terbanyak. Pada Gambar 6 merupakan tampilan dari tahap *cropping*.



Gambar 6 Proses *Cropping*

### 3.5 Ekstraksi Fitur

*Extraction feature* 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 Pattern* (LBP) dan *Gray Level Co-occurrence Matrix* (GLCM). Proses ekstraksi dibagi menjadi 2 tahap Antara lain

a. LBP

*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

mendeskrripsikan 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 1, dan Persamaan 2 untuk menghitung nilai biner dari hasil *thresholding* menjadi angka desimal [6].

$$s(x) \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4)$$

$$LBPp,r(xc, y) = \sum_{p=0}^{p=1} S(gp - gc)2^p \quad (5)$$

Dimana :

P : banyaknya piksel tetangga

R : nilai jarak/radius

gc : nilai dari piksel x dan y

gp : nilai piksel tetangga

xc , yc : koordinat pusat

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 digabungkan 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].

#### b. GLCM

Tahap selanjutnya yaitu *extraction feature* merupakan bagian dari teknik pengenalan pola (*pattern recognition*) yang bertujuan untuk mengambil atau mengekstraksi nilai-nilai unik dari suatu objek yang membedakan dengan objek yang lain. Penelitian ini menggunakan metode *extraction feature Gray Level Co-occurrence Matrix* (GLCM). *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 (θ) tertentu dalam citra[11].

Pada penelitian Haralick dkk. mengusulkan fitur tekstural yang mengandung informasi tentang karakteristik tekstur[21]. Pada Tabel I menjelaskan rumus perhitungan fitur tersebut.

TABEL I. TABEL FITUR TEKSTUR GLCM

N o	Properti	Rumus	Ps
1	Angular Second Moment : Energy/ Uniformity	$f_1 = \sum_i \sum_j \{p(i, j)\}^2$	(6)
2	Contrast	$f_2 = \sum_{n=0}^{Ng-1} n^2 \left\{ \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p(i, j) \right\}_{ i-j =n}$	(7)
3	Correlation	$f_3 = \frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$ Dimana $\mu$ merupakan means dan $\sigma$ merupakan standar deviasi dari p. $N_g$ merupakan jumlah kolom/baris. n jumlah pixel. i merupakan baris ke-i, j merupakan kolom ke-j.	(8)
4	Inverse Difference Moment : Homogeneity	$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j)$	(9)
5	Entropy	$f_9 = -\sum_i \sum_j p(i, j) \log(p(i, j))$	(10)

Pada jurnal sebelumnya [22] didapatkan 5 fitur yang paling baik untuk digunakan yaitu *Energy*, *Contrast*, *Correlation*, *Homogeneity* dan *Entropy*.

*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 (θ) tertentu dalam citra [23]. Proses ekstraksi fitur dengan metode GLCM dibagi menjadi dua tahap, yaitu pembentukan matriks GLCM dan perhitungan nilai fitur GLCM.

#### 1. Pembentukan matriks GLCM

Misalkan terdapat citra *grayscale* dengan matriks seperti pada Gambar 8.

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 2 & 2 & 2 \end{bmatrix}$$

Gambar 8 Contoh matriks citra *grayscale*

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

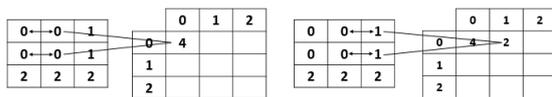
- Langkah pertama yaitu membuat matriks dengan ordo n x n dimana n merupakan nilai elmen terbesar dari matriks *grayscale* A. pada matriks A nilai elmen terbesar adalah 2 sehingga ukuran

matriks yang akan dibuat adalah 2x2 dengan nilai indeks dimulai dari 0 seperti yang terlihat pada Gambar 9 berikut.

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

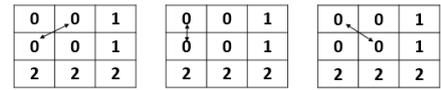
Gambar 9 Matriks GLCM 3x3

- 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.
- 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 10 menunjukkan proses pembentukan matriks GLCM pada sudut  $0^\circ$ .



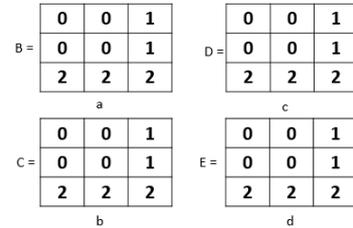
Gambar 10 Pembentukan matriks GLCM sudut  $0^\circ$

- Iterasi selanjutnya adalah mengecek jumlah matriks ketetanggaan yang bernilai (0,1) untuk diisi di matriks B di elemen (0,1), begitu seterusnya sampai elemen terakhir pada matriks B.
- 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 11.



Gambar 11 Arah matriks GLCM a. Sudut  $45^\circ$ . (b) Sudut  $90^\circ$ . (c) Sudut  $135^\circ$

- Didapatkan empat buah matriks GLCM seperti pada Gambar 12.



Gambar 12 Hasil pembentukan matriks GLCM (a)  $\theta = 0^\circ d = 1$ . (b)  $\theta = 45^\circ d = 1$ . (c)  $\theta = 90^\circ d = 1$ . (d)  $\theta = 135^\circ d = 1$

- Matriks GLCM kemudian dinormalisasi agar jumlah seluruh elemennya sama dengan satu. Gambar 13 merupakan hasil normalisasi terhadap matriks B.

$$F = \begin{array}{|c|c|c|} \hline 0.333 & 0.1666 & 0.0 \\ \hline 1.666 & 0.0 & 0.0 \\ \hline 0.0 & 0.0 & 0.333 \\ \hline \end{array}$$

Gambar 13 Matriks GLCM yang telah dinormalisasi

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

## 2. 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*.

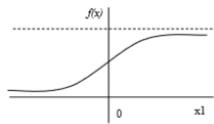
## 3.6 Konsep Dasar Jaringan Syaraf Tiruan

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 [24], yaitu :

- 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.
- Lapisan tersembunyi, unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di mana *output*-nya tidak dapat secara langsung diamati.

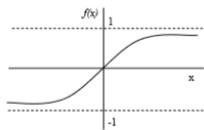
### 3.7 Fungsi Aktivasi

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 13.



Gambar 13 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 14.



Gambar 14 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[24] :  $f(x) = x$

### 3.8 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 [25]:

Langkah 0 : Bobot dengan bilangan acak kecil antara 0 hingga 1 diinisialisasi

Langkah 1 : Jika kondisi penghentian belum terpenuhi, langkah 2 - 9 dilakukan

Langkah 2 : Langkah 3 - 8 berikut ini dilakukan untuk setiap data pelatihan.

### Tahap I : Propagasi Maju

Langkah 3 : Sinyal input diterima unit *input* dan diteruskan ke *hidden neuron*.

Langkah 4 : *Output* di setiap *hidden neuron*  $z_j$  ( $j = 1, 2, \dots, p$ ) dihitung.

$$z_{net_j} = v_{j_0} + \sum_{i=0}^n x_i v_{ji} \quad (11)$$

Dimana : bias = 1

Fungsi aktivasi *sigmoid biner*:

$$z_j = f(z_{net_j}) = \frac{1}{1+e^{-z_{net_j}}} \quad (13)$$

Fungsi aktivasi *sigmoid bipolar*:

$$z_j = f(z_{net_j}) = \frac{2}{1+e^{-z_{net_j}}} - 1 \quad (14)$$

Langkah 5 : *Output* di setiap unit  $y_k$  ( $k = 1, 2, \dots, m$ ) dihitung

$$y_{net_k} = bias + \sum_{j=1}^p z_j w_{kj} \quad (15)$$

Dimana : bias =1.

Fungsi aktivasi *sigmoid biner*:

$$z_j = f(z_{net_j}) = \frac{1}{1+e^{-z_{net_j}}} \quad (16)$$

Fungsi aktivasi *sigmoid bipolar*:

$$z_j = f(z_{net_j}) = \frac{2}{1+e^{-z_{net_j}}} - 1 \quad (17)$$

### Tahap II : Propagasi mundur

Langkah 6 : Faktor  $\delta$  unit *output* dihitung berdasarkan kesalahan di setiap unit *output*  $y_k$  ( $k = 1, 2, \dots, m$ )

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (18)$$

Suku perubahan bobot  $w_{kj}$  dihitung dengan laju pembelajaran  $\alpha$ .

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (19)$$

dimana :  $k = 1, 2, \dots, m$  dan  $j = 0, 1, \dots, p$

Langkah 7 : Faktor  $\delta$  pada setiap *hidden neuron* dihitung berdasarkan kesalahan di setiap *hidden neuron*  $z_j$  ( $j = 1, 2, \dots, p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (20)$$

Faktor  $\delta$  unit tersembunyi:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (21)$$

Suku perubahan bobot  $v_{ji}$  dihitung.

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (22)$$

dimana :  $j = 1, 2, \dots, p$  dan  $i = 0, 1, \dots, n$

### Tahap III : Perubahan bobot

Langkah 8 : Semua perubahan bobot dihitung.

Perubahan bobot garis yang menuju ke unit *output*:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (23)$$

Dimana :  $k = 1, 2, \dots, m$  dan  $j = 0, 1, \dots, p$

Perubahan bobot garis yang menuju ke *hidden neuron*:

$$v_{ji}(\text{baru}) = v(\text{lama}) + \Delta v_{ji} \quad (24)$$

dimana :  $j = 1, 2, \dots, p$  dan  $i = 0, 1, \dots, n[25]$ .

### 3.9 Teknik Pengujian

Tahapan pengujian sistem merupakan tahapan yang ditujukan sebagai bahan evaluasi sistem untuk mengetahui model yang dibuat telah sesuai dengan perancangan ataupun model diharapkan. Adapaun parameter yang digunakan sebagai tolak ukur penentuan model terbaik yaitu akurasi, presisi, *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 model yang dibuat. Dalam melakukan *confusion matrix* seperti yang tertera Tabel I.

TABEL II. CONFUSION MATRIX 3X3.

Predict Class	Actual Class	
	TP	FP
FN	TN	

Dimana:

TP= True Positif

TN= True Negatif

FP= False Positif

FN= False Negatif

Berikut ini merupakan cara menghitung nilai akurasi, *recall*, presisi menggunakan *Confusion matrix* pada Tabel II.

$$\text{Akurasi} = \frac{TP+TN}{(TP+FP+FN+TN)} * 100 \% \quad (25)$$

$$\text{Presisi} = \frac{TP}{(TP+FP)} * 100 \% \quad (26)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} * 100 \% \quad (27)$$

## 4. HASIL DAN PEMBAHASAN

### 4.1 Skenario Pengujian

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 *slovin* yang perhitungannya dapat dilihat pada persamaan 1.

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

$$T = \frac{\text{jumlah api yang terdeteksi}}{\text{jumlah frame yang di ambil}} 100\% \quad (28)$$

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 pembanding 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.

- Penentuan rasio, 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.
- Setelah itu, maka dilanjutkan dengan penentuan variasi jumlah *hidden layer* yang baik (1,2,3), kemudian menentukan nilai ambang

### 4.2 Pengaruh Variasi Pembagian Dataset

Pada penelitian ini dilakukan pengujian pengaruh pembagian dataset terhadap hasil akurasi yang didapatkan. Dimana dataset dibagi menjadi 70:30, 80:20, 90:10, dengan jumlah *input-an* yakni 20 dan 1 *hidden layer*. Berikut hasil akurasi terbaik dari pembagian dataset yang disajikan pada Tabel II.

TABEL III. AKURASI TERBAIK VARIASI PEMBAGIAAN DATASET

		DataSet			
		60:40	70:30	80:20	90:10
Test	Akurasi	0.92	<b>0.96</b>	0.95	0.96
	Percission	0.93	<b>0.97</b>	0.95	0.96
	Recall	0.93	<b>0.97</b>	0.95	0.96
Train	Akurasi	0.93	<b>0.97</b>	0.96	0.96
	Percission	0.93	<b>0.97</b>	0.96	0.97
	Recall	0.93	<b>0.97</b>	0.96	0.97
error		0.08	<b>0.04</b>	0.04	0.04

Dari Tabel III dapat disimpulkan bahwa akurasi tertinggi yang didapatkan sebesar 96% pada

pembagian dataset 70:30. Kemudian, untuk pengaruh variasi pembagian dataset terhadap tingkat akurasi yang didapatkan tidak memberikan dampak yang besar. Hal ini karena nilai akurasi yang didapatkan tidak mengalami perbedaan yang signifikan. Dengan demikian 70:30 merupakan pembagian data set yang akan digunakan untuk tahap penelitian selanjutnya

### 4.3 Pengaruh Batas Error Terhadap Unjuk Kerja ANN

Pada penelitian ini dilakukan pengujian pengaruh batas error terhadap hasil akurasi. Dimana variasi error yang digunakan pada penelitian ini adalah 0.001, 0.0001, 0.00001 dengan 20 jumlah input-an dan pembagian dataset 70:30. Berikut pada Tabel III disajikan hasil akurasi terbaik dari variasi error yang didapatkan.

TABEL IV. AKURASI TERBAIK VARIASI ERROR

		Batas Error		
		0.001	0.0001	0.00001
Test	Akurasi	<b>0.96</b>	0.94	0.94
	Percission	<b>0.97</b>	0.95	0.95
	Recall	<b>0.97</b>	0.95	0.95
Train	Akurasi	<b>0.97</b>	0.94	0.95
	Percission	<b>0.97</b>	0.95	0.95
	Recall	<b>0.97</b>	0.95	0.95
error		<b>0.04</b>	0.05	0.04

Dari tabel Tabel IV, dapat disimpulkan bahwa pengaruh batas error yang telah ditentukan pada penelitian ini adalah tak berpengaruh sama sekali. Hal ini karena error yang didapatkan masih jauh dari dari batas error yang telah ditentukan

### 4.4 Uji Pengaruh Pengaruh Batas Epoch

Pada penelitian ini dilakukan pengujian pengaruh batas epoch terhadap akurasi yang didapatkan. Dimana variasi epoch yang digunakan adalah 100, 200, 300, 400, 500 dengan 20 jumlah input-an dan pembagian dataset sebesar 70:30. Berikut pada Tabel IV disajikan hasil akurasi dari penelitian yang didapatkan

TABEL V. AKURASI TERBAIK VARIASI BATAS EPOCH

Epoch	Test			Train			error
	Akurasi	Percission	Recall	Akurasi	Percission	Recall	
100	0.92	0.93	0.93	0.93	0.94	0.94	0.07
200	0.92	0.93	0.93	0.93	0.93	0.93	0.07
300	0.93	0.94	0.94	0.94	0.94	0.94	0.06
400	0.94	0.94	0.94	0.95	0.95	0.95	0.05
500	<b>0.96</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.04</b>
600	0.94	0.94	0.94	0.94	0.95	0.94	0.06
700	0.94	0.94	0.94	0.94	0.95	0.95	0.06
800	0.94	0.95	0.95	0.95	0.95	0.95	0.06
900	0.95	0.95	0.95	0.95	0.96	0.96	0.05
1000	0.95	0.95	0.95	0.96	0.96	0.96	0.05

Dari Tabel V, dapat disimpulkan bahwa hasil tertinggi yang didapatkan terdapat pada epoch 500 dengan tingkat akurasi, *percission* dan *recall* berturut-turut sebesar 96%, 97%, 97%. Kemudian, pengaruh peningkatan batas epoch yang telah ditentukan pada penelitian adalah cukup memberikan pengaruh terhadap peningkatan tingkat akurasi yang didapatkan. Namun titik maksimal peningkatan akurasi yang didapatkan berada pada epoch 500. Dengan demikian 500 merupakan batas epoch yang akan digunakan untuk tahap penelitian selanjutnya.

### 4.5 Uji Pengaruh Learning rate

Pada penelitian ini dilakukan pengujian terhadap pengaruh variasi learning rate terhadap tingkat akurasi yang didapatkan. Dimana variasi learning rate yang digunakan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dengan pembagian dataset serta epoch menggunakan hasil terbaik sebelumnya. Berikut pada Tabel V disajikan hasil akurasi terbaik dari penelitian yang didapatkan.

TABEL VI. AKURASI TERBAIK VARIASI LEARNING RATE

LR	Test			Train			error
	Akurasi	Persisi	Recall	Akurasi	Persisi	Recall	
0.1	0.93	0.93	0.93	0.93	0.94	0.94	0.07
0.2	0.93	0.94	0.93	0.94	0.94	0.94	0.07
0.3	0.94	0.95	0.95	0.95	0.96	0.96	0.06
<b>0.4</b>	<b>0.96</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>0.04</b>
0.5	0.93	0.94	0.94	0.94	0.94	0.94	0.07
0.6	0.93	0.94	0.94	0.94	0.94	0.94	0.07
0.7	0.93	0.94	0.94	0.94	0.94	0.94	0.07
0.8	0.93	0.93	0.93	0.93	0.93	0.93	0.07
0.9	0.94	0.94	0.94	0.94	0.95	0.95	0.05

Dari Tabel VI, dapat disimpulkan bahwa pengaruh peningkatan nilai learning rate pada penelitian ini tidak memberikan pengaruh terhadap tingkat akurasi yang didapatkan. Dimana hasil terbaik yang didapatkan pada learning rate 0.4 dengan akurasi sebesar 96%. Dengan demikian 0.4 merupakan learning rate yang akan digunakan untuk tahap penelitian selanjutnya

### 4.6 Uji Pengaruh Variasi Jumlah Hidden layer

Pada penelitian selanjutnya yakni tentang pengaruh perbedaan variasi hidden layer dengan parameter-parameter yang terbaik yang telah didapatkan pada tahap-tahap sebelumnya. Berikut parameter yang digunakan dalam pengujian sebagai berikut:

- Input layer : 20 neuron
- Output layer : 1 neuron
- Fungsi aktivasi : Sigmoid biner

- d. Batas *epoch* : 500
- e. Batas *error* : 0.001
- f. *Learning rate* : 0.4
- g. Pembagian dataset : 70:30

Dengan rincian *hidden layer* untuk masing-masing arsitektur jaringan yang akan diuji dapat dilihat pada Tabel VII. Hasil dari pengujian masing-masing skenario dapat dilihat pada Tabel VIII.

TABEL VII. HIDDEN LAYER ARSITEKTUR PENGUJIAN

Arsitektur ke-	Jumlah <i>hidden layer</i>	Jumlah neuron
1	1	7
2	2	7, 4
3	3	7, 4, 2

TABEL VIII. HASIL PENGUJIAN ARSITEKTUR JARINGAN

		<i>Hidden Layer</i>		
		1	2	3
Test	Akurasi	<b>0.96</b>	0.93	0.94
	Percission	<b>0.97</b>	0.94	0.94
	Recall	<b>0.97</b>	0.94	0.94
Train	Akurasi	<b>0.97</b>	0.94	0.94
	Percission	<b>0.97</b>	0.95	0.95
	Recall	<b>0.97</b>	0.95	0.95
error		<b>0.04</b>	0.07	0.06

Dari Tabel VIII, dapat disimpulkan bahwa arsitektur dengan jumlah 1 *hidden layer* merupakan hasil dengan akurasi tertinggi. Dengan demikian, arsitektur tersebut akan digunakan pada tahap pengujian selanjutnya. Penggunaan arsitektur ini dilakukan dengan cara memindahkan bobot dan bias yang diperoleh dari Arsitektur yang didapatkan secara manual, dan melakukan fase *feedforward* dari algoritma *Backpropagation*.

#### 4.7 Uji Pengaruh Nilai Ambang (*Threshold*)

Selanjutnya dilakukan proses pencarian nilai ambang (*threshold*) yang terbaik yang digunakan untuk menentukan video termasuk api atau non api dengan menggunakan arsitektur ANN yang telah didapatkan pada pengujian sebelumnya. Adapun video yang digunakan untuk menentukan pembandingan terbaik adalah dataset video api dan non api yang digunakan pada proses training dan *testing* sebelumnya, dalam artian di pengujian kembali. Pada Tabel VIII disajikan hasil pengujian yang didapatkan dari 50 video api dan 50 video non api.

TABEL IX. HASIL PENGUJIAN ARSITEKTUR JARINGAN

	25%		50%		75%	
	Api	Non Api	Api	Non Api	Api	Non Api
TRUE	49	49	46	47	36	41
FALSE	1	1	4	3	14	9

Dari Tabel IX, dapat disimpulkan bahwa nilai ambang (*threshold*) yang terbaik untuk menentukan api atau non api dari sebuah video adalah 25%.

#### 4.8 Validasi

Pada tahap ini, dilakukan proses pengujian ulang terhadap arsitektur yang telah didapatkan dengan menggunakan video yang berbeda dari video yang digunakan pada tahap training dan *testing*. Dimana video berjumlah 30 buah yang terdiri dari 15 video api dan 15 video non api. Berikut parameter yang digunakan dalam validasi sebagai berikut:

- a. *Input layer* : 20 neuron
- b. *Output layer* : 1 neuron
- c. Fungsi aktivasi : Sigmoid biner
- d. Batas *epoch* : 500
- e. Batas *error* : 0.001
- f. *Learning rate* : 0.4
- g. Pembagian dataset : 70:30
- h. Ambang : 25%

Berikut pada Tabel IX merupakan tabel hasil pengujian ulang dengan 30 video tersebut.

TABEL X. TABEL HASIL VALIDASI

Video	Status Sebenarnya	Prediksi	Waktu(menit)	Keterangan
1	api	api	6,36	valid
2	api	api	5,44	valid
3	api	api	6,80	valid
4	api	api	7,80	valid
5	api	api	8,27	valid
6	api	api	8,12	valid
7	api	api	3,83	valid
8	api	api	7,68	valid
9	api	api	3,39	valid
10	api	api	2,90	valid
11	api	api	3,21	valid
12	api	api	6,14	valid
13	api	api	6,11	tidak valid
14	api	api	3,89	valid
15	api	non api	2,78	tidak valid
16	non api	non api	11,1	valid
17	non api	non api	4,08	valid
18	non api	api	2,90	tidak valid
19	non api	non api	8,03	valid
20	non api	non api	7,49	valid
21	non api	non api	3,19	valid
22	non api	non api	3,42	valid
23	non api	non api	9,46	valid
24	non api	non api	2,49	valid
25	non api	api	7,96	tidak valid
26	non api	non api	19,4	valid
27	non api	non api	7,57	valid
28	non api	non api	6,36	valid
29	non api	non api	2,06	valid
30	non api	non api	2,64	valid

Dari Tabel X, tingkat akurasi dan rata-rata waktu yang dibutuhkan untuk mendeteksi api atau non api tersebut dihitung sebagai berikut:

$$\text{Akurasi} = \frac{26}{30} \times 100 = 86.6 \%$$

$$\text{Waktu rata-rata} = \frac{\sum \text{waktu}}{30} = 6.029 \text{ menit}$$

Berdasarkan proses validasi yang telah dilakukan terdapat nilai waktu yang cukup besar yakni 19,4 menit yang dibutuhkan untuk mengklasifikasikan video termasuk api atau *non* api. Hal tersebut terjadi karena ukuran hasil *cropping* yang didapatkan pada *frame* video cukup besar sehingga waktu yang dibutuhkan untuk mengklasifikasikan juga cukup besar. Kemudian tingkat akurasi yang diperoleh cukup baik, yaitu sebesar 86.6% dari 30 video yang diuji dengan waktu rata-rata yang dibutuhkan untuk mendeteksi api atau non api pada sebuah video sejumlah 6.029 menit. Dari hasil tersebut maka dapat disimpulkan bahwa model arsitektur ANN *Backpropagation* tersebut cukup baik untuk mendeteksi api.

## KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Berdasarkan pengujian dan analisis dari model arsitektur ANN yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

- Model arsitektur jaringan *backpropagation* terbaik untuk kasus pendeteksian api pada video ini adalah arsitektur jaringan dengan 1 *hidden layer*, dan *learning rate* sebesar 0.4, *epoch* 500, dan pembagian dataset sebesar 70:30, yang menghasilkan akurasi training sebesar 97.02% dan akurasi *testing* sebesar 96.78% dari 100 data video.
- Tingkat akurasi model arsitektur ANN setelah di uji menggunakan ulang menggunakan 30 video yang berbeda diperoleh akurasi sebesar 86.6% dengan waktu rata-rata 9,68 menit.
- Penggunaan hanya menggunakan pixel warna untuk posisi dari api berjalan dengan kurang sempurna. Hal ini karena terdapat beberapa kesalahan dalam penentuan posisi api.

### 5.2. Saran

Ada beberapa saran yang dapat penulis berikan apabila penelitian ini akan dikembangkan kembali antara lain sebagai berikut.

- Menggunakan metode segmentasi yang lain karena terdapat beberapa kesalahan dalam proses segmentasi dengan metode *Adaptive*

*Gaussian Mixture Model* (GMM).

- Menambahkan proses perbaikan kualitas citra pada tahap *preprocessing*
- Tambahkan proses *resize* pada tahap *pre-processing* karena untuk menyamakan ukuran citra hasil *cropping*.

## DAFTAR PUSTAKA

- [1] BNPB, "Data Informasi Bencana Indonesia." [Online]. Available: <http://dibi.bnpb.go.id/>. [Accessed: 04-Oct-2019].
- [2] S. Dani, A. Mahendra, "Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis Iot Dan Sms Gateway Menggunakan Arduino," Jurnal SIMETRIS, vol. 8, no. 2, November 2017..
- [3] A. Prahara, "Deteksi Kebakaran pada Video Berbasis Pengolahan Citra dengan Dukungan GPU," *Semin. Nas. Apl. Teknol. Inf.*, ISSN: 1907 – 5022, 2015.
- [4] R. R. Suryadi, I. Wijayanto, A. Rusdinar, "Perancangan Dan Implementasi Sistem Pendeteksi Api Pada Robot Pemadam Api Dengan Menggunakan Sensor Api Dan Design and Implementation System Fire Detection on Fire," *e-Proceeding of Engineering* vol. 4 no. 3, Desember 2017.
- [5] A. Kamarudin, V. Suhartono, R. A. Pramunendar, "Deteksi Api Menggunakan Background Substraction Dan", Jurnal Teknologi Informasi vol. 12 no.1, April 2016.
- [6] D. Retnoningrum, A. W. Widodo, M. A. Rahman, "Ekstraksi Ciri Pada Telapak Tangan Dengan Metode Local Binary Pattern ( LBP ),", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer vol. 3, no. 3, Maret 2019.
- [7] R. Amat, J. Y. Sari, I. P. Ningrum, "Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana Dan Katakana Pada Smartphone," *JUTI J. Ilm. Teknol. Inf.*, vol. 15 no. 2, Juli 2017.
- [8] Miladiah, Rusydi. Umar, I. Riadi, "Implementasi Local Binary Pattern untuk Deteksi Keaslian Mata Uang Rupiah," *Jepin*, vol. 5 no. 2, Agustus 2019.
- [9] F. Wibowo, A. Harjoko, "Klasifikasi Mutu Pepaya Berdasarkan Ciri Tekstur GLCM Menggunakan Jaringan Saraf Tiruan," *Khazanah Informatika* Vol. 3 No.2, Desember 2017.
- [10] N. Lihayati, R. E. Pawening, M. Furqan, " Klasifikasi Jenis Daging Berdasarkan Tekstur Menggunakan Metode Gray Level Coocurent Matrix," *Prosiding SENTIA* Vol. 8- ISSN: 2085-2347, 2016.
- [11] R. Widodo, A. W. Widodo, A. Supriyanto, "Pemanfaatan Ciri Gray Level Co-Occurrence Matrix ( GLCM ) Citra Buah Jeruk Keprok ( Citrus reticulata Blanco ) untuk Klasifikasi Mutu," *J. Pengemb. Teknol.*

- Inf. dan Ilmu Komput.* vol. 2 No. 11, November 2018.
- [12] S. Saifudin, A. Fadlil, "Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Coocurrence Matrix ( Glcm ) Dengan Klasifikasi Jarak Euclidean," *SINERGI* Vol. 19, No. 3, Oktober 2015.
- [13] A. Azwar, "Integrasi Ekstraksi Fitur Local Binary Pattern Dan Gray-Level Coocurrence Metrix Untuk Pengenalan Ekspresi Mulut Pembelajar," *Ilkom Jurnal Ilmiah*, vol. 9, no. 1, 2017.
- [14] H. Husdi, "Pengenalan Ekspresi Wajah Pengguna Elearning Menggunakan Artificial Neural Network Dengan Fitur Ekstraksi Local Binary Pattern Dan Gray Level Co-Occurrence Matrix," *Ilkom Jurnal Ilmiah* vol. 8 no. 3, 2016.
- [15] Y. Ramdhani, S. Susanti, M. F. Adiwisastra, S. Topiq, "Penerapan Algoritma Neural Network Untuk Klasifikasi Kardiotokografi," *Jurnal Informatika* vol. 5 no. 1, April 2018.
- [16] E. Permata, A. Suherman, A. Maulana, "Klasifikasi Daun Tanaman Theobroma Cacao Menggunakan Metode Neural Network," Seminar. Nasional Teknologi Informasi dan Komunikasi ISSN: 2089-9813, 2014.
- [17] B. A. Prabowo, T. A. Budi, F. Sthevanie, "Sistem Deteksi Api Berbasis Spatial Temporal dengan Metode Ekstraksi Kontur dan Area Movement Analysis," Telkom University,.
- [18] W. B. Horng, J. W. Peng, "Image-based fire detection using neural networks," Department of Computer Science and Information Engineering, Oktober 2006.
- [19] M. Pradana, A. Reventiary, "Pengaruh Atribut Produk Terhadap Keputusan Pembelian Sepatu Merek Customade (Studi Di Merek Dagang Customade Indonesia)," *Jurnal.Manajemen* vol. 6 no. 1, Juni 2016.
- [20] M. Nasir, M. Arhami, "Sistem Pendeteksi Dini Kebakaran Menggunakan Colour Image Processing dan Raspberry Pi," *Proceeding Seminar Nasional Politeknik Negeri Lhokseumawe* vol. 2 no. 1, September 2018.
- [21] R. M. Haralick and K. Shanmugam, "Textural Features for Image Classification," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.
- [22] T. Sree Sharmila, "Efficient analysis of satellite image denoising and resolution enhancement for improving classification accuracy," pp. 1–7, 2014.
- [23] Noordama, "Identifikasi varietas durio zibethinus berdasarkan sebaran trikoma daun menggunakan glcm dan knn noordama," *Inst. Pertan. BOGOR*, 2014.
- [24] Y. A. Lesnussa, S. Latuconsina, E. R. Persulesy, "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, Oktober 2015.
- [25] I. N. T. Adnyana, I G. P. S. Wijaya, M. A. Albar, "Jaringan Syaraf Tiruan *Backpropagation* Untuk Peramalan Suhu Minimum dan Maksimum, Kelembaban, Tekanan Udara, Jumlah Hari Hujan, dan Curah Hujan Bulanan Di Kota Mataram," *J- Cosine* vol. 3, Desember 2019