

**IMPLEMENTASI ALGORITMA *BACKPROPAGATION*  
UNTUK PENGENALAN WAJAH BERBASIS CITRA**

**SKRIPSI**

**OLEH :**

**MAYA DANI SARI  
178160077**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 12/7/23

Access From (repository.uma.ac.id)12/7/23

**IMPLEMENTASI ALGORITMA *BACKPROPAGATION* UNTUK  
PENGENALAN WAJAH BERBASIS CITRA**

**SKRIPSI**

Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area



**OLEH :**

**MAYA DANI SARI  
178160077**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 12/7/23

Access From (repository.uma.ac.id)12/7/23



### HALAMAN PERNYATAAN

Saya menyatakan dengan sungguh-sungguh bahwa tugas akhir ini adalah hasil penelitian, pemikiran dan presentasi asli saya sendiri. Saya tidak mencantumkan tanpa pengakuan bahan yang telah diterbitkan atau ditulis oleh orang lain sebelumnya, atau sebagai bahan yang telah diajukan untuk gelar diploma di Universitas Medan Area atau perguruan tinggi lainnya. Apabila dikemudian hari terdapat kejanggalan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai dengan ketentuan yang berlaku di Universitas Medan Area. Demikian pernyataan ini saya buat.



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai civitas akademika Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Maya Dani Sari  
NPM : 178160077  
Program Studi : Teknik Informatika  
Fakultas : Teknik  
Jenis Karya : Tugas Akhir/Skripsi/Tesis

Demi pengembangan ilmu pengetahuan, setuju untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul: **"Implementasi Algoritma Backpropagation Untuk Pengenalan Wajah Berbasis Citra"** Bersama dengan perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti yang bersifat *non-eksklusif* ini, Universitas Medan Area berhak menyimpan, mengalihkan media/Format, mengelola dalam bentuk database, memelihara dan mempublikasikan tugas akhir/tesis/skripsi saya selama saya tetap menyebut nama saya sebagai pencipta/penulis dan sebagai pemilik hak cipta. Demikian Surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Medan  
Pada tanggal: 07 Juni 2023  
Yang menyatakan



(Maya Dani Sari)

## ABSTRAK

Umumnya sistem absensi karyawan pada kantor dilakukan dengan mengisi buku absen menggunakan mesin sidik jari dimana jika dilihat dari segi keamanan, sistem ini mempunyai kelemahan antara lain adalah sering mengalami *human error* seperti *scan* sidik jari sulit diterima. Hal ini dapat diakibatkan kondisi permukaan jari yang tidak normal, seperti basah, kotor, terlalu kering, ujung jari cacat dan akhirnya sistem menolak, maka dalam menyelesaikan masalah tersebut diatas memerlukan suatu metode cepat tepat dan akurat. Pada penelitian ini dilakukan pengenalan wajah berbasis citra dengan mengimplementasikan algoritma *Backpropagation*. Dataset citra wajah yang digunakan pada penelitian ini bersumber dari situs [kaggle.com](https://www.kaggle.com) dengan jumlah wajah yang akan dilatih sebanyak 1866 citra wajah dengan berbagai ekspresi setiap wajah. Pada sistem ini ada dua tahap yang dilakukan yaitu pelatihan semua citra wajah sebagai dataset untuk memperoleh bobot setiap citra dan tahap selanjutnya dilakukan pengujian yaitu tahap pengenalan. Hasil percobaan bahwa aplikasi dapat melakukan pembacaan nilai piksel citra pelatihan dengan parameter jaringan maksimal *error* sebesar 0.001, *learning rate* sebesar 0.30 dan 0.50. Aplikasi dapat melakukan pengenalan dengan hasil terbaik pada nilai *learning rate* 0.50 dengan nilai akurasi pengenalan sebesar 82 %.

Kata kunci: Sistem Absensi, Keamanan Data, Pengenalan wajah, Jaringan Syaraf Tiruan, Algoritma *Backpropagation*.

## ABSTRACT

*In general, the employee attendance system at the office is done by filling out the attendance book using a fingerprint machine where from a security point of view, this system has weaknesses, among others, is that it often experiences human errors such as fingerprint scans that are difficult to accept. This can be caused by abnormal finger surface conditions, such as wet, dirty, too dry, defective finger tips and finally the system refuses, so solving the problems mentioned above requires a fast, precise and accurate method. In this study, image-based face recognition was carried out by implementing the Backpropagation algorithm. The facial image dataset used in this study was sourced from the site kaggle.com with the number of faces to be trained as many as 1866 face images with various expressions for each face. In this system there are two stages, namely training all face images as a dataset to obtain the weight of each image and the next stage is testing, namely the introduction stage. The experimental results show that the application can read the pixel value of the training image with a maximum error of 0.001 network parameters, learning rates of 0.30 and 0.50. The application can perform the introduction with the best results at a learning rate of 0.50 with a recognition accuracy value of 82%.*

*Keywords: Attendance System, Data Security, Face recognition, Artificial Neural Network, Backpropagation Algorithm.*

## RIWAYAT HIDUP

Maya Dani Sari, dilahirkan di Medan pada tanggal 18 Mei 2000. Anak pertama dari tiga bersaudara yaitu pasangan dari Refizal dan Suriyati. Penulis menyelesaikan pendidikan sekolah dasar di Sekolah Dasar (SD) N 068083, Kecamatan Medan Sunggal, Kabupaten Kota Medan pada tahun 2011. Pada tahun yang sama penulis melanjutkan pendidikan Madrasah Tsanawiyah (MTs) pada Madrasah Tsanawiyah (MTs) Amal Shaleh, Kecamatan Medan Tuntungan, Kabupaten Kota Medan, selama 3 tahun penuh dan selesai pada tahun 2014. Penulis melanjutkan pendidikan selanjutnya pada Sekolah Menengah Kejuruan (SMK) Swasta Mulia Medan, pada tahun 2014 dan lulus pada tahun 2017. Pada tahun yang sama penulis kembali melanjutkan pendidikan pada perguruan tinggi swasta, tepatnya pada Universitas Medan Area (UMA) Fakultas Teknik pada program studi Informatika. Selama masa perkuliahan penulis mengikuti berbagai kegiatan seperti kegiatan Program Kreativitas Mahasiswa (PKM) selain itu penulis juga ikut dalam beberapa penelitian yang dipublikasikan dalam bentuk jurnal dan mengikuti Organisasi Himpunan Mahasiswa Informatika (HMIF). Pada tahun 2020 penulis melaksanakan kerja praktek pada Sekolah Dasar (SD) N 060922 Medan.



## KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi dengan judul “Implementasi Algoritma Backpropagation Untuk Pengenalan Wajah Berbasis Citra”. Penulis menyadari bahwa skripsi ini dapat terselesaikan berkat dorongan, motivasi, bantuan, bimbingan, arahan dan kerjasama dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, atas berkat dan rahmat-Nya skripsi ini dapat terselesaikan.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng., M.Sc., selaku Rektor Universitas Medan Area.
3. Dr. Rahmad Syah, S.Kom, M.Kom, selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Ibu Susilawati, S.Kom., M.Kom., selaku Wakil Dekan Bidang Akademik Universitas Medan Area.
5. Bapak Rizki Muliono, S.Kom., M.Kom., selaku Ketua Prodi Teknik Informatika Universitas Medan Area.
6. Ibu Nurul Khairina, S.Kom, M.Kom, selaku dosen pembimbing I yang telah memberikan banyak masukan, kritik, saran dan motivasi kepada penulis serta membimbing penulis dalam menyelesaikan tugas akhir/skripsi ini terselesaikan.
7. Orang Tua Bapak dan Ibu penulis yang telah mendukung, memberi semangat, motivasi dan banyak perhatian serta memenuhi segala kebutuhan yang dibutuhkan penulis selama masa penyusunan tugas akhir/skripsi ini.
8. Bapak/Ibu yang berada di Laboratorium STMIK Triguna Dharma yang telah memberikan waktu dan tenaganya dalam membantu saya mengumpulkan data wajah sebagai bahan untuk penelitian ini.
9. Teman-teman mahasiswa yang telah memberikan dukungan dan kebersamaan selama 4 tahun masa perkuliahan hingga saat ini.

10. Serta semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir/skripsi ini, yang namanya tidak bisa disebutkan satu persatu. Terima kasih banyak.

Sebagai manusia, penulis tidak pernah luput dari kesalahan, penulis menyadari bahwa Tugas Akhir/Skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk pengembangan selanjutnya.

Medan, 07 Juni 2023





## DAFTAR ISI

	Halaman
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	vii
RIWAYAT HIDUP .....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL .....	xiv
DAFTAR GAMBAR .....	xi
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah .....	4
<b>BAB II LANDASAN TEORI</b> .....	<b>6</b>
2.1 Pengertian Pengenalan Pola .....	6
2.2 Pengertian Citra.....	6
2.2.1 Nilai Piksel Citra Warna .....	7
2.2.2 Nilai Piksel Citra <i>Grayscale</i> .....	7
2.2.3 Proses <i>Thresholding</i> Citra.....	8
2.3 Pengolahan Citra .....	9
2.4 Syaraf Biologi .....	10
2.5 Jaringan syaraf ( <i>Neural Network</i> ) .....	11
2.6 Algoritma Backpropagation .....	13

2.6.1 Propagasi Maju (Feed Forward) .....	13
2.6.2 Propagasi Balik ( <i>Back Forward</i> ) .....	14
2.6.3 Modifikasi Bobot .....	14
2.6.4 Pelatihan Jaringan .....	14
<b>BAB III ANALISIS DAN PERANCANGAN</b> .....	<b>20</b>
3.1 Analisis Sistem .....	20
3.1 Analisa Kebutuhan Sistem .....	20
3.2 Analisa Sistem Yang Diusulkan .....	20
3.3 Metode Pengembangan Sistem .....	21
3.4.2 Proses Pelatihan <i>Backpropagation</i> .....	26
3.5.2 <i>Flowchart</i> Pengenalan Wajah .....	38
3.5.4 Perancangan <i>Form</i> Pelatihan.....	39
3.5.5 Perancangan <i>Form</i> Pengenalan .....	40
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>42</b>
4.1 Implementasi Sistem .....	42
4.1.1 Spesifikasi <i>Hardware</i> Dan <i>Software</i> .....	42
4.1.2 Dataset Yang Digunakan .....	42
4.1.3 Tampilan Menu Utama Aplikasi.....	42
4.1.4 Tampilan Parameter Pelatihan .....	43
4.1.5 Tampilan Pengenalan Wajah.....	44
4.2 Hasil .....	45
4.2.1 Pengujian.....	45
4.2.2 Tahap Preprocessing Citra .....	46
4.2.3 Pengujian.....	46
4.2.4 Pembahasan.....	50
<b>BAB V KESIMPULAN DAN SARAN</b> .....	<b>51</b>

5.1 Kesimpulan .....	51
5.2 Saran.....	51
DAFTAR PUSTAKA .....	52
LAMPIRAN – LAMPIRAN .....	54



## DAFTAR TABEL

Tabel 2.1 Perbedaan Jaringan Syaraf Biologis dengan Jaringan Syaraf Tiruan ...	11
Tabel 2.2 Penelitian terdahulu.....	18
Tabel 3.1 Nilai Piksel Citra .....	27
Tabel 3.2 Data Input Proses <i>Training Backpropagation</i> .....	27
Tabel 3.3 Biner Target .....	30
Tabel 3.4 Perhitungan Bobot Inisialisasi.....	32
Tabel 4.1 Hasil Pengenalan Percobaan $\alpha= 0.30$ , <i>Epoch</i> =9 .....	47
Tabel 4.2 Hasil Pengenalan Percobaan $\alpha= 0.50$ , <i>Epoch</i> =15 .....	48
Tabel 4.3 Hasil Pengujian .....	50



## DAFTAR GAMBAR

Gambar 2.1 Intensitas <i>Grayscale</i> (Jumadi et al., 2021) .....	8
Gambar 2.2 Citra <i>Grayscale</i> (Jumadi et al., 2021).....	8
Gambar 2.3 Citra <i>Threshold</i> (Jumadi et al., 2021).....	9
Gambar 2.4 Alur Tahap <i>Preprocessing</i> Citra menjadi <i>Threshold</i> .....	9
Gambar 2.5 Susunan <i>Neuron Biologis</i> (Puspitaningrum, 2010).....	10
Gambar 2.6 Model Sederhana Neuron (Suryo Putro S et al., 2018) .....	13
Gambar 3.1 Blok Diagram Penelitian .....	22
Gambar 3.2 Resize 30 x 15 piksel (google.com) .....	23
Gambar 3.3 Citra input dan matriks RGB.....	23
Gambar 3.4 Matriks <i>Grayscale</i> .....	24
Gambar 3.5 Matriks Biner.....	25
Gambar 3.6 Citra Hasil Binerisasi.....	26
Gambar 3.7 <i>Flowchart</i> Pelatihan .....	37
Gambar 3.8 <i>Flowchart</i> Pengenalan Wajah .....	38
Gambar 3.9 Rancangan <i>Form</i> Menu Utama .....	39
Gambar 3.10 Rancangan <i>Form</i> Pelatihan .....	40
Gambar 3.11 Rancangan <i>Form</i> Pengenalan.....	40
Gambar 4.1 Dataset Wajah.....	42
Gambar 4.2 Tampilan Menu Utama Aplikasi .....	43
Gambar 4.3 Tampilan Parameter Pelatihan.....	43
Gambar 4. 4 Tampilan Hasil Pelatihan .....	44
Gambar 4. 5 Tampilan Pengenalan .....	45
Gambar 4. 6 Citra wajah warna (a) dan <i>Grayscale</i> (b) .....	46



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Wajah ialah bagian paling penting untuk identifikasi manusia dan pusat perhatian ekspresi utama. Sebagai tanda saat berinteraksi dengan orang dan saat mengenali orang. Oleh karena itu, wajah digunakan untuk tujuan seperti pengumpulan data sistem kependudukan, keamanan, dan kehadiran berdasarkan penggunaan pengenalan wajah. Pada umumnya sistem absensi pada kantor-kantor swasta atau instansi pemerintahan, cara absensi dilakukan dengan menulis pada buku dan ada juga dengan menggunakan mesin sidik jari yang memiliki kelemahan antara lain adalah sering mengalami *human error* (kesalahan manusia) seperti memindai sidik jari sulit diverifikasi. Hal ini dapat mengakibatkan keadaan permukaan jari yang abnormal, kotor, seperti terkena air, ujung jari cacat, terlalu kering dan akhirnya sistem menolak, maka dalam menyelesaikan masalah diatas memerlukan suatu metode cepat akurat dan tepat.

Kecerdasan buatan adalah suatu bidang ilmu komputer yang memungkinkan mesin computer untuk melakukan pekerjaan mereka dengan cepat akurat dan tepat. Salah satunya adalah metode *Neural Network* (NN) atau jaringan syaraf tiruan, metode ini merupakan salah satu system pengolahan inFormasi yang dibuat untuk meniru cara kerja otak manusia dengan memecahkan masalah, yaitu mengadakan pelatihan (*training*), perubahan bobot sinaptik. NN dikonfigurasi untuk aplikasi tertentu seperti pengenalan pola atau klasifikasi/identifikasi data melalui proses pembelajaran. Pada tahap pembelajaran memasukkan pola masukan ke dalam jaringan syaraf tiruan atau output, jaringan diajarkan untuk memberikan kemungkinan jawaban yang dapat diterima.

Pada bagian penelitian tentang NN yang berjudul Sistem Absensi Menggunakan Wajah pada Jaringan Syaraf Tiruan dengan Algoritma *Learning Vector Quantization* (LVQ), penelitian ini menghasilkan hingga 20 *epoch* dan ketepatan pengenalan ciri pola wajah 50 hingga mendapatkan tingkat akurasi pengenalan pola wajah sebanyak 75% untuk 20 sampel dan 15 sampel. Dalam penelitian penulis menggunakan algoritma *learning vector quantization*,

pembelajaran dalam penelitian ini, ia menyarankan agar ia digabungkan dengan algoritma SOM Kohonen untuk output yang lebih cepat dan akurat dalam menemukan kesesuaian atau kemiripan wajah. Selain itu peneliti Kurniawan (2019) dengan judul Kombinasi *Learning Vector Quantization (LVQ)* Algoritma Artificial Neural Network dan Self Organizing Kohonen pada Kecepatan Pengenalan Pola, menerapkan gabungan jaringan syaraf tiruan. Mereka kemudian menyimpulkan bahwa kinerja komputasi mempercepat dan dapat menangani keduanya. dalam pelatihan dan pengakuan.

Metode pengenalan wajah digunakan sebagai teknologi utama di berbagai bidang. Seperti kontrol akses di gedung, kontrol akses untuk komputer pribadi dan kantor, penggunaan mesin Anjungan Tunai Mandiri (ATM) dalam bisnis sehari-hari saat menarik uang dari rekening bank, urusan dengan kantor pos, bidang investigasi kriminal dan lain sebagainya (Taghi Zadeh et al., 2019).

Teknologi komputasi saat ini semakin luas penggunaannya pada bidang Kecerdasan Buatan (*Artificial Intelligence*) khususnya pada ruang lingkup pengenalan wajah (*Face Recognition*) yang memiliki metode-metode dalam mendeteksi dan pengenalan wajah seseorang, diantaranya *Haar Cascade Classifier*, Histogram Gradien Berorientasi (HOG), Histogram Pola Biner Lokal (LBPH), Analisis Komponen Prinsip (PCA), *Eigenfaces*, *Fisherfaces*, CNN (*Convolutional Neural Network*) dan lain sebagainya.

Menurut Hafiz et al. (2018), arsitektur jaringan saraf mendalam (*Deep Neural Network*) disebut sebagai HOG-CNN untuk pengenalan wajah, tujuannya adalah pengenalan wajah secara *realtime* dengan menggunakan media *webcam* dan pengambilan wajah melalui video. Metode HOG, gambar yang ditangkap akan dikonversi menjadi skala abu-abu, untuk penghitungan nilai pada piksel gambar agar memudahkan mendeteksi semua wajah dalam bingkai gambar. HOG sangat baik untuk penangkapan batas tepi dalam gambar.

Pada penelitian Taghi Zadeh et al. (2019) yang berjudul Pengenalan emosi wajah yang cepat Menggunakan Jaringan Neural Konvolusional dan Filter Gabor dilakukan pengenalan emosi dengan cara mendeteksi wajah menggunakan algoritma Jaringan Syaraf Konvolusional dan Filter Gabor. Hasil penelitian algoritma CNN-Gabor memberikan hasil yang lebih baik dari segi kecepatan dan

akurasi dibandingkan dengan algoritma CNN konvensional. Pada percobaan proses deteksi dengan hasil akurasi yang terbaik adalah 91-92 % diperoleh dengan waktu 9 menit (541 detik) yang dirasakan terlalu lama.

Pada penelitian Prastya (2017) yang berjudul *Research on Face Detection based on fast Haar feature*, dimana dilakukan pengenalan wajah dengan algoritma deteksi wajah AdaBoost berdasarkan fitur Haar yang diimplementasikan dalam fase deteksi wajah untuk menentukan apakah wajah yang terdeteksi adalah wajah dengan mata berdasarkan proyeksi integral.

Dengan melihat permasalahan diatas serta penelitian terkait, maka pada penelitian ini dipaparkan bagaimana melakukan pengenalan wajah dimulai dengan pengolahan awal, ekstraksi ciri sampai dengan pelatihan dengan teknik jaringan syaraf tiruan dan dengan melihat penelitian diatas, maka penulis melakukan pengenalan wajah dengan menggunakan algoritma *Backpropagation* berbasis citra.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka dari itu yang menjadi rumusan masalah pada penelitian ini adalah:

1. Bagaimana menerapkan metode jaringan syaraf tiruan untuk melakukan pengenalan wajah berbasis citra.
2. Bagaimana pengaruh variasi ekspresi wajah pada tingkat akurasi dan ketepatan pengenalan wajah.
3. Bagaimana mengetahui jumlah *epoch* setiap melakukan training pada wajah.

## 1.3 Tujuan Penelitian

Adapun yang menjadi tujuan penelitian ini adalah:

1. Mengimplementasikan atau menerapkan jaringan syaraf tiruan algoritma *Backpropagation* dalam pengenalan wajah.
2. Mengetahui pengaruh variasi ekspresi wajah pada tingkat akurasi dan ketepatan pengenalan wajah.
3. Untuk mendapatkan jumlah *epoch* pada setiap training wajah untuk pengujian.

#### 1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Hasil penelitian nantinya dapat mengaplikasikan pengenalan wajah pada sistem absensi agar prosesnya dapat dilakukan dengan cepat, tepat dan akurat.
2. Menambah pengetahuan penulis dalam Jaringan Syaraf Tiruan khususnya algoritma *Backpropagation*.

#### 1.5 Batasan Masalah

Pada penelitian ini ada beberapa batasan masalah antara lain adalah sebagai berikut:

1. Input data pelatihan dan pengujian berupa citra wajah berekstensi \*.jpg berdimensi 142 x 142 piksel.
2. Citra wajah yang digunakan pada penelitian ini bersumber dari situs <https://www.kaggle.com/datasets/dataturks/face-detection-in-images>.
3. Jumlah wajah yang akan ditraining sebanyak 1866 citra wajah dan 50 citra sebagai data pengujian dengan berbagai ekspresi wajah.
4. Bagian wajah yang akan dideteksi adalah dahi, alis, mata, hidung, mulut serta dagu.
5. Wajah yang dapat terdeteksi harus memiliki pencahayaan yang baik.
6. Maksimal *error* 0.001, *learning rate* sebesar 0.30 dan 0.50.

#### 1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima (5) bagian utama sebagai berikut:

##### **Bab 1 : Pendahuluan**

Bab ini berisi latar belakang dari penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, hipotesis penelitian, batasan masalah dan sistematika penulisan.

##### **Bab 2 : Landasan Teori**

Bab ini berisi teori-teori yang diperlukan untuk memahami permasalahan yang dibahas pada penelitian ini serta teori-teori yang berhubungan Jaringan Syaraf Tiruan khususnya algoritma *Backpropagation*.

### **Bab 3 : Analisis dan Perancangan**

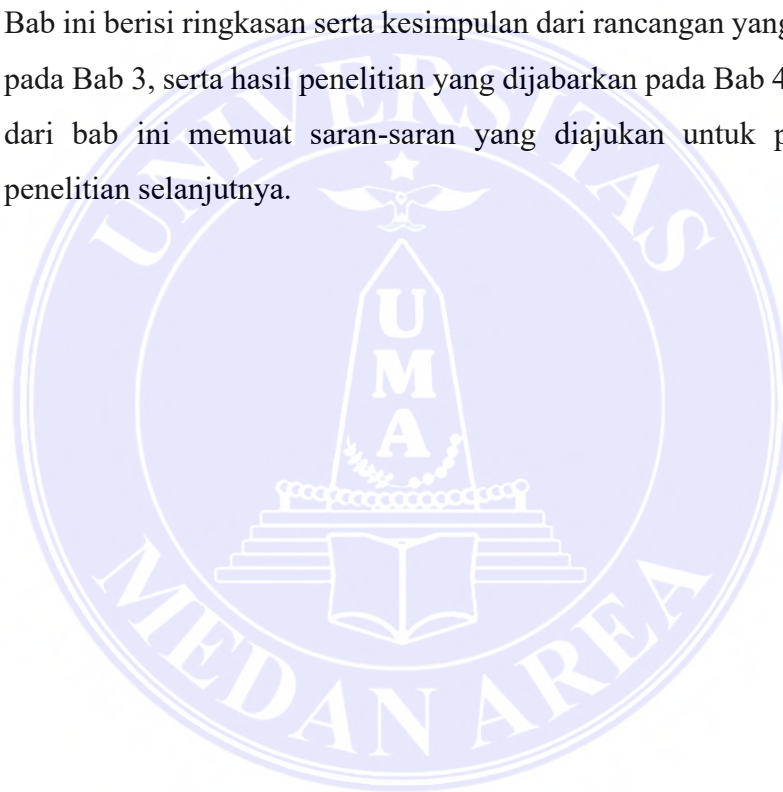
Bab ini berisi analisis arsitektur umum, tiap langkah algoritma, *flowchart*, serta penerapan algoritma *Backpropagation* dalam pengenalan wajah berbasis citra.

### **Bab 4 : Hasil dan Pembahasan**

Bab ini berisi tentang implementasi hasil perancangan yang telah dijabarkan pada Bab 3. Selain itu hasil yang didapatkan dari pengujian yang dilakukan terhadap implementasi yang dilakukan juga dijabarkan pada bab ini.

### **Bab 5 : Kesimpulan dan Saran**

Bab ini berisi ringkasan serta kesimpulan dari rancangan yang telah dibahas pada Bab 3, serta hasil penelitian yang dijabarkan pada Bab 4. Bagian akhir dari bab ini memuat saran-saran yang diajukan untuk pengembangan penelitian selanjutnya.



## BAB II

### LANDASAN TEORI

#### 2.1 Pengertian Pengenalan Pola

Proses pengenalan pola adalah suatu disiplin ilmu yang mengajarkan bagaimana mengkuantifikasi atau merepresentasikan sesuatu berdasarkan ukuran, bentuk, atau ciri-ciri utama suatu tikungan tertentu. Pola adalah entitas terdefinisi yang dapat diidentifikasi, diberi nama, dan yang dengan sendirinya dapat berupa kumpulan hasil atau monitor. Itu juga dapat dicatat dalam vektor atau notasi matematika. (Jumadi et al., 2021).

#### 2.2 Pengertian Citra

Citra adalah representasi atau gambaran dari suatu objek dari dunia nyata yang mirip dengan objek itu sendiri. Itu juga bisa berupa foto atau hasil kreatif dari sistem pemrosesan data apa pun. Ini bisa analog dalam bentuk tampilan video sederhana atau televisi analog dengan gambar. Ini digital dan dapat dimasukkan dengan mulus ke dalam media penerbitan. Ada beberapa jenis gambar dalam kehidupan kita sehari-hari, baik analog maupun digital. (Prastya, 2019).

Gambar analog adalah gambar yang berkesinambungan, seperti foto yang dicetak di atas kertas, gambar lukisan atau pemandangan yang digambar di atas kanvas atau kertas, gambar yang berkesinambungan seperti gambar di televisi, sinar-X, foto yang dicetak di atas kertas foto, lukisan, pemandangan alam. Hasil CT scan, gambar disimpan di kaset dan banyak lagi. Karena gambar analog tidak dapat ditampilkan di komputer, maka gambar analog juga tidak dapat diproses langsung di komputer.

Gambar apa pun dengan ukuran file yang dapat ditangani komputer dianggap digital. Gambar berFormat raster (grid/grid) disebut sebagai gambar digital. Piksel adalah bujur sangkar (ubin) yang memiliki koordinat  $(x, y)$ . Sumbu Y berbentuk vertikal dan terdiri dari baris (garis, baris) dan kolom (kolom) pada sumbu X (horizontal). Menunjukkan tingkat intensitas keabuan untuk piksel tertentu diwakili

oleh nilai (angka atau nilai), dan tingkat keabuan sesuai dengan nilai kode warna atau tingkat keabuan. (Atina, 2019)

### 2.2.1 Nilai Piksel Citra Warna

Bagian citra yang paling kecil adalah piksel yang memiliki nilai real. Pada citra warna terdapat 3 komponen nilai warna *red green* dan *blue* (RGB) dan untuk mendapatkan nilainya dilakukan dengan membaca setiap komponen warnanya dengan menggunakan rumusan:(Jumadi et al., 2021).

$$\text{Piksel R} = c \text{ and } 255 \dots\dots\dots (2.1)$$

$$\text{Piksel G} = (c \text{ and } 65.280)/256 \dots\dots\dots (2.2)$$

$$\text{Piksel B} = ((c \text{ and } 16.711.680)/256)/256 \dots\dots\dots (2.3)$$

$c$  = nilai intensitas piksel

Citra biner adalah citra digital yang memiliki dua nilai intensitas warna untuk setiap pikselnya, yaitu 1 dan 0, dimana value atau nilai 0 mewakili warna hitam dan nilai 1 mewakili warna putih. Gambar ini adalah gambar skala abu-abu serendah mungkin saat membuat gambar digital. Keunggulan citra biner dalam pengolahan citra digital adalah hanya memiliki 2 bit per piksel sehingga pemrosesannya cepat. Sumber citra biner adalah citra berwarna dengan rumus:

$$f(x) = (R+G+B)/3 \dots\dots\dots (2.4)$$

Nilai  $f(x)$  dilakukan *Threshold* 128, jika nilai  $f(x) <$  dari 128 maka  $f(x) = 0$  dan jika nilai  $f(x) >$  dari 128, maka  $f(x) = 1$ .

### 2.2.2 Nilai Piksel Citra *Grayscale*

Citra *Grayscale* adalah citra yang mempunyai value atau nilai *Grayscale* dan hanya berisi in*Formasi* intensitas putih dan hitam, dan citra ini terdiri dari koordinat  $x$  dan  $y$  dalam koordinat spasial dan mempunyai value atau nilai intensitas. Pada citra ini, setiap piksel memiliki nilai intensitas citra 8 bit antara 0 (hitam) dan 255 (putih) dalam citra 8 *bit*-nya (Jumadi et al., 2021). Sebuah gambar dikatakan *Grayscale* jika tidak memiliki warna RGB atau dapat kita katakan bahwa citra mempunyai value atau nilai mulai dari intensitas maksimum putih sampai dengan intensitas rendah hitam seperti terlihat pada Gambar 2.1.



Gambar 2.1 Intensitas *Grayscale* (Jumadi et al., 2021)

Sebuah citra *Grayscale* dapat dilihat seperti pada Gambar 2.2.



Gambar 2.2 Citra *Grayscale* (Jumadi et al., 2021)

Algoritma perhitungan skala abu-abu mengubah piksel dari gambar yang mengandung warna RGB (Red, Hijau, Biru) menjadi warna berbagai nuansa abu-abu (I) dengan menambahkan semua nilai warna bersama-sama dan kemudian membaginya dengan tiga sehingga bisa melakukan nilai rata-rata.

### 2.2.3 Proses *Thresholding* Citra

Proses *Thresholding* gambar digital adalah teknik yang meningkatkan nilai piksel suatu gambar dengan mengubah gambar menjadi hitam putih. Dalam *Thresholding* membutuhkan ambang batas atau patokan untuk ditetapkan sebagai batas saat mengubah elemen matriks gambar menjadi hitam atau putih. Jika nilai piksel di bawah ambang batas ini maka akan memiliki nilai piksel 0 (hitam), jika di atas ambang batas elemen akan dikonversi menjadi 1 dan untuk gambar RGB



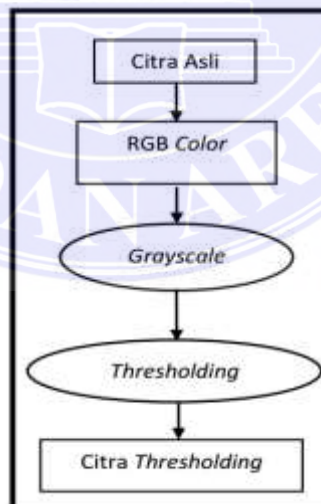
terlebih dahulu dikonversi menjadi gambar skala abu-abu (*Grayscale*) (Jumadi et al., 2021).



Gambar 2.3 Citra Hasil *Threshold* (Jumadi et al., 2021)

### 2.3 Pengolahan Citra

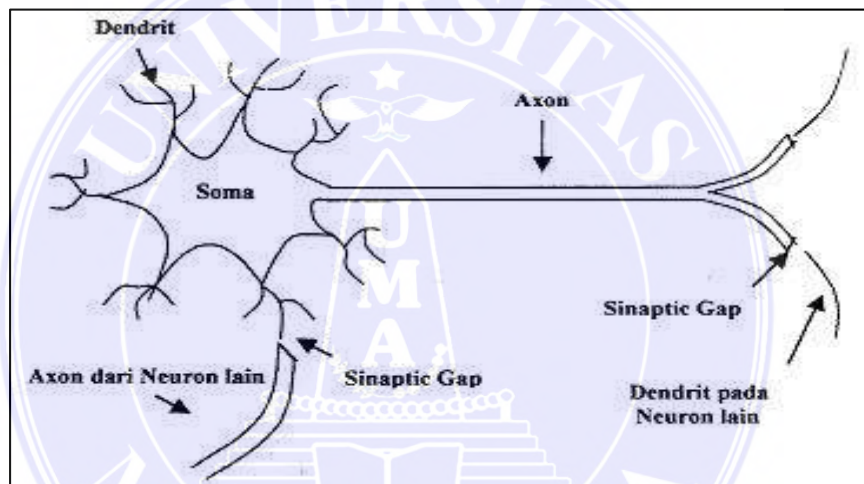
Pengolahan citra melibatkan perubahan citra dengan tiga lapisan nilai merah, hijau, dan biru menjadi citra skala abu-abu dengan 8 bit abu-abu dan terakhir citra biner dengan 2 bit yaitu hitam dan putih. Proses ini mengubah nilai piksel terkuantisasi pada citra menjadi *Threshold Grayscale* (Jumadi, J., Yupianti, Y. & Sartika, D., 2021). Tahapan pengolahan citra ditunjukkan pada Gambar 2.4.



Gambar 2.4 Tahap Pengolahan Citra

## 2.4 Syaraf Biologi

Otak manusia sangat kompleks dan mempunyai kemampuan yang luar biasa. Otak terdiri dari neuron dan koneksi yang disebut sinapsis, yang bekerja pada sinyal yang diterima dan mengirimkannya ke neuron lain. Pada manusia, terdapat 1012 neuron dan  $6 \times 10^{18}$  sinapsis, sangat banyak sehingga otak manusia dapat mengenali pola, melakukan perhitungan, dan mengontrol organ lebih cepat daripada komputer digital (Ghozali & Adikara, 2018). Jumlah dan kapasitas neuron berkembang seiring pertumbuhan fisik seseorang, terutama antara usia 0 dan 2 tahun. Selama dua tahun pertama kehidupan seseorang, sejuta sinapsis terbentuk setiap detik. Sinapsis atau neuron dan hubungannya ditunjukkan pada Gambar 2.5.



Gambar 2.5 Susunan Neuron Biologis (Ghozali & Adikara, 2018)

Di bawah ini adalah perbedaan terminologi antara jaringan saraf biologis dan tiruan seperti yang ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Perbedaan Jaringan Syaraf Biologis dengan Jaringan Syaraf Tiruan

Jaringan Syaraf Biologis	Jaringan Syaraf Tiruan
Soma	Node atau neuron
Dendrit	Input
Axon	Output
Synapse	Weight atau bobot
Kecepatan rendah	Kecepatan tinggi
Neuron banyak ( $10^9$ )	Neuron beberapa (+-100)

Neuron mempunyai tiga komponen utama yaitu dendrit, badan sel dan akson. Dendrit menerima sinyal dari neuron lain. Sinyal-sinyal ini adalah impuls listrik yang dikirim oleh proses kimia melintasi celah sinaptik. Sinyal diubah (diperkuat/dilemahkan) di celah sinaptik. Soma kemudian merangkum semua sinyal yang masuk (Chau et al., 2019). Jika jumlahnya cukup kuat dan melebihi ambang batas, sinyal ditransmisikan sepanjang akson ke sel lain. Frekuensi sinyal bervariasi dari sel ke sel. Neuron biologis adalah sistem yang "salah" dalam dua pengertian. Pertama, manusia dapat mendeteksi sinyal masukan yang sedikit berbeda dengan sinyal yang diterima sebelumnya. Misalnya, kita sering dapat mengidentifikasi orang yang pernah melihat wajahnya sendiri di sebuah foto, atau yang wajahnya terlihat sedikit berbeda karena kita sudah lama tidak melihatnya. Kedua, otak manusia dapat berfungsi meskipun beberapa neuron tidak berfungsi dengan baik. Jika sebuah neuron rusak, neuron lain dapat dilatih untuk merubah fungsi sel yang rusak tersebut (Kurniawan et al., 2019).

## 2.5 Jaringan syaraf (*Neural Network*)

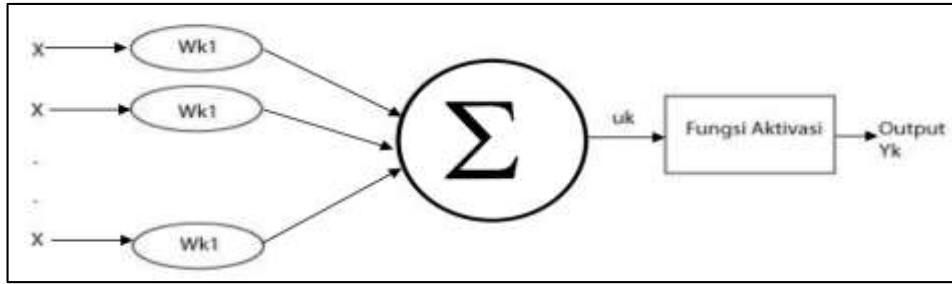
Jaringan saraf adalah jaringan yang dirancang untuk melakukan pelatihan, terinspirasi oleh sistem pembelajaran biologis yang muncul dari jaringan sel saraf (neuron) yang saling berhubungan. Adapun definisi sebuah Neural Network adalah:

1. Jaringan syaraf adalah teknik pemrosesan inFormasi berbasis komputer yang mensimulasikan dan memodelkan sistem saraf biologi.
2. Suatu model matematik yang berisi berbagai macam factor pemrosesan yang disiapkan berlapis-lapis.

3. Sistem komputer yang terdiri dari sekumpulan elemen pemrosesan sederhana yang saling berhubungan yang memproses inFormasi dari input eksternal dan merespon kondisi dinamis.
4. Jaringan syaraf tiruan adalah teknologi komputasi yang hanya didasarkan pada model saraf biologis yang mencoba mensimulasikan perilaku dan perilaku model saraf.
5. Jaringan syaraf adalah sistem pemrosesan inFormasi dengan karakteristik yang mirip dengan jaringan saraf biologis. Jaringan ini dibentuk sebagai model generalisasi matematis dari jaringan saraf biologis dan mengasumsikan sebagai berikut:
  - a. Pemrosesan inFormasi berlangsung di banyak elemen sederhana (neuron).
  - b. sinyal dikirim antara neuron melalui terminal.
  - c. Hubungan antar neuron memiliki bobot yang menambah atau mengurangi sinyal.
  - d. Setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) untuk menentukan keluarannya, yang ditentukan oleh jumlah masukan yang diterimanya. Ukuran output itu kemudian dibandingkan dengan ambang batas.

Di jaringan saraf lainnya, model adalah properti model yang mendefinisikan properti jaringan konsep dalam jaringan, yang mengacu pada koneksi beberapa neuron yang terletak di lapisan berbeda. Secara umum, lapisan jaringan saraf dibagi menjadi tiga bagian:

1. Lapis masukan (*input layer*) terdiri dari neuron yang mendapatkan data masukan dari variabel X. Semua neuron di lapis ini bisa terhubung ke neuron dilapis tersembunyi atau pribadi ke lapisan keluaran. Jika jaringan tidak menggunakan lapisan tersembunyi.
2. Lapisan tersembunyi termasuk neuron yang memperoleh inFormasi dari lapisan masuk.
3. Lapisan luaran (*output layer*) terdiri dari neuron-neuron yang menerima data langsung dari lapisan perantara atau lapisan masukan, dan yang nilai keluarannya menghasilkan nilai X ke Y yang dihitung Y.



Gambar 2.6 Model Sederhana Neuron (Suryo Putro S et al., 2018)

Secara matematis, neuron adalah fungsi yang menerima input dari level sebelumnya. Fungsi ini biasanya mengambil vektor dan mengubahnya menjadi skalar, mensintesis besaran bobot nonlinier, di mana  $K$  adalah fungsi khusus, sering disebut fungsi aktivasi, dan  $w$  adalah beban atau bobot.

## 2.6 Algoritma Backpropagation

Algoritma back-propagation adalah jenis algoritma pembelajaran terawasi yang menggunakan data input dan output untuk mendapatkan nilai bobot yang diinginkan untuk pelatihan jaringan. Proses pelatihan algoritma terdiri dari tiga langkah:

- a) *Forward propagation*, langkah dimana pola input dimulai dari input layer ke output layer dengan menggunakan fungsi aktivasi.
- b) *Backpropagation of error*, yang terjadi ketika perbedaan antara output dan target merupakan *error* yang merambat kembali pada unit-unit dari output layer..
- c) Fase untuk mengubah atau memperbaiki bobot antar lapisan.

Pada tahap ini dilakukan perubahan nilai bobot yang dipertahankan dari hasil latihan. Tahap ketiga di atas diulangi sampai diperoleh nilai kesalahan terkecil atau minimum. Hanya tahap pertama yang diperlukan saat menguji sampel data.

### 2.6.1 Propagasi Maju (Feed Forward)

Dalam propagasi sinyal input ( $x_i$ ) di-*transfer* ke lapisan tersembunyi menggunakan fungsi aktivasi yang diberikan. *Output* unit tersembunyi ( $Z_1$ ) disebarkan ke lapisan tersembunyi berikutnya menggunakan fungsi aktivasi untuk

menghasilkan *output* jaringan ( $y_k$ ). Selain itu, nilai  $y_k$  dibandingkan dengan nilai target ( $t_k$ ), selisih ( $t_k - y_k$ ) adalah nilai kesalahan (*error*), dan loop (*epoch*) berhenti jika nilai kesalahan kurang dari kesalahan maksimum, jika kesalahan maksimum lebih besar, bobot setiap baris dari jaringan diperbaiki untuk mengurangi kesalahan (Amalia et al., 2020).

### 2.6.2 Propagasi Balik (*Back Forward*)

Pada propagasi ini nilai *error* yang diperoleh ( $t_k - y_k$ ) dihitung faktor  $\delta_k$  dimana nilai  $k=1, 2, 3 \dots, m$  yang difungsikan untuk menyalurkan nilai *error* di unit  $Y_k$  ke semua unit *hidden* yang terkoneksi langsung dengan unit  $Y_k$ . Nilai  $\delta_k$  juga berfungsi sebagai pengubah nilai bobot *line* yang terkoneksi langsung ke unit *output*. Dengan cara tersebut dapat dikalkulasi faktor  $\delta_j$  pada unit di *hidden layer* dengan acuan perubahan bobot pada semua *line* dengan awal dari unit *hidden layer* pada *layer* sesudahnya. Selanjutnya semua faktor  $\delta$  pada unit *hidennya* yang terkoneksi ke unit *inputnya* dilakukan perhitungan yang sama (Taghi Zadeh et al., 2019).

### 2.6.3 Modifikasi Bobot

Pada perubahan bobot, semua jalur *neuron* ditetapkan pada waktu yang sama, dan perubahan bobot jalur neuron dihitung berdasarkan nilai faktor  $\delta$  neuron dari lapisan di atasnya. Memodifikasi bobot garis yang terhubung langsung ke lapisan keluaran berdasarkan lapisan yang berakhir di unit keluaran. Ulangi proses di atas. Kondisi berhenti yaitu jumlah maksimum *epoch* atau kesalahan terpenuhi (Anmol & Vinay, 2018.).

### 2.6.4 Pelatihan Jaringan

Algoritma *backpropagation*, jaringan *feedforward* atau umpan maju, menghitung bobot di akhir pelatihan untuk mendapatkan bobot terbaik, dan secara iteratif menyusun ulang bobot selama proses untuk meminimalkan kesalahan yang diperkenalkan (kesalahan). Nilai *error* dihitung berdasarkan *mean squared error* (MSE). Kesalahan kuadrat rata-rata juga digunakan sebagai dasar untuk

menghitung kinerja fungsi aktivasi. Sebagian besar pelatihan jaringan *feedforward* menggunakan gradien fungsi aktivasi untuk menentukan cara menyesuaikan bobot guna meminimalkan kinerja. Gradien ini ditentukan menggunakan teknik yang disebut *Backpropagation* (Kurniawan et al., 2019).

Pelatihan standar untuk algoritma ini adalah menyetel nilai dalam arah gradien negatif dengan menyetel bobot jaringan ke arah di mana fungsi aktivasi meluruh dengan cepat. Berikut langkah-langkah pelatihannya:

- Step 0 : Inisialisasi bobot awal koneksi antar neuron menggunakan angka acak kecil misalnya (-0.4 hingga +0.4, -0.5 hingga +0.5, dan -1 hingga +1)..
- Step 1 : Lakukan langkah 2 sampai 9 kecuali kondisi berhenti yang ditentukan terpenuhi.
- Step 2 : Selesaikan langkah 3 sampai 8 untuk setiap pasangan pelatihan.

**Propagasi Maju:**

- Step 3 : Setiap unit input ( $x_i, i=1, \dots, n$ ) menerima sinyal input  $x_i$  dan mengirimkannya ke semua unit di lapisan tersembunyi.
- Step 4 : Bobot total setiap unit tersembunyi ( $x_i, I = 1, \dots, p$ ) sinyal input:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.5)$$

$v_{0j}$  = terapkan fungsi aktivasi bias unit tersembunyi  $j$  untuk mengilangkan sinyal keluaran  $z_j = f(z\_in_j)$  dan menerapkan sinyal ini ke semua unit dilapisan atas (unit keluaran).

- Step 5 : setiap unit keluaran ( $y_k, k = 1, \dots, m$ ) jumlahkan bobot sinyal masukannya:

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots (2.6)$$

$w_{0k}$  = bias pada unit keluaran  $k$  dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,  $y_k = f(y\_in_k)$

**Propagasi Balik:**

Step 6 : tiap unit keluaran ( $y_k$ ,  $k = 1, \dots, m$ ) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan *inFormasinya*,

$$\delta_k = (t_k - y_k) f'(y\_in_k) \dots\dots\dots (2.7)$$

hitung koreksi bobotnya (digunakan untuk mempengaruhi  $w_{jk}$  nantinya),

$$\Delta w_{jk} = \alpha \delta_k z_j \dots\dots\dots (2.8)$$

hitung koreksi biasnya (digunakan untuk mempengaruhi  $w_{ok}$  nantinya)

$$\Delta w_{ok} = \alpha \delta_k \dots\dots\dots (2.9)$$

dan kirimkan  $\delta_k$  ke unit-unit pada lapisan dibawahnya,

Step 7 : Setiap unit lapisan tersembunyi ( $z_j$ ,  $j = 1, \dots, p$ ) jumlah hasil perubahan masukannya (dari unit-unit lapisan diatasnya),

$$\delta\_in_j = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (2.10)$$

kalikan dengan turunan fungsi aktivasinya untuk menghitung *inFormasi* kesalahannya,

$$\delta_j = \delta\_in_j f'(z\_in_j) \dots\dots\dots (2.11)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \dots\dots\dots (2.12)$$

hitung koreksi bias

$$\Delta v_{oj} = \alpha \delta_j \dots\dots\dots (2.13)$$

Step 8 : Update bobot dan bias pada hubungan antar lapisan

$$w_{jk} (baru) = w_{jk} (lama) + \Delta w_{jk} \dots\dots\dots (2.14)$$

$$v_{ij} (baru) = v_{ij} (lama) + \Delta v_{ij} \dots\dots\dots (2.15)$$

Step 9 : Tes kondisi terhenti

*Backpropagation* pada dasarnya terdiri dari dua tahap, yaitu fase maju dan fase mundur. Fase maju mengharapkan ini menjadi nilai input untuk mendapatkan output yang diharapkan. Untuk menghasilkan pola, kami merangkum bobot input



dan memetakannya ke fungsi aktivasi jaringan. *Output*-nya dapat dihitung sebagai berikut (Tandrian & Kusnadi, 2019):

$$o_j = \frac{1}{1+e^{-a_{netj}}} \dots\dots\dots(2.16)$$

di mana,

$$a_{net.j} = (\sum_{i=1} w_{ij}o_i) + \theta_j \dots\dots\dots (2.17)$$

dimana,

$o_j$  : *input* dari  $j$  unit

$w_{ij}$  : bobot yang dihubungkan dari unit  $i$  ke unit  $j$

$a_{net,j}$  : jaringan keluaran untuk  $j$  unit

$\theta_j$  : bias untuk  $j$  unit

Pada fase inversi, pola keluaran (*output* aktual) dibandingkan dengan keluaran dan sinyal kesalahan untuk setiap keluaran dihitung. Sinyal tersebut kemudian disebarkan kembali dari lapisan keluaran ke setiap unit di lapisan transisi yang secara langsung berkontribusi pada keluaran, bobot disesuaikan selama proses pembelajaran, dan kesalahan selanjutnya diminimalkan selama penurunan terarah.

Fungsi kesalahan neuron output ditulis sebagai:

$$E = \frac{1}{2} \sum_{k=1}^n (t_k - o_k)^2 \dots\dots\dots (2.18)$$

dimana,

$n$  : angka pada modul keluaran didalam lapisan *output*

$t_k$  : keluaran yang dikendaki dari keluaran unit  $k$

$o_k$  : keluaran jaringan dari keluaran unit  $k$

Parameter  $\alpha$  adalah kecepatan pemahaman yang menentukan kecepatan iterasi. Nilai  $\alpha$  antara 0 dan 1 ( $0 \leq \alpha \leq 1$ ). Semakin tinggi nilai  $\alpha$ , semakin sedikit iterasi yang digunakan. Namun, jika harga  $\alpha$  terlalu tinggi, pola yang benar akan rusak, sehingga pemahaman akan lambat. Siklus pelatihan yang berisi semua pola disebut *epoch* dan nilai  $\alpha$  yang digunakan dalam penelitian ini adalah 0,3. Nilai awal dari bobot memiliki pengaruh yang kuat pada apakah jaringan saraf tiruan mencapai minimum global atau minimum lokal kesalahan atau *error*, dan proses bergerak cepat menuju konvergensi. Jika bobot awal terlalu besar, input ke lapisan tersembunyi atau output akan muat di wilayah di mana turunan dari fungsi sigmoid sangat kecil. Jika bobot awal terlalu kecil, input ke setiap lapisan tersembunyi atau output akan terlalu kecil. Hal ini membuat proses pelatihan sangat lambat. Bobot

awal biasanya diinisialisasi secara acak dengan nilai antara -0.5 dan 0.5. Setelah ini selesai, jaringan dapat digunakan untuk pelatihan pengenalan pola (Tandrian & Kusnadi, 2019)

## 2.2 Penelitian Terdahulu

Berikut adalah beberapa rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini seperti yang terlihat pada Tabel 2.2.

Tabel 2.2 Penelitian terdahulu

No	Peneliti	Judul Penelitian	Metode	Keterangan
1	Zadeh <i>et al</i> (2019)	Pengenalan emosi Wajah Cepat Menggunakan Jaringan Syaraf Konvolusional dan Filter Gabor	Filter Gabor	Pada percobaan proses deteksi dengan hasil akurasi yang terbaik adalah 91-92 % diperoleh dengan waktu 9 menit (541 detik).
2	Wang <i>et al</i> , (2017)	Deteksi Wajah berdasarkan fitur Haar cepat,	AdaBoost	Dilakukan pengenalan wajah dengan algoritma deteksi wajah AdaBoost berdasarkan fitur Haar
3	Enriquez K., (2018)	Deteksi wajah yang lebih cepat menggunakan algoritma <i>Convolutional Neural Networks &amp; Viola-Jones</i>	<i>Convolutional Neural Networks &amp; Viola-Jones</i>	Dilakukan pendeteksian wajah dengan algoritma Viola-Jones dan <i>Convolutional Neural Networks (CNN)</i>
4	Prastya, M. A. 2019.	Sistem Pengenalan Wajah Manusia Menggunakan Algoritma Viola-Jones Dan	Algoritma Viola-Jones Dan Principal Component Analysis	Hasil pengujian. tingkat akurasi pengenalan wajah mencapai 87,5%

		Principal Component Analysis		
5.	Subiyanto, Priliyana, D., Riyadani, M. E., Iksan, N. & Wibawanto, H. 2020.	Sistem Pengenalan Wajah Dengan Algoritma PCA-GA Untuk Keamanan Pintu Rumah Pintar Menggunakan Rasberry Pi	Algoritma PCA-Genetika	metode PCA-GA memiliki tingkat akurasi yang besar, yaitu 90 %, dibandingkan dengan algoritma PCA dan LBPH-GA dengan akurasi 80 % dan 90 %.
6.	Kurniawan, E., Wibawanto, H. & Widodo, D. A. 2019.	Implementasi Metode Backpropogation Dengan Inisialisasi Bobot Nguyen Widrow Untuk Peramalan Harga Saham	Backpropogation Dengan Inisialisasi Bobot Nguyen Widrow	Hasil peramalan untuk harga <i>close</i> saham BBKA.JK memiliki nilai MAPE 0,85% dan untuk harga <i>close</i> saham AALI.JK memiliki nilai MAPE sebesar 1,84%.
7.	Mahfuzh, H. F., Widiyanto, D. & Chamidah, N. 2020.	Pengaruh Algoritma Inisialisasi Nguyen-Widrow Terhadap Algoritma Backpropagation Dalam Prediksi Indeks Harga Konsumen (IHK)	Algoritma Inisialisasi Nguyen-Widrow	Inisialisasi Nguyen-Widrow memiliki nilai MSE yaitu 0,000115 dan nilai koefisien determinasi sebesar 0,99778, inisialisasi random 0,000143 dan koefisien determinasi sebesar 0,99445.

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Analisis Sistem

Pada penelitian ini, pengenalan wajah dilakukan dengan menggunakan data wajah berbasis citra. Bab ini membahas dua tahapan, yaitu tahapan analisis dan tahapan perancangan sistem. Pada tahap analisis, data yang digunakan untuk mengolah analisis dan analisis data yang digunakan pada setiap tahap dilakukan. Pada tahap desain sistem, kami mempertimbangkan untuk merancang antarmuka sistem untuk pengguna. Adapun dataset wajah pada penelitian ini adalah berupa *file* citra wajah berekstensi \*.jpg yang berdimensi 142 x 142 piksel. Citra wajah yang digunakan bersumber dari situs <https://www.kaggle.com/datasets/dataturks/face-detection-in-images> dimana jumlah citra wajah yang akan dilatih sebanyak 1916 citra wajah dengan berbagai ekspresi wajah.

#### 3.1 Analisa Kebutuhan Sistem

Dengan sistem yang berjalan saat ini, terdapat beberapa hal yang harus dipenuhi yaitu diperlukan adanya sistem yang mampu dalam pengenalan wajah berbasis citra yang selama ini sudah banyak dilakukan dengan tingkat akurasi yang bervariasi tergantung citra wajah termasuk untuk citra wajah tanpa ekspresi atau emosi maupun atribut yang dipakai seperti kaca mata serta topi. Sedang untuk pengenalan wajah dengan bermacam-macam ekspresi emosi tingkat akurasinya masih rendah.

#### 3.2 Analisa Sistem Yang Diusulkan

Sistem pengenalan wajah berbasis citra pada penelitian ini dirancang untuk sistem yang berbasis desktop, dengan menggunakan *software* MATLAB dan dimana untuk membangun sistem ini yang terdiri dari 2 halaman dimana pada halaman awal terdiri dari menu Training. Dalam penggunaan sistem, pengguna terlebih dahulu diharuskan untuk memasukan data training dimana data ini terdiri dari data-data citra wajah yang akan dilatih oleh sistem untuk memperoleh bobot

akhir. Setelah citra wajah dilatih sistem, maka dapat dilakukan pengujian pada proses pengenalan wajah yang belum diketahui.

### 3.3 Metode Pengembangan Sistem

Metode pengembangan yang digunakan dalam penelitian ini adalah *Software Development Life Cycle* (SDLC), perangkat lunak mengikuti siklus hidup yang jelas yang mencakup semua aspek produk perangkat lunak dari awal hingga akhir. Siklus hidup pengembangan perangkat lunak (SDLC) adalah kerangka kerja pengenalan wajah yang akan digunakan untuk pengembangan perangkat lunak. SDLC adalah pendekatan sistematis untuk pengembangan sistem yang efisien tetapi tanpa pengujian itu tidak mungkin. Karena SDLC memberi tahu proses pengembangan sistem untuk meningkatkan kualitas tetapi tidak membantu dalam menemukan cacat sistem. *Software Development Life Cycle* (SDLC) adalah strategi membangun atau memelihara sistem perangkat lunak.

### 3.4 Tahapan Penelitian

Tahapan penelitian berisi langkah-langkah dilakukannya dimulai dengan identifikasi permasalahan hingga hasil dan evaluasi hasil penelitian. Adapun langkah-langkah dari penelitian pengenalan wajah dengan algoritma *Backpropagation* terdiri dua proses, yang pertama yaitu proses pelatihan dan kedua adalah proses pengujian. Pada diagram pelatihan dijabarkan alur proses untuk melakukan penghitungan bobot akhir setiap citra wajah, sedangkan diagram pengujian adalah proses untuk melakukan pengenalan wajah dari citra inputan berdasarkan nilai bobot akhirnya. Hasil akhir adalah dilakukan perhitungan nilai akurasi pengenalan citra wajah serta kesimpulan nilai akurasinya. Pada penelitian ini dilakukan pengenalan wajah dengan algoritma *Backpropagation* dengan langkah-langkah penelitian ini adalah sebagai berikut:

1. Preprosesing citra wajah

Pada proses ini terdiri dari *resize*, *Grayscale* serta binerisasi citra dengan metode *Threshold*.

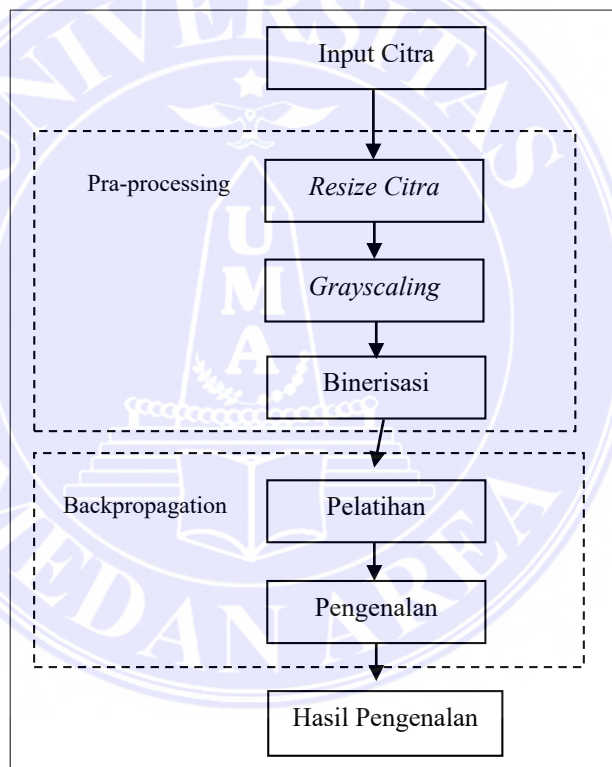
2. Pelatihan citra wajah

Pada pelatihan ini dilakukan perhitungan bobot setiap citra wajah yang terdiri 1866 citra.

3. Pengenalan wajah

Pada proses ini dilakukan perhitungan jarak bobot citra pengujian dengan setiap bobot citra pelatihan.

Diagram Penelitian Pengenalan Wajah Metode yang diusulkan dalam penelitian ini ditunjukkan pada Gambar 3.1. dengan langkah sebagai berikut yaitu menginput file gambar, mengubah gambar berwarna menjadi gambar abu-abu kemudian menjadi gambar biner.



Gambar 3.1 Diagram Penelitian

Dari Gambar 3.1 diatas dapat dijelaskan sebagai berikut:

1. Input file citra adalah pemilihan file citra wajah.
2. *Resize* citra adalah menormalkan ukuran dimensi citra (lebar x tinggi) agar sesuai dengan sistem.
3. *Grayscale* citra adalah perubahan nilai piksel citra warna menjadi keabuan.

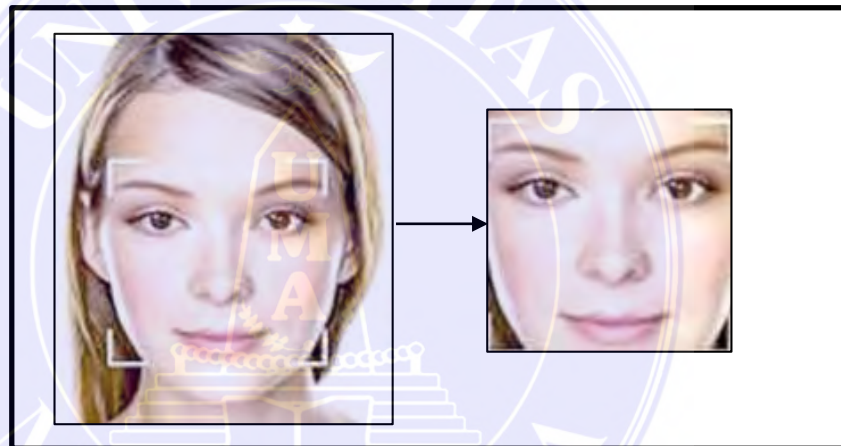
4. Binerisasi adalah perubahan citra keabuan menjadi satu warna hitam atau putih.
5. Proses pelatihan citra wajah dengan algoritma *Backpropagation*.
6. Proses pengenalan wajah.

### 3.4.1 Representasi Citra

Tahap awal yang akan dilakukan sebelum melakukan ekstraksi ciri pada citra adalah pemrosesan citra yaitu:

#### 1. *Resize citra*

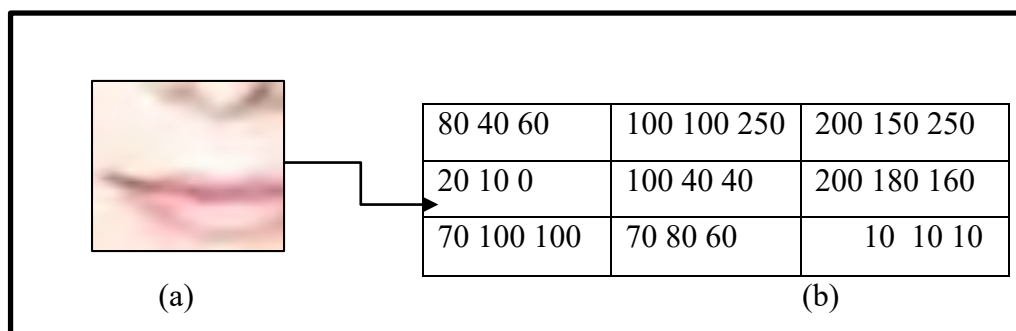
Resize citra adalah penyeragaman ukuran citra menjadi ukuran 142 x 142 piksel seperti pada Gambar 3.2.



Gambar 3.2 Resize 142 x 142 piksel (google.com)

#### 2. Proses Citra Keabuan (*Grayscale*)

Gambar yang dipotong di atas diubah menjadi mode warna abu-abu. Misalnya nilai RGB dari citra input dalam matriks 3 x 3 piksel, seperti terlihat pada Gambar 3.3.



Gambar 3.3 Contoh Nilai RGB piksel Citra Wajah

Pada Gambar 3.3 diatas, nilai RGB per piksel citra akan diubah ke nilai keabu-abuan dengan persamaan berikut:

$$f_0(x,y) = \left( \frac{f^R(x,y) + f^G(x,y) + f^B(x,y)}{3} \right) \dots\dots\dots (3.1)$$

dimana  $f^R$  : nilai komponen *red*  
 $f^G$  : nilai komponen *green*  
 $f^B$  : nilai komponen *blue*

Hasil nilai keabu-abuan citra adalah:

- Piksel 1 : 80 40 60 = (80+40+60)/3 = 60
- 2 : 100 100 250 = (100+100+250)/3 = 150
- 3 : 200 150 250 = (200+150+250)/3 = 200
- 4 : 20 10 0 = (20+10+0)/3 = 10
- 5 : 100 40 40 = (100+40+40)/3 = 60
- 6 : 200 180 160 = (200+180+160)/3 = 180
- 7 : 70 100 100 = (70+100+100)/3 = 90
- 8 : 70 80 60 = (70+80+60)/3 = 70
- 9 : 10 10 10 = (10+10+10)/3 = 10

Tahap selanjutnya adalah dimana nilai keabu-abuan citra tadi diplotkan pada matriks citra *Grayscale* seperti pada Gambar 3.4.

60	150	200
10	60	180
90	70	10

Gambar 3.4 Matriks *Grayscale*

### 3. Proses citra hitam-putih (Binerisasi)

Proses citra ini adalah mengubah nilai pixel intensity menjadi 0 atau 1 saja. 0 berwarna hitam dan 1 berwarna putih. Nilai keabuan (*Grayscale*)matriks pada Gambar 3.4 dikonversi ke biner menggunakan ambang batas dari rumus:

$$Th = \frac{f_{maks} + f_{Min}}{2} \dots\dots\dots (3.2)$$

Pada citra diatas, nilai ambang piksel diperoleh dengan persamaan (3.2) adalah:



$$Th = \frac{200+10}{2} = 105$$

Diperoleh nilai ambang sebesar 105 yang akan dapat dipakai jika persentase nilai piksel lebih besar dari 105 adalah lebih besar dari 50 %, jika persentase nilai piksel lebih besar dari 105 adalah lebih kecil dari 50 %, maka dipakai persamaan:

$$Th = \frac{fBitMaks}{2} = 256:2 = 128 \dots\dots\dots (3.3)$$

Pada matriks citra dapat diperoleh persentase nilai piksel lebih besar dari 105 adalah:  $(2:9) * 100 \% = 22.22 \%$ , maka dipakai angka nilai ambang pada persamaan (3.3) menghasilkan nilai biner sebagai berikut.

60	150	200
10	60	180
90	70	10

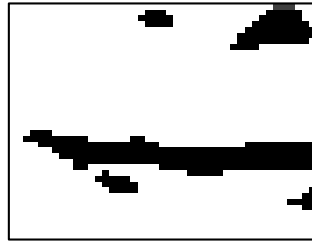
- Nilai Piksel 1 = 60, P1 lebih kecil dari 128 maka nilai P1 bernilai 0.
- 2 = 150, P2 lebih besar dari 128 maka nilai P2 bernilai 1.
- 3 = 200, P3 lebih besar dari 128 maka nilai P3 bernilai 1.
- 4 = 10, P4 lebih kecil dari 128 maka nilai P4 bernilai 0.
- 5 = 60, P5 lebih kecil dari 128 maka nilai P5 bernilai 0.
- 6 = 180, P6 lebih besar dari 128 maka nilai P6 bernilai 1.
- 7 = 90, P7 lebih kecil dari 128 maka nilai P7 bernilai 0.
- 8 = 70, P8 lebih kecil dari 128 maka nilai P8 bernilai 0.
- 9 = 10, P9 lebih besar dari 128 maka nilai P9 bernilai 0.

Untuk selanjutnya semua nilai biner hasil *Threshold* diatas dimasukkan ke dalam matriks citra biner seperti pada Gambar 3.5.

0	1	1
0	0	1
0	0	0

Gambar 3.5 Matriks Biner

Dari matriks biner diatas, maka diperoleh representasi wajah mode warna biner seperti pada Gambar 3.6.



Gambar 3.6 Citra Biner

### 3.4.2 Proses Pelatihan *Backpropagation*

Proses pelatihan citra wajah dilakukan dengan menggunakan parameter jaringan berupa jumlah neuron sebagai berikut:

1. Pada *input layer* adalah sejumlah piksel citra wajah yang dipilih.
2. Pada *hidden layer* yaitu 3.
3. Pada *output layer* yaitu 10 sejumlah wajah yang akan dikenali.

Untuk setiap neuron dalam jaringan, ia menjumlahkan masukan yang diproses oleh fungsi transfer sigmoid dari lapisan tersembunyi dan lapisan keluaran dikalikan dengan bobot aslinya. Jumlah neuron masukan untuk setiap wajah, 3 neuron pada lapisan tersembunyi dan 10 neuron pada lapisan keluaran. Untuk setiap neuron pada *hidden layer* 1, outputnya dapat dihitung dengan persamaan (3.4) dimana  $X_i$  adalah nilai dari variabel input,  $W_{ji}$  adalah bobot antara input layer dan *hidden layer*, sedangkan  $W_{j0}$  adalah nilai bias.

$$X_k = \tan \text{sig} (\sum_{k=1}^n X_j W_{kj} + W_{k0}) \dots\dots\dots (3.4)$$

Selanjutnya pada setiap neuron di *hidden layer* akan dihitung *output*-nya dengan memakai persamaan (3.5), dimana  $X_j$  adalah nilai dari neuron ke- $j$  pada hidden layer,  $W_{kj}$  dan  $W_{k0}$  adalah nilai bias. Variabel  $n$  adalah banyaknya neuron pada *hidden layer* yaitu 10.

$$X_j = \tan \text{sig} (\sum_{i=1}^m X_i W_{ji} + W_{j0}) \dots\dots\dots (3.5)$$

Untuk mendapatkan nilai output dengan memakai persamaan (3.6) dimana  $X_k$  adalah nilai dari neuron ke- $k$  pada *hidden layer* dan output layer, sedangkan  $W_{l0}$  adalah nilai bias.

$$Y' = \tan \text{sig} (\sum_{l=1}^0 X_k W_{lk} + W_{l0}) \dots\dots\dots (3.6)$$

Setelah proses inisialisasi selesai, maka dihitung total error dengan persamaan 3.7.

$$E = \frac{1}{2} \sum_{p=1}^n (y_p - y'_p) \dots\dots\dots (3.7)$$

Proses pelatihan memakai 2 alur perhitungan yaitu alur maju dan alur mundur.

- a. Dari hasil praprosesing citra setiap citra wajah terdiri neuron yang menjadi masukan dengan target sesuai jumlah wajah pada citra dataset tersebut. Sebagai contoh nilai piksel wajah seperti pada Tabel 3.1.

Tabel 3.1 Nilai Piksel Citra

1	0	1	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
1	0	1	1	1
0	1	0	0	0
1	0	1	0	1

Data pada Tabel 3.1 adalah merupakan vektor input citra wajah dan semua vektor citra wajah dimasukkan vektor seperti pada Tabel 3.2.

Tabel 3.2 Data Input Proses Training *Backpropagation*

No	Vektor Input					Vektor Output	Nama
1	1	0	1	0	1	A	Citra-A
	1	1	0	1	0		
	0	0	1	0	1		
	1	0	1	0	1		
	0	1	0	1	0		
	1	0	1	0	1		
	1	0	1	1	1		
	0	1	0	0	0		
	1	0	1	0	1		
2	1	1	1	0	1	B	Citra -B
	1	0	0	1	0		
	0	1	1	0	1		
	1	1	1	0	1		
	0	0	1	1	0		
	1	0	0	0	0		
	0	0	0	1	1		
	1	1	1	0	0		

	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>		
3	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	C	Citra -C
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
4	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	D	Citra -D
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>		
5	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	E	Citra -E
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>		
6	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	F	Citra -F
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>		
	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		Citra -G
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		
	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>		
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>		

7	0	0	1	0	1	G	
	0	0	1	1	1		
	0	1	0	0	0		
	1	1	1	0	0		
8	0	1	0	0	0	H	Citra -H
	1	1	0	1	0		
	1	0	1	1	1		
	1	1	1	0	1		
	1	1	0	1	0		
	0	0	1	0	1		
	0	0	1	1	1		
	0	1	0	0	0		
	1	1	1	0	0		
9	0	1	0	0	1	I	Citra -I
	1	1	0	1	0		
	1	0	1	1	1		
	1	1	1	0	1		
	1	1	0	1	0		
	0	0	1	0	1		
	0	0	1	1	1		
	0	1	0	0	0		
	1	1	1	0	0		
10	0	1	1	1	1	J	Citra -J
	1	1	0	1	0		
	1	0	1	1	1		
	1	1	1	0	1		
	1	1	0	1	0		
	0	0	1	0	1		
	0	0	1	1	1		
	0	1	0	0	0		
	1	1	1	0	0		

b. Selanjutnya input nilai biner pada baris pertama untuk setiap matrik target input seperti pada Tabel 3.3.

Tabel 3.3 Biner Target

No	Citra	Target Input	Biner
1	Citra -A	A	10101
2	Citra -B	B	11101
3	Citra -C	C	10000
4	Citra -D	D	10111
5	Citra -E	E	00101
6	Citra -F	F	00110
7	Citra -G	G	00111
8	Citra -H	H	01000
9	Citra -I	I	01001
10	Citra -J	J	01111

c. Tentukan jumlah *epoch* atau iterasi yang akan digunakan, nilai *error* minimal serta rasio pembelajaran ( $\alpha$ ). Selanjutnya berikut contoh perhitungan dengan parameter sebagai berikut:

1. Minimal *error* adalah 0.001.
2. Nilai pembelajaran adalah 0,30 dan 0.50.

d. Bobot awal acak.

1. Bobot awal input ke *hidden* ( $v$ ) adalah nilai nilai acak

$$V_i = [0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01]$$

Hitung faktor skala ( $\beta$ )

$$\beta = 0.7(p)^{1/n}$$

$n$  = jumlah unit masukan = 45

$p$  = jumlah unit tersembunyi = 3.

$$\beta = 0,7 (3)^{1/45}$$

$$\beta = 0,7 (3)^{0,022}$$

$$\beta = 0.717$$

Jadi batasan nilai bias yang dipakai adalah faktor skala yang merupakan bilangan acak antara -0.717 hingga 0.717

2. Bobot awal bias ke *hidden* ( $v_0$ )

$$V_1 = [0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01]$$

$$V_2 = [0.02 \ 0.02 \ 0.02 \ 0.02 \ 0.02]$$

$$V_3 = [0.03 \ 0.03 \ 0.03 \ 0.03 \ 0.03]$$

$$\begin{aligned} \|V_1\| &= \sqrt{0.01^2 + 0.01^2 + 0.01^2 + 0.01^2 + 0.01^2} = \\ &= \sqrt{0.001 + 0.001 + 0.001 + 0.001 + 0.001} \\ &= 0.022 \end{aligned}$$

$$\begin{aligned} \|V_2\| &= \sqrt{0.02^2 + 0.02^2 + 0.02^2 + 0.02^2 + 0.02^2} = \\ &= \sqrt{0.0004 + 0.0004 + 0.0004 + 0.0004 + 0.0004} \\ &= 0.044 \end{aligned}$$

$$\begin{aligned} \|V_3\| &= \sqrt{0.03^2 + 0.03^2 + 0.03^2 + 0.03^2 + 0.03^2} = \\ &= \sqrt{0.009 + 0.009 + 0.009 + 0.009 + 0.009} \\ &= 0.067 \end{aligned}$$

Berikut bobot inisialisasi merupakan bobot yang dipakai sebagai inisialisasi dengan rumus:

$$\|V_j\| = \frac{w_{vij}(lama)}{\|V_j\|} \dots\dots\dots (3.8)$$

$$\|V_{11}\| = \frac{0.717 \cdot 0.07}{\|0.022\|} = 0.0554$$

$$\|V_{12}\| = \frac{0.717 \cdot 0.97}{\|0.022\|} = 0.04521$$

$$\|V_{13}\| = \frac{0.717 \cdot 0.45}{\|0.022\|} = 0.03326$$

$$\|V_{14}\| = \frac{0.717 \cdot 0.39}{\|0.022\|} = 0.02845$$

$$\|V_{15}\| = \frac{0.717 \cdot 0.22}{\|0.022\|} = 0.02250$$

$$\|V_{21}\| = \frac{0.717 \cdot 0.02}{\|0.044\|} = 0.0965$$

$$\|V_{22}\| = \frac{0.717 \cdot 0.45}{\|0.044\|} = 0.0523$$

$$\|V_{23}\| = \frac{0.717 \cdot 0.32}{\|0.044\|} = 0.0215$$

$$\|V_{24}\| = \frac{0.717 \cdot 0.21}{\|0.044\|} = 0.0522$$

$$\|V_{25}\| = \frac{0.717 \cdot 0.10}{\|0.044\|} = 0.0441$$

$$\|V_{31}\| = \frac{0.717 \cdot 0.04}{\|0.067\|} = 0.0254$$

$$\|V_{32}\| = \frac{0.717 \cdot 0.32}{\|0.067\|} = 0.0369$$

$$\|V_{33}\| = \frac{0.717 \cdot 0.56}{\|0.067\|} = 0.0657$$

$$\|V_{34}\| = \frac{0.717 \cdot 0.56}{\|0.067\|} = 0.0554$$

$$\|V_{35}\| = \frac{0.717 \cdot 0.32}{\|0.067\|} = 0.0541$$

Perhitungan Bobot Inisialisasi diatas dimasukkan seperti pada Tabel 3.4.

Tabel 3.4 Perhitungan Bobot Inisialisasi

	Z1	Z2	Z3
X1	0.05540	0.04521	0.03326
X2	0.02845	0.02250	0.0965
X3	0.0523	0.0215	0.0522
X4	0.0441	0.0254	0.0369
X5	0.0657	0.0554	0.0541
1	-1	1	-1

e. Proses perhitungan dilakukan pada saat nilai iterasi masih di bawah jumlah siklus yang ditentukan dan tingkat kesalahan lebih besar (>) dari tingkat kesalahan minimum, dengan proses langkah e hingga g diulang untuk setiap data.

f. Alur maju:

Data = *xI* Citra -A 10101

*Epoch* = 1

Operasi pada *hidden layer* :

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_{in1} = 0 + (0.0554 \cdot 1) + (0.02845 \cdot 0) + (0.0523 \cdot 1) + (0.0441 \cdot 0) + (0.0657 \cdot 1)$$

$$z_{in1} = 0.0657$$

$$z_{in2} = 0 + (0.04521 \cdot 1) + (0.02250 \cdot 0) + (0.0215 \cdot 1) + (0.0254 \cdot 0) + (0.0554 \cdot 1)$$

$$z_{in2} = 0.0554$$

$$z_{in3} = 0 + (0.03326 \cdot 1) + (0.0965 \cdot 0) + (0.0522 \cdot 1) + (0.0369 \cdot 0) + (0.0541 \cdot 1)$$

$$z_{in3} = 0.0541$$



$$Z_1 = \frac{1}{1 + e^{-0.0657}} = 0.02514$$

$$Z_2 = \frac{1}{1 + e^{-0.0554}} = 0.03254$$

$$Z_3 = \frac{1}{1 + e^{-0.0541}} = 0.05412$$

Operasi pada *output layer*

$$y_{ink} = w_0 + \sum_{i=1}^n w_{jk} * z_j$$

$$y_k = \frac{1}{1 + e^{-y_{in_k}}}$$

$$\begin{aligned} y_{ink} &= 0 + (0.022 * 0.02514) + (0.044 * 0.03254) + (0.067 * 0.05412) \\ &= 0.1330 \end{aligned}$$

$$y_k = \frac{1}{1 + e^{-0.1330}} = 0.05244$$

Hitung nilai eror :

$$error = T_{ik} - y_k$$

$$error = 1 - 0.05244 = 0.9475$$

g. Alur mundur

Hitung faktor unit kesalahan ( $\delta$ ) :

$$\delta_k = (T_{ik} - y_k) * \left(\frac{1}{1 + e^{-y_{in}}}\right) * \left[1 - \left(\frac{1}{1 + e^{-y_{in}}}\right)\right]$$

$$\delta_k = (1 - 0.05244) * \left(\frac{1}{1 + e^{-0.05244}}\right) * \left[1 - \left(\frac{1}{1 + e^{-0.05244}}\right)\right]$$

$$\delta_k = 0.0013$$

Hitung suku perubahan bobot  $w$  ( $\Delta w_i$ ) :

$$\Delta w_{jk} = \alpha * \delta_k * z_j$$

$\alpha$  = rasio

$\delta$  = faktor unit kesalahan

$$\Delta w_{10} = 0.05 * 0.0013 * 0.02514 = 0.0000016341$$

$$\Delta w_{20} = 0.05 * 0.0013 * 0.03254 = 0.00000253812$$

$$\Delta w_{30} = 0.05 * 0.0013 * 0.05412 = 0.0000035178$$

Hitung faktor unit kesalahan pada *layer* tersembunyi:

$$\delta_{inj} = \delta * w_{jk}$$

$$\begin{aligned} \delta_{in1} &= 0.0013 * 0.0000016341 \\ &= 0.0000000212433 \end{aligned}$$

$$\begin{aligned} \delta_{in2} &= 0.0013 * 0.00000253812 \\ &= 0.00000003299556 \end{aligned}$$

$$\begin{aligned}\delta_{in3} &= 0.0013 * 0.0000035178 \\ &= 0.0000000457314\end{aligned}$$

Hitung faktor kesalahan unit tersembunyi :

$$\delta_i = \delta_{in1} * \left(\frac{1}{1+e^{-z_j}}\right) * \left[1 - \left(\frac{1}{1+e^{-z_j}}\right)\right]$$

$$\begin{aligned}\delta_1 &= 0.00000000212433 * \left(\frac{1}{1+e^{-0.02514}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.02514}}\right)\right] \\ &= 0.0000012548\end{aligned}$$

$$\begin{aligned}\delta_2 &= 0.000000003299556 * \left(\frac{1}{1+e^{-0.03254}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.03254}}\right)\right] \\ &= 0.000002154\end{aligned}$$

$$\begin{aligned}\delta_3 &= 0.0000000457314 * \left(\frac{1}{1+e^{-0.05412}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.05412}}\right)\right] \\ &= 0.0000011188\end{aligned}$$

Hitung suku perubahan bobot v :

$$\Delta v_i = \alpha * \delta_j * x_i,$$

$$\Delta v_{1,1} = 0.05 * 0.0000012548 * 1 = 0.00000006274$$

$$\Delta v_{1,2} = 0.05 * 0.000002154 * 1 = 0.0000001077$$

$$\Delta v_{1,3} = 0.05 * 0.0000011188 * 1 = 0.00000005594$$

$$\Delta v = 0.0000, 0.0000, 0.0000, 0.0000, 0.0000$$

#### h. Perubahan bobot

Hitung bobot w baru:

$$w_{ik} = W_{ik} + \Delta W_{ik}$$

$$w_1 = 0.022 + 0.0000016341 = 0.0220016341$$

$$w_2 = 0.044 + 0.00000253812 = 0.04400253812$$

$$w_3 = 0.067 + 0.0000035178 = 0.0670035178$$

Hitung bobot v baru:

$$v_{i,j} = v_{i,j} + \Delta v_{i,j} = 10101$$

$$\begin{aligned}V &= [1 + 0.00000006274, 0 + 0.0000001077, 1 + 0.00000005594, \\ &\quad 0 + 0.00000005794, 1 + 0.00000004494]\end{aligned}$$

$$\begin{aligned}&= 1.0000006274, 0.0000001077, 1.00000005594, 0.00000005794, \\ &\quad 1 + 0.00000004494\end{aligned}$$

i. Bobot Akhir Hasil Perhitungan.

Nilai bobot akhir pada setiap iterasi dihitung sampai kondisi berhenti terpenuhi, kemudian dilanjutkan pada data ke-2 ( $x_2$ ). Setelah kondisi terpenuhi, maka bobot akhir yang diperoleh untuk data 1 setelah perulangan selesai adalah:

$$v = 0.000254, 0.0005541, 0.000214, 0.0008981, 0.0006652$$

$$w = 0.03265, 0.02514, 0.05874, 0.02512, 0.02122$$

### 3.4.3 Proses Pengenalan

Proses pengenalan wajah adalah proses jaringan yang menguji pengenalan data uji menggunakan bobot akhir yang diperoleh sebelumnya dalam proses pelatihan. Sebagai contoh, pengenalan citra wajah dengan input vektor citra adalah sebagai berikut:

[1 0 1 0 1] adalah dengan terlebih dahulu menghitung jarak dari dua data input berbobot  $W_1$  dan  $W_2$ . Langkah-langkahnya adalah menghitung bobot data uji untuk setiap citra wajah menggunakan perhitungan pendahuluan yang dilakukan secara terpisah untuk setiap bobot  $w$  dan  $v$  pada saat mengekstraksi setiap fitur citra wajah dan jika data yang dihasilkan memenuhi tujuan, perhitungan berhenti.

$$z_{in_j} = v o_j + \sum_{i=1}^n x_i v_{ij}$$

Vektor input: [1 0 1 0 1]

Bobot hasil pelatihan adalah:

$$0.8875, 0.3223, 0.5462, 0.1122, 0.4405$$

$$z_{in1} =$$

$$0 + (1 \cdot 0.8875) + (0 \cdot 0.3223) + (1 \cdot 0.5462) + (0 \cdot 0.1122) + (1 \cdot 0.4405)$$

$$= 3.2541$$

$$z_{in1} = 3.2541$$

$$z_{in2} = 3.4452$$

$$z_{in3} = 3.7488$$

$$z_{in4} = 3.9652$$

$$z_{in5} = 3.5874$$

$$Z1 = \frac{1}{1 + e^{-3.2541}} = 0.8958$$

$$Z2 = \frac{1}{1 + e^{-3.4452}} = 0.7785$$

$$Z3 = \frac{1}{1 + e^{-3.7488}} = 0.8556$$

$$Z4 = \frac{1}{1 + e^{-5.2577}} = 0.7741$$

$$Z5 = \frac{1}{1 + e^{-3.5874}} = 0.8774$$

Operasi pada *output layer* :

$$y^k = \frac{1}{1 + e^{-y_{in,k}}}$$

$$\begin{aligned} y_{in} &= 0 + (3.2541 * 0.8958) + (3.4452 * 0.7785) + (3.7488 * 0.7741) + \\ &\quad (3.5874 * 0.8774) \\ &= 2.9851 \end{aligned}$$

$$y = \frac{1}{1 + e^{-2.9851}}$$

$$y1 = 0.5552$$

$$y2 = 0.4587$$

$$y3 = 0.5562$$

$$y4 = 0.3541$$

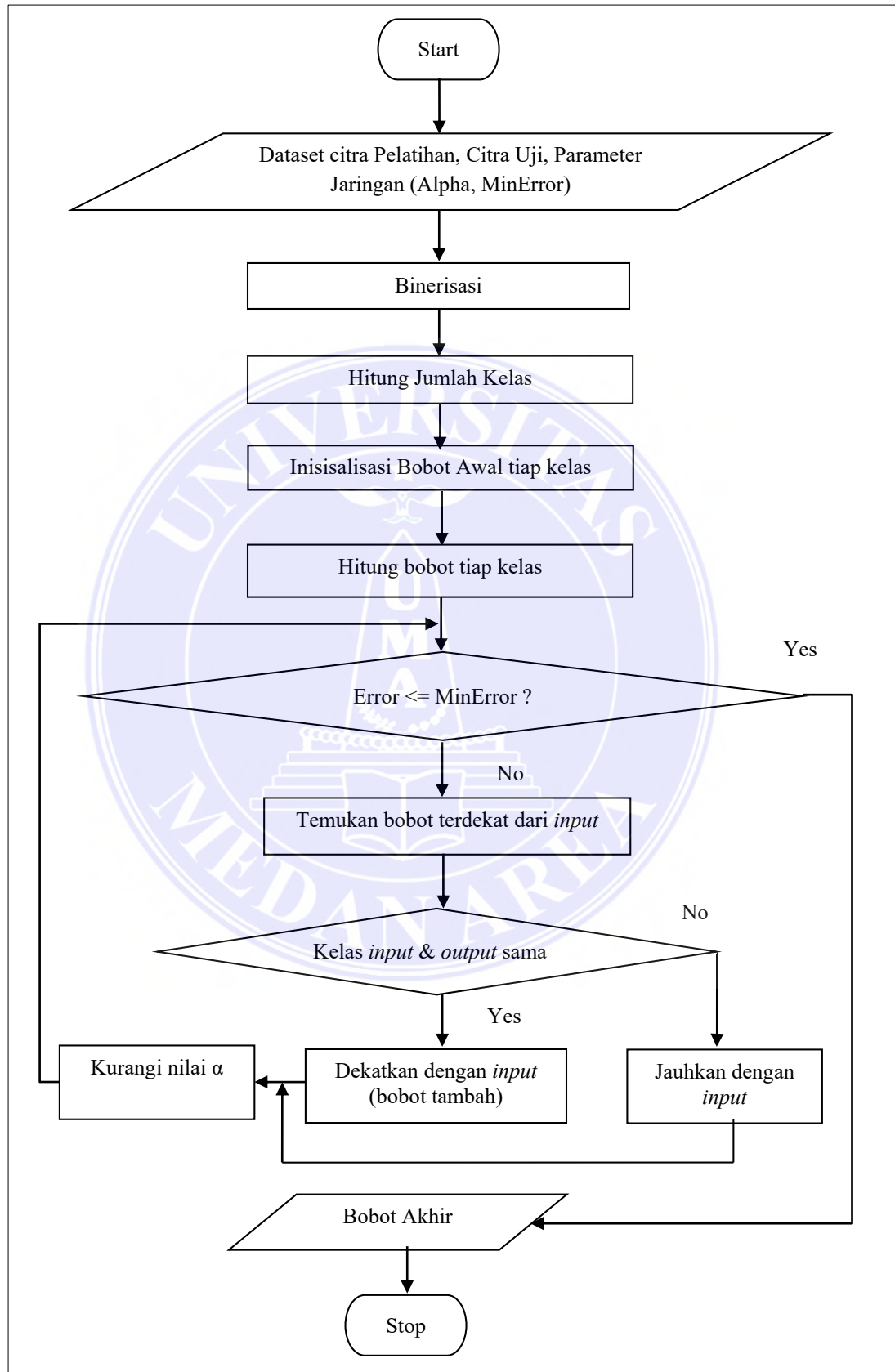
$$y5 = 0.8888$$

Nilai  $y_n$  hasil perhitungan pada langkah diatas menjadi hasil pengenalan dan kemudian tentukan nilai *Threshold* dari setiap citra wajahnya. Nilai *Threshold* yang digunakan adalah sebesar 0.5 yang berarti jika variabel  $y$  mempunyai angka dibelakang koma adalah lebih besar ( $>$ ) atau sama dengan ( $=$ ) dengan angka 5, maka nilai variabel itu dibulatkan keatas atau kebawahnya. Pada perhitungan diatas didapatkan nilai  $y = \{0.5552, 0.4587, 0.5562, 0.3541, 0.8888\}$ , maka hasil  $y = \{0,0,1,0,1\}$ , artinya citra uji memiliki nilai biner  $\{0,0,1,0,1\}$  sesuai dengan target yaitu citra wajah Citra -E.

### 3.5 Perancangan

#### 3.5.1 Flowchart Pelatihan

Dalam pelatihan, model gambar input dibaca dan disajikan dalam *Format* vektor sehingga kelas dapat dicari. Setiap neuron keluaran mengekspresikan kelas atau kategori tertentu, sehingga pola masukan dapat diidentifikasi oleh kelas neuron keluaran yang dihasilkan. Diagram alir proses pelatihan untuk pengenalan visa ditunjukkan pada Gambar 3.7.



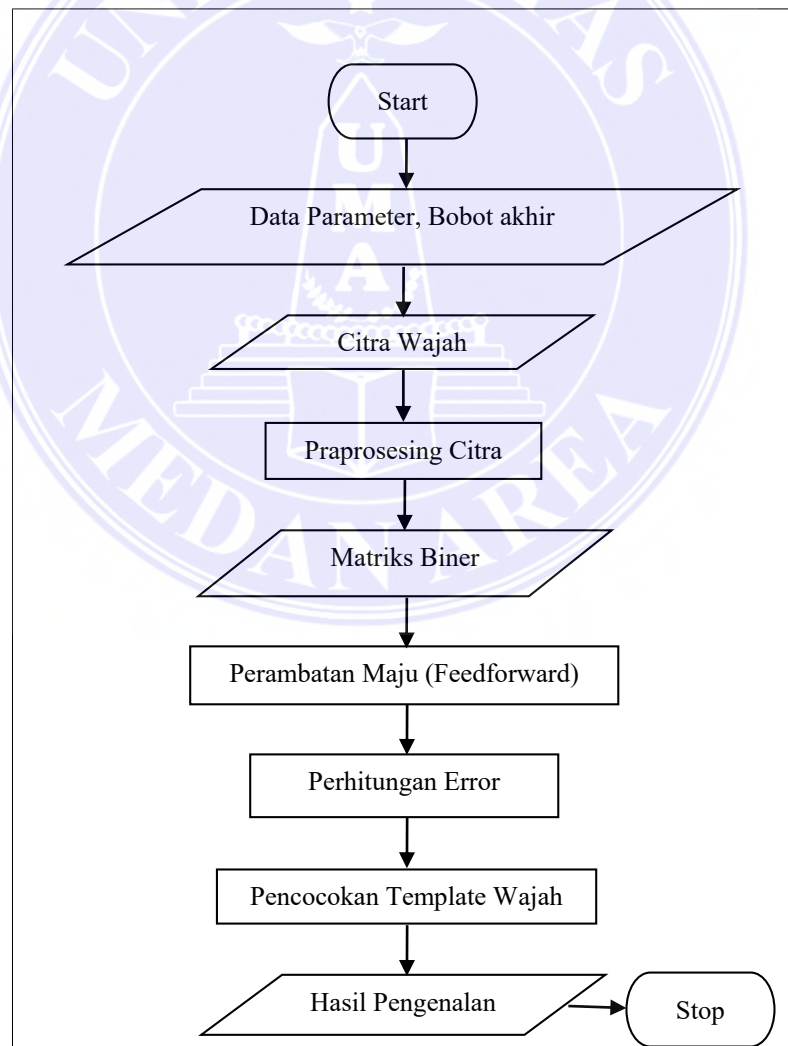
Gambar 3.7 Flowchart Pelatihan

Keterangan:

1. Input berupa citra pelatihan dan parameter jaringan.
2. Proses berupa *forward* dan *backward* dengan batasan nilai *error* serta *learning rate*.
3. Proses perhitungan bobot akhir setiap citra pelatihan.

### 3.5.2 Flowchart Pengenalan Wajah

*Flowchart* pengenalan Wajah merupakan bagan atau gambar yang menunjukkan aliran proses untuk menjelaskan alur program untuk melakukan proses pengenalan wajah selanjutnya dibawah ini *flowchart* pengenalan wajah yang dapat dilihat seperti pada Gambar 3.8.



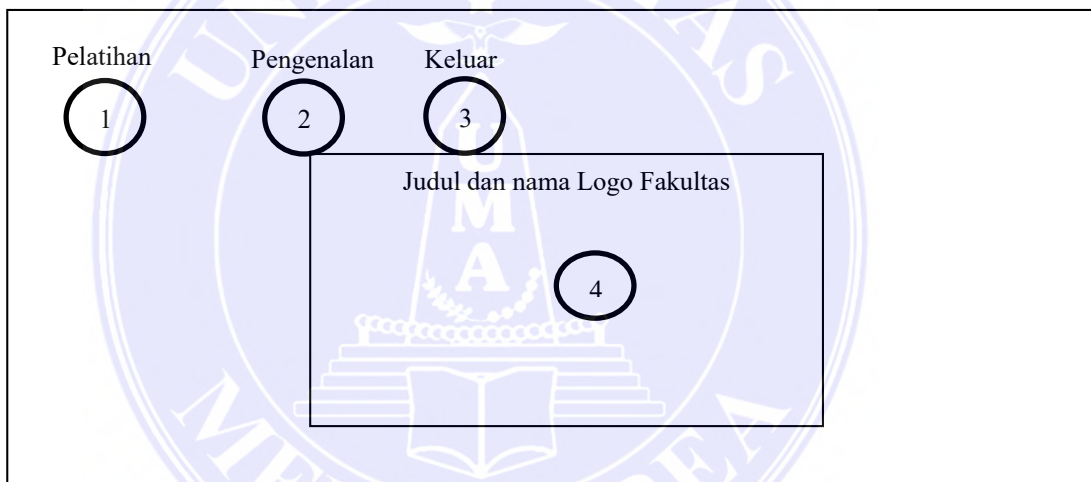
Gambar 3.8 *Flowchart* Pengenalan Wajah

Keterangan:

1. Input berupa nilai piksel citra uji dan nilai bobot akhir.
2. Proses *feedforward* berupa perhitungan bobot baru dari nilai piksel citra uji.
3. Proses perhitungan *error* yaitu jarak input tersebut dengan kedua bobot akhir hasil *training*.
4. Proses pengenalan wajah dengan pencocokan *template*.

### 3.5.3 Perancangan Form Menu Utama

Perancangan *Form* Menu Utama adalah rancangan yang berfungsi untuk meletakkan menu aplikasi pelatihan dan pengenalan wajah seperti pada Gambar 3.9.



Gambar 3.9 Rancangan Form Menu Utama

Keterangan:

1. Menu = tempat menu Pelatihan
2. Menu = tempat menu Pengenalan
3. Button = tombol menutup aplikasi
4. Edit = tempat menampilkan Judul Penelitian dan Logo Fakultas

### 3.5.4 Perancangan Form Pelatihan

Perancangan Pelatihan adalah rancangan yang berfungsi untuk melakukan pelatihan semua citra wajah dengan inputan nilai error minimum dan nilai *learning*

rate yang pada penelitian ini bernilai 0.3 dan 0.5. Adapun rancangan *Form* pelatihan dapat dilihat seperti pada Gambar 3.10.

The form consists of two rows. The first row has the label 'Error Min' followed by a rectangular input box. The second row has the label 'Learning Rate' followed by another rectangular input box.

Gambar 3.10 Rancangan Form Pelatihan

### 3.5.5 Perancangan *Form* Pengenalan

Perancangan *Form* pengenalan adalah rancangan yang berfungsi untuk melakukan pengenalan citra wajah seperti pada Gambar 3.11.

The form is a rectangular window containing several elements:
 

- Three large rectangular boxes at the top, labeled 'Citra Uji' (1), 'Citra Uji Biner' (2), and 'Citra Hasil Pengenalan' (3).
- Below these, a 'File' input field (4), a 'Ukuran' input field (5), a 'Hasil' input field (6), and a 'Lama' input field (7).
- At the bottom, a row of four buttons: 'File' (8), 'Proses' (9), 'Bersih' (10), and 'Keluar' (11).

Gambar 3.11 Rancangan Form Pengenalan

Keterangan:

1. Axes = tempat menampilkan citra uji
2. Axes = tempat menampilkan citra biner
3. Axes = tempat menampilkan citra hasil pengenalan
4. Text = tempat menampilkan nama file citra uji
5. Text = tempat menampilkan ukuran file citra uji
6. Text = tempat menampilkan hasil pengujian
7. Text = tempat menampilkan lama pengujian
8. Button = tombol untuk mencari dan menampilkan citra uji
9. Button = tombol untuk melakukan proses pengenalan



10. Button = tombol untuk melakukan pembersihan *Form*

11. Button = tombol untuk melakukan penutupan *Form*



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melakukan pengenalan citra wajah dengan terlebih melakukan pelatihan citra wajah-wajah sebanyak 1866 citra yang berdimensi 142 x 142 piksel, maka dapat diambil kesimpulan bahwa:

- a. Aplikasi dapat melakukan pembacaan nilai piksel citra pelatihan dengan parameter jaringan minimal *error* sebesar 0.001,  $\alpha = 0.30$  pada *epoch* =9 dan  $\alpha = 0.50$  pada *epoch* =15.
- b. Aplikasi dapat melakukan pelatihan semua citra wajah.
- c. Aplikasi dapat melakukan pengenalan 50 citra wajah dengan variasi ekspresi yang berbeda-beda.
- d. Nilai akurasi yang terbaik proses pengenalan wajah dengan *learning rate*  $\alpha = 0.50$ , *Epoch* =15 adalah sebesar 82 %.

#### 5.2 Saran

Adapun saran-saran pada penelitian ini adalah untuk menaikkan nilai hasil pengenalan wajah agar dilakukan proses praprosesing citra seperti menambah ketajaman, kontras citra terlebih dahulu serta ekstraksi fitur. Aplikasi yang telah dibangun diatas dapat dikembangkan lagi menjadi sistem absensi dengan menambahkan fitur camera untuk melakukan *capture* citra wajah menjadi file dengan *Format* yang sesuai.

## DAFTAR PUSTAKA

- Amalia, N., Hidayat, E. W., & Aldya, A. P. (2020). Sundanese Script Recognition Using Backpropagation Neural Network Methods and Canny Edge Detection. *CESS (Journal of Computer Engineering, System and Science)*, 5(1), 19. <https://doi.org/10.24114/cess.v5i1.14839>
- Anmol, M., & Vinay, S. (2018). *Studi Kontemporer Sistem Pengenalan Wajah untuk Sistem Otonomi Absensi*. 8(285), 285–295.
- Atina, A. (2019). Matlab Applications in Medical Imaging Technology. *Journal of Research in Physics and Its Applied (JUPITER)*, 1(1), 28. <https://doi.org/10.31851/jupiter.v1i1.3123>
- Chau, S., Banjarnahor, J., Irfansyah, D., Kumala, S., & Banjarnahor, J. (2019). Analisis Deteksi Pola Wajah Menggunakan Metode Haar-Like Feature. *Journal of InFormation Technology Education: Research*, 2(2), 70–76. <https://doi.org/10.31289/jite.v2i2.2133>
- Ghozali, I., & Adikara, P. P. (2018). Implementation of the Backpropagation Method for Recognizing Text in Natural Scene Images. *Jurnal Pengembangan Teknologi InFormasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 2(8), 2527–2533.
- Jumadi, J., Yupianti, Y., & Sartika, D. (2021). Digital Image Processing for Object Identification Using Hierarchical Agglomerative Clustering Methods. *JST (Jurnal Sains Dan Teknologi)*, 10(2), 148–156. <https://doi.org/10.23887/jstundiksha.v10i2.33636>
- Kurniawan, E., Wibawanto, H., & Widodo, D. A. (2019). Implementation of the Backpropagation Method with Nguyen Widrow's Weight Initialization for Stock Price Forecasting. *Jurnal Teknologi InFormasi Dan Ilmu Komputer*, 6(1), 49. <https://doi.org/10.25126/jtiik.201961904>
- Prastya, M. A. (2019). Human Face Recognition System Using Viola-Jones Algorithm And Principal Component Analysis. *Ubiquitous: Computers and Its Applications Journal*, 2, 85–92. <https://doi.org/10.51804/ucaiaj.v2i2.85-92>
- Suryo Putro S, B. C., Mustika, I. W., & Nugroho, L. E. (2018). Algoritma Jaringan Syaraf Tiruan Back-propagation yang Dioptimalkan untuk Pertanian Cerdas Applications. *Proceedings - 2018 4th International Conference on Science*

*and Technology, ICST 2018, 1, 1–5.*  
<https://doi.org/10.1109/ICSTC.2018.8528655>

Taghi Zadeh, M. M., Imani, M., & Majidi, B. (2019). Pengenalan emosi wajah yang cepat Menggunakan Jaringan Neural Konvolusional dan Filter Gabor. *2019 IEEE 5th Conference on Knowledge Based Engineering and Innovation, KBEI 2019, 577–581.* <https://doi.org/10.1109/KBEI.2019.8734943>

Tandrian, A. H., & Kusnadi, A. (2019). Leaf Pattern Recognition With Backpropagation Neural Networks. *ULTIMA Computing, 10(2), 53–58.* <https://doi.org/10.31937/sk.v10i2.1063>



## LAMPIRAN - LAMPIRAN



Similarity Report ID: oId:29477:37392314

PAPER NAME	AUTHOR
Perbaikan Skripsi Maya Dani 12 Juni 2023-15-66.pdf	maya dani sari

---

WORD COUNT	CHARACTER COUNT
10158 Words	53913 Characters

---

PAGE COUNT	FILE SIZE
52 Pages	1.0MB

---

SUBMISSION DATE	REPORT DATE
Jun 12, 2023 4:08 PM GMT+7	Jun 12, 2023 4:09 PM GMT+7

---

**25% Overall Similarity**  
The combined total of all matches, including overlapping sources, for each database.

- 24% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 8% Submitted Works database


**Excluded from Similarity Report**

- Small Matches (Less than 10 words)



Summary





**UNIVERSITAS MEDAN AREA**  
**FAKULTAS TEKNIK**

Kampus I : Jalan Keleni Honor I Medan Estate (Jalan PPS) Nomor 1 Medan (061) 7392070, 7360108, 7304348, 7390781, Fax (061) 7399998 Medan 20223  
Kampus II : Jalan Setiabudi Nomor 70 / Jalan Sei Selayu Honor 70 A, MF (061) 8079902, Fax (061) 8220331 Medan 20122  
Website: www.teknik.uma.ac.id E-mail: umy\_mechanics@uma.ac.id

---

Nomor : 289 /FT.6/01.10/X/2022  
Lamp : -  
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

1 Oktober 2022

Yth. Kepala Laboratorium STMIK Triguna Dharma  
Jln. A. H. Nasution No. 73 F  
Di  
Medan

Dengan hormat, kami mohon kesediaan Bapak/Ibu kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :


NO	NAMA	NPM	PRODI
1	Maya Dani Sari	178160077	Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di Laboratorium STMIK Triguna Dharma.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi, merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana lengkap pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

**Implementasi Algoritma Backpropagation untuk Pengenalan Wajah Berbasis Citra.**

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.



Dr. Ratu Dicky S. Kom, M. Kom

Tembusan :  
1. Ka. BAMA  
2. Mahasiswa  
3. File

