

# Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time

Muhammad Zufar dan Budi Setiyono

Jurusan Matematika, Fakultas MIPA, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*e-mail:* budi@matematika.its.ac.id

**Abstrak**—Identifikasi identitas individu melalui pengenalan wajah secara otomatis merupakan suatu persoalan besar yang menarik dan banyak sekali berbagai macam pendekatan untuk menyelesaikan persoalan ini. Apalagi di dalam skenario kehidupan nyata yang tidak terkontrol, wajah akan terlihat dari berbagai sisi dan tidak selalu menghadap ke depan yang membuat permasalahan klasifikasi menjadi lebih sulit diselesaikan. Dalam Tugas Akhir ini digunakan salah satu metode *deep neural networks* yaitu *Convolutional Neural Networks* (CNN) sebagai pengenalan wajah secara real-time yang sudah terbukti sangat efisien dalam klasifikasi wajah. Metode diimplementasikan dengan bantuan *library* OpenCV untuk deteksi multi wajah dan perangkat Web Cam M-Tech 5MP. Dalam penyusunan arsitektur model *Convolutional Neural Networks* dilakukan konfigurasi inisialisasi parameter untuk mempercepat proses training jaringan. Hasil uji coba dengan menggunakan konstruksi model *Convolutional Neural Networks* sampai kedalaman 7 lapisan dengan input dari hasil ekstraksi *Extended Local Binary Pattern* dengan radius 1 dan neighbor 15 menunjukkan kinerja pengenalan wajah meraih rata-rata tingkat akurasi lebih dari 89% dalam  $\mp 2$  frame per detik.

**Kata Kunci**—Convolutional Neural Networks, Extended Local Binary Pattern, Pengenalan Wajah, Real-Time.

## I. PENDAHULUAN

Sebuah sistem autentikasi identitas manusia yang memiliki akurasi tinggi sangat diperlukan saat ini karena meningkatnya jumlah kejahatan dan kerugian melalui penipuan identitas. Sistem berbasis token (*traditional token-based system*) dan berbasis pengetahuan (*knowledge-based system*) memiliki resiko tinggi dalam kasus pencurian atau lupa sandi, oleh karena itu secara luas sistem saat ini menggunakan sistem biometrik seperti kontrol akses, identifikasi kriminal, *autonomous vending*, dan *automated banking* karena keunikan fitur biometrik dan karakteristik yang tidak dapat dipindah tangankan (*non-transferable characteristic*)[1]. Biometrik dibagi menjadi dua kategori yaitu berdasarkan perilaku seseorang seperti ritme mengetik, gaya berjalan atau suara dan berdasarkan fisiologis seperti sidik jari, wajah, iris, atau tanda tangan. Diantara tipe biometrik fisiologis yang disebutkan sebelumnya, pengenalan wajah hingga saat ini masih menjadi daerah penelitian sejak tahun 1960 dengan cakupan ruang yang luas untuk terus dilakukan penyempurnaan.

Pengenalan wajah merupakan permasalahan yang menantang. Salah satu faktornya adalah karena berbagai macam posisi gambar wajah. Kamera dapat menangkap posisi wajah dari depan, samping, atau dari sudut tertentu sehingga menyebabkan beberapa fitur wajah seperti mata

atau hidung menjadi tidak terlihat secara penuh. Faktor lain adalah karena ada atau tidak adanya komponen struktural seperti jenggot, kumis, atau dengan / tanpa kacamata pada gambar wajah. Komponen-komponen struktural memiliki banyak variabilitas termasuk bentuk, warna dan ukuran. Faktor-faktor lain yang dapat mempengaruhi akurasi meliputi pencahayaan, oklusi dan ekspresi wajah. Iluminasi adalah perubahan dari persebaran cahaya karena sifat reflektansi kulit dan kontrol kamera internal yang dapat melemparkan bayangan pada beberapa bagian dari wajah. Oklusi adalah hasil dari suatu objek yang meliputi bagian wajah, seperti syal, sorban, dll. Contoh ekspresi wajah yang tersenyum, tertawa, marah, sedih, terkejut, dan takut[1].

Salah satu teknik yang paling terkenal dalam menangani masalah ini adalah teknik klasifikasi Jaringan Syaraf Tiruan (JST). Jaringan syaraf tiruan adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia. JST merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut dengan kata lain teknik ini memiliki kemampuan untuk belajar dari pengalaman. Jenis model jaringan syaraf tiruan yang memiliki beberapa lapisan disebut sebagai *Multi-Layer Perceptron* (MLP) yang menghubungkan secara penuh antar neuronnya memiliki kemampuan klasifikasi yang powerful. Akan tetapi, MLP memiliki beberapa masalah ketika input berupa gambar. Gambar harus dilakukan pre-processing, segmentasi, dan ekstraksi fitur untuk mendapatkan kinerja optimal. Hal ini menyebabkan MLP memiliki banyak parameter bebas atau informasi yang berlebihan dalam arsitektur. Parameter bebas berasal dari pembentukan oleh skema koneksi penuh antara input dan peta fitur dari lapisan yang berhubungan.

Variasi lain dari MLP yang dapat mengatasi permasalahan yang dijelaskan sebelumnya disebut sebagai *Convolutional Neural Networks* (CNN). CNN terinspirasi oleh korteks mamalia visual sel sederhana dan kompleks. Model ini dapat mengurangi sejumlah parameter bebas dan dapat menangani deformasi gambar input seperti translasi, rotasi dan skala[2].

Berdasarkan penjelasan kelebihan CNN tersebut, dapat diambil kesimpulan bahwa CNN memiliki kemampuan klasifikasi yang diperuntukkan untuk data gambar sehingga pada Tugas Akhir ini model CNN akan digunakan sebagai pengenalan wajah dari berbagai sisi secara real-time.

## II. TINJAUAN PUSTAKA

### A. Citra Digital

Citra adalah representasi, kemiripan atau imitasi dari suatu objek atau benda[3]. Secara matematis, citra dinyatakan sebagai suatu fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Citra dibedakan menjadi dua yaitu citra kontinu diperoleh dari sistem optik yang menerima sinyal analog(mata manusia dan kamera analog) dan citra diskrit(digital) dihasilkan melalui proses digitalisasi terhadap citra kontinu.

Proses digitalisasi pada citra digital dibagi menjadi dua proses yakni sampling dan kuantisasi. Proses sampling merupakan proses pengambilan nilai diskrit koordinat ruang(x,y) secara periodik dengan periode sampling T. Proses kuantisasi merupakan proses pengelompokkan nilai tingkat keabuan citra kontinu kedalam beberapa level atau merupakan proses membagi skala keabuan (0,L) menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat (*integer*), dinyatakan sebagai  $G = 2^m$ , dengan G adalah derajat keabuan dan m adalah bilangan bulat positif.

Dengan demikian citra digital dapat didefinisikan suatu matriks A berukuran M (baris) x N (kolom) dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya menyatakan tingkat keabuan pada titik tersebut.

### B. Extended Local Binary Pattern

Standar operator *Local Binary Pattern* (LBP) didefinisikan sebagai perbandingan nilai biner piksel pada pusat citra dengan 8 nilai piksel disekelilingnya. Secara umum LBP didefinisikan sebagai berikut[4]:

$$s(g_p - g_c) = \begin{cases} 1, & g_p \geq g_c \\ 0, & g_p < g_c \end{cases} \quad (1)$$

$$LBP_{R,P} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (2)$$

- $P$  : Jumlah *neighbour*  
 $R$  : Radius pada *circular neighbour*  
 $g_c$  : Nilai intensitas pusat piksel (*threshold*)  
 $g_p$  : Nilai intensitas *neighbour*  
 $p$  : Indeks *neighbour*

Perluasan dari LBP yaitu *Extended Local Binary Pattern* (ELBP) atau *Rotated Local Binary Pattern* (RLBP). Perluasan ini dalam rangka untuk membuat LBP invarian terhadap rotasi maka dilakukan penggeseran bobot secara sirkular sesuai dengan arah domain. Arah domain(D) dalam *neighbourhood* adalah indeks pada *neighbour* yang memiliki perbedaan maksimal dengan nilai intensitas pusat piksel. Didefinisikan sebagai berikut:

$$D = \underset{p \in (0,1 \dots P-1)}{\arg \max} |g_p - g_c| \quad (3)$$

Rotasi *neighbourhood* dengan pusatnya bergeser ke arah D dengan sudut yang sama. Sebagai contoh *neighbourhood* [23 25 28; 167 35 31; 56 67 72] setelah dilakukan *thresholding* kode biner menjadi [01111000], indeks D adalah 4 ditunjukkan pada tulisan bercetak

tebal pada angka biner, berkorespondensi ke nilai piksel 167. Jika gambar diputar  $45^\circ$  berlawanan arah jarum jam, maka *neighbourhood* juga berputar dengan sudut yang sama dan kode biner bergeser menjadi [00111100]. Indeks D tetap berkorespondensi dengan nilai yang sama 167 tetapi bergeser satu langkah. Oleh karena arah dominan diambil sebagai acuan dalam *circular neighbourhood*, maka bobot juga berdasarkan acuan tersebut. Sehingga operator RLBP didefinisikan sebagai berikut:

$$RLBP_{R,P} = \sum_{p=0}^{P-1} s(g_p - g_c)2^{\text{mod}(p-D,P)} \quad (4)$$

Dimana *mod* mengindikasikan operasi modulo. Pada definisi bobot  $2^{\text{mod}(p-D,P)}$  bergantung pada D. Bobot yang sirkuler bergeser sehubungan dengan arah dominan. Hasil penggeseran invarian terhadap rotasi, karena bobot sekarang tergantung pada *neighbourhood* dan bukan pada susunan yang dipilih sebelumnya.

### C. Operasi Konvolusi

Citra digital terbatas pada ukuran  $M \times N$  dengan kernel berukuran  $m \times n$  sehingga operasi fungsi konvolusi didefinisikan sebagai berikut[3]:

$$(f * g)(x, y) := \sum_{a=x-h}^{x+h} \sum_{b=y-w}^{y+h} f(a, b)g(x - a, y - b) \quad (5)$$

Dimana  $m = 2h + 1$  adalah tinggi kernel dan  $n = 2w + 1$  adalah lebar kernel. Operasi dilakukan dengan melakukan kombinasi linear dari mengambil bagian input citra yang sama dengan kernel dan nilai hasil operasi disimpan berupa elemen nilai matriks kemudian dilanjutkan dengan menggeser kernel piksel per piksel sampai terhimpun keseluruhan nilai piksel baru[5].

### D. Convolutional Neural Networks (CNN)

CNN adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan syaraf manusia. Penelitian awal yang mendasari penemuan ini pertama kali dilakukan oleh Hubel dan Wiesel[6] yang melakukan penelitian *visual cortex* pada indera penglihatan kucing. *Visual cortex* pada hewan sangat *powerful* dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan model-model baru diantaranya seperti Neocognitron [7], HMAX [8], dan LeNet-5 [2].

*Convolutional Neural Networks* merupakan suatu *layer* yang memiliki susunan *neuron* 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada jumlah *layer*. Secara umum jenis *layer* pada CNN dibedakan menjadi dua yaitu:

- Layer* ekstraksi fitur gambar, letaknya berada pada awal arsitektur tersusun atas beberapa *layer* dan setiap *layer* tersusun atas *neuron* yang terkoneksi pada daerah lokal (*local region*) *layer* sebelumnya. *Layer* jenis pertama adalah *layer* konvolusi dan *layer* kedua adalah *layer pooling*. Setiap *layer* diberlakukan fungsi aktivasi. Posisinya berselang-seling antara jenis pertama dengan jenis kedua. *Layer* ini menerima input gambar secara langsung dan memrosesnya

hingga menghasilkan keluaran berupa vektor untuk diolah pada layer berikutnya.

- b. *Layer* klasifikasi, tersusun atas beberapa layer dan setiap *layer* tersusun atas *neuron* yang terkoneksi secara penuh (*fully connected*) dengan *layer* lainnya. *Layer* ini menerima input dari hasil keluaran *layer* ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa skoring kelas untuk klasifikasi.

Dengan demikian CNN merupakan metode untuk mentransformasikan gambar original *layer* per *layer* dari nilai piksel gambar kedalam nilai skoring kelas untuk klasifikasi. Dan setiap *layer* ada yang memiliki *hyperparameter* dan ada yang tidak memiliki parameter (bobot dan bias pada *neuron*).

#### 1. Convolutional Layer

*Layer* yang pertama kali menerima input gambar langsung pada arsitektur. Operasi pada *layer* ini sama dengan operasi konvolusi yaitu melakukan operasi kombinasi linier *filter* terhadap daerah lokal. *Filter* merupakan representasi bidang reseptif dari *neuron* yang terhubung kedalam daerah lokal (*local connectivity*) pada input gambar. Bentuk *layer* direpresentasikan sebagai volume  $B \times K \times L$  atau *layer* ukuran  $B \times K$  dengan jumlah sebanyak  $L$ . *Convolutional layer* memiliki *hyperparameter* dan parameter.

**Tabel 1.** Hyperparameter pada Convolutional Layer.

No	Hyperparameter	Keterangan
1	Depth	Kedalaman <i>layer</i> atau jumlah <i>layer</i> konvolusi
2	Stride	Jumlah pergeseran <i>filter</i> pada proses konvolusi
3	Zero-padding	Jumlah penambahan nilai intensitas nol di daerah sekitar input gambar

#### 2. Pooling Layer C1

*Pooling layer* akan mereduksi ukuran spasial dan jumlah parameter dalam jaringan serta mempercepat komputasi dan mengontrol terjadinya *overfitting*. *Pooling layer* bekerja dengan blok spasial yang bergerak sepanjang ukuran *feature pattern*. Ukuran pergeseran blok pada umumnya adalah ukuran pada dimensi blok ( $H \times H$ ) itu sendiri sehingga tidak ada *overlapping* seperti pada *Convolutional Layer*. Pergerakan blok diikuti dengan perhitungan *pooling* pada masukan pola fitur. Pada *layer* ini tidak memiliki parameter karena parameter sudah ditentukan sebelumnya (*fixed*). *Pooling layer* memiliki beberapa macam tipe antara lain *average pooling*, *max pooling*, dan *Lp Pooling*.

#### 3. Fungsi Aktivasi (Neurons)

Fungsi aktivasi atau fungsi transfer merupakan fungsi non-linier yang memungkinkan sebuah jaringan untuk dapat menyelesaikan permasalahan permasalahan non trivial. Setiap fungsi aktivasi mengambil sebuah nilai dan melakukan operasi matematika. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir keluaran *feature map* atau sesudah proses perhitungan konvolusi atau *pooling* untuk menghasilkan suatu pola fitur.

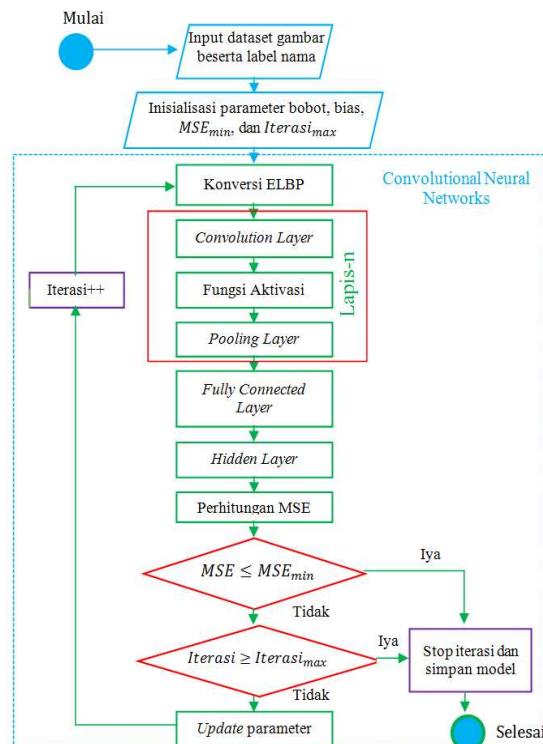
Beberapa macam fungsi aktivasi yang sering digunakan dalam penelitian antara lain fungsi sigmoid, tanh, *Rectified Linear Unit* (ReLU), Leaky ReLU (LReLU) dan *Parametric ReLU*.

#### E. Library OpenCV

OpenCV adalah sebuah library fungsi perrograman yang ditujukan untuk *computer vision*. Awalnya dikembangkan oleh pusat penelitian Intel di Nizhny Novgorod (Rusia), kemudian didukung oleh Willow Garage dan sekarang dikelola oleh Itseez. Library OpenCV di bawah lisensi BSD *open-source* gratis dan *cross-platform* untuk digunakan. Didalamnya terdapat ratusan algoritma *computer vision*.

### III. METODE PENELITIAN

Dalam melakukan penelitian ini, langkah pertama yaitu pengumpulan dataset himpunan gambar wajah yang dibagi menjadi dua jenis himpunan. Himpunan gambar pertama adalah himpunan gambar wajah indoor atau gambar wajah dalam kondisi pencahayaan minim sedangkan himpunan gambar kedua adalah gambar wajah outdoor atau gambar wajah dalam kondisi pencahayaan terang. Kemudian dilakukan desain arsitektur dimulai dari menentukan kedalaman jaringan, susunan layer, dan pemilihan jenis layer yang akan digunakan untuk mendapatkan model berdasarkan input dataset dan index label nama. Berikut langkah-langkah algoritma CNN untuk mendapatkan model dengan *training* jaringan:



**Gambar 1.** Diagram alir training Convolutional Neural Networks.






### IV. PERANCANGAN DAN IMPLEMENTASI SISTEM

#### A. Pengumpulan dan Perluasan Dataset

Pengumpulan data gambar dilakukan langsung menggunakan kamera Web Cam. Setiap gambar wajah akan dirubah dan disimpan dalam bentuk *grayscale* berukuran  $48 \times 48$  piksel untuk dijadikan sebagai *dataset*.

Akan tetapi *dataset* yang didapatkan masih sangat terbatas untuk mendapatkan kinerja optimal pada sistem, oleh karena itu dilakukan *data augmentation* untuk menambah variasi *dataset* yang disajikan pada Tabel 1.

Tabel. 2. Daftar metode *data augmentation*.

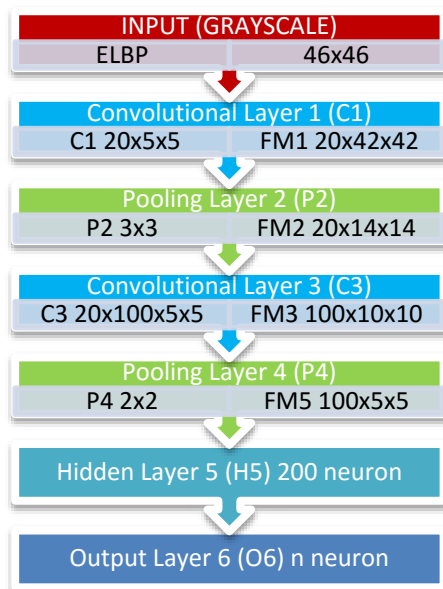
1	<i>Horizontally flipping</i> Pembalikan gambar wajah secara horizontal	
2	<i>Translation</i> Pergeseran gambar wajah dari 1 piksel ke 2 piksel	
3	<i>Rotation</i> Perputaran gambar wajah mulai dari 3 sampai 6 derajat	
4	<i>Scaling</i> Resize dengan skala antara 0.9 ~ 1.1	
5	<i>Randomly blackening</i> Penghitaman secara acak dengan intensitas 20%	

B. *Data preprocessing*

Gambar 48x48 grayscale diekstraksi terlebih dahulu menggunakan metode *Extended Local Binary Pattern* sebelum masuk jaringan. Metode ini dipilih karena hasil ekstraksi lebih toleran terhadap perubahan intensitas pencahayaan, distorsi perspektif, blur, perubahan skala gambar, dan sebagainya. Pada Tugas Akhir ini pengaturan radius diberi nilai 1 dan *neighbor* diberi nilai 15 sehingga dimensi gambar dataset berubah menjadi 46x46.

C. *Desain Arsitektur Convolutional Neural Networks*

Data Arsitektur *Convolutional Neural Networks* dalam Tugas akhir ini dirancang dengan kedalaman 7 layer model konvolusi antara lain *input layer*, *convolutional layer C1*, *pooling layer P2*, *convolutional layer C3*, *pooling layer P4*, *hidden layer H5* dan *output layer O6*. Gambaran susunan arsitektur disajikan pada Gambar 2.



Gambar 2. Arsitektur Convolutional Neural Networks untuk pengenalan wajah.

D. *Pemodelan Convolutional Neural Networks*

Konstruksi model jaringan mengacu pada arsitektur *Convolutional Neural Networks* yang sudah dibangun. Model berupa persamaan umum fungsi transfer antara pola fitur dengan pola fitur lainnya. Berikut hasil model pada setiap *layer* arsitektur jaringan.

1) *Convolutional Layer C1*

$$FM1_{(i_1, j_1)}^{(m_1)} = \tanh \left( b1^{(m_1)} + \sum_{r_1=0}^4 \sum_{c_1=0}^4 C1_{(r_1, c_1)}^{(m_1)} * I_{((r_1+i_1), (c_1+j_1))} \right) \quad (6)$$

Keterangan variabel dan index pada *Convolutional Layer C1* adalah sebagai berikut:

- $I$  : Input gambar
- $b1$  : Bias konvolusi layer ke-1
- $C1$  : *Filter* konvolusi layer ke-1
- $FM1$  : *Feature Map-1*
- $x$  : Index panjang baris input gambar
- $y$  : Index panjang kolom input gambar
- $r_1$  : Index panjang baris *filter* konvolusi layer ke-1
- $c_1$  : Index panjang kolom *filter* konvolusi layer ke-1
- $i_1$  : Index baris *feature map* layer ke-1
- $j_1$  : Index kolom *feature map* layer ke-1
- $m_1$  : Index jumlah *feature pattern* ke-1
- $n_x$  : Panjang baris input gambar
- $n_y$  : Panjang kolom input gambar
- $n_{r_1}$  : Panjang baris *filter* konvolusi layer ke-1
- $n_{c_1}$  : Kolom *filter* konvolusi layer ke-1
- $n_{i_1}$  : Panjang baris *feature map* layer ke-1
- $n_{j_1}$  : Panjang kolom *feature map* layer ke-1
- $n_{m_1}$  : Jumlah jumlah *feature pattern* ke-1

2) *Pooling Layer P2*

$$FM2_{(i_2, j_2)}^{(m_2)} = \tanh \left( b2^{(m_2)} + \sum_{i_1=3i_2}^{3n_{i_1}+2} \sum_{j_1=3j_2}^{3n_{j_1}+2} P2_{(i_1, j_1)}^{(m_2)} * FM1_{(i_1, j_1)}^{(m_1)} \right) \quad (7)$$

Keterangan variabel dan index pada *Pooling Layer P2* adalah sebagai berikut:

- $b2$  : Bias *filter* pooling layer ke-2
- $P2$  : *Filter* pooling layer ke-2
- $i_2$  : Index baris *feature map* ke-2
- $j_2$  : Index kolom *feature map* ke-2
- $n_{i_2}$  : Panjang baris *feature map* ke-2
- $n_{j_2}$  : Panjang kolom *feature map* ke-2

3) *Convolutional Layer C3*

$$FM3_{(i_3, j_3)}^{(m_3)} = \tanh \left( b3^{(m_3)} + \sum_{m_1=0}^{n_{m_1}-1} \sum_{r_3=0}^{n_{r_3}-1} \sum_{c_3=0}^{n_{c_3}-1} C3_{(r_3, c_3)}^{(m_1, m_2)} * FM2_{(r_3+i_3, c_3+j_3)}^{(m_2)} \right) \quad (8)$$

Keterangan variabel dan index pada *Convolutional Layer C3* adalah sebagai berikut:

- $b3$  : Bias konvolusi layer ke-3
- $C3$  : *Filter* konvolusi layer ke-3
- $FM3$  : *Feature Map-3*
- $m_2$  : Index kedalaman *feature pattern* ke-2
- $r_3$  : Index panjang baris *filter* konvolusi layer ke-3
- $c_3$  : Index panjang kolom *filter* konvolusi

- layer ke-3
- $i_3$  : Index baris *feature map* layer ke-3
  - $j_3$  : Index kolom *feature map* layer ke-3
  - $n_{m_2}$  : Jumlah kedalaman *feature pattern* ke-2
  - $n_{r_3}$  : Panjang baris *filter* konvolusi layer ke-3
  - $n_{c_3}$  : Kolom *filter* konvolusi layer ke-3
  - $n_{i_3}$  : Panjang baris *feature map* layer ke-3
  - $n_{j_3}$  : Panjang kolom *feature map* layer ke-3

#### 4) Pooling Layer P4

$$FM4_{(i_4, j_4)}^{(m_2)} = \tanh \left( b4^{(m_2)} + \sum_{i_3=2i_4}^{3n_{i_3}+2} \sum_{j_3=2j_4}^{3n_{j_3}+2} P4^{(m_2)} * FM3_{(i_3, j_3)}^{(m_2)} \right) \quad (9)$$

Keterangan variabel dan index pada *Pooling Layer P4* adalah sebagai berikut:

- $b4$  : Bias *filter* pooling layer ke-4
- $P4$  : *Filter* pooling layer ke-4
- $i_4$  : Index baris *feature map* ke-4
- $j_4$  : Index kolom *feature map* ke-4
- $n_{i_4}$  : Panjang baris *feature map* ke-4
- $n_{j_4}$  : Panjang kolom *feature map* ke-4

#### 5) Hidden Layer H5

$$N^{(k)} = \tanh \left( b5^{(k)} + \sum_{m_2=0}^{n_{m_2}-1} \sum_{i_4=0}^{n_{i_4}-1} \sum_{j_4=0}^{n_{j_4}-1} H_{(i,j)}^{(k, m_2)} * FM4_{(i_4, j_4)}^{(m_2)} \right) \quad (10)$$

Keterangan variabel dan index pada *hidden layer H5* adalah sebagai berikut:

- $N$  : Neuron *hidden layer*
- $b5$  : Bias *hidden layer*
- $H$  : Bobot *hidden layer*
- $k$  : Index jumlah neuron *hidden layer*
- $n_k$  : Jumlah neuron *hidden layer*

#### 6) Output Layer O6

$$O^{(l)} = \tanh \left( b6^{(l)} + \sum_{k=0}^{n_k-1} W_{(k)}^{(l)} * N5_{(k)} \right) \quad (11)$$

Keterangan variabel dan index pada *neuron layer O6* adalah sebagai berikut:

- $O$  : Neuron *output layer*
- $b6$  : Bias *output layer*
- $W$  : Bobot *output layer*
- $l$  : Index jumlah neuron *output layer*
- $n_l$  : Jumlah neuron *output layer*

### E. Konfigurasi parameter Convolutional Neural Networks

Inisialisasi nilai *parameter* bobot dilakukan dengan membagi bilangan acak dari fungsi standar distribusi normal  $N(0,1)$  ( $mean=0$  dan  $variance=1$ ) dengan akar dari jumlah parameter bobot:

$$w(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-\mu)^2}{2\sigma^2}} \frac{1}{\sqrt{n}} \quad (12)$$

- $r$  : Bilangan acak (*random*)
- $n$  : Jumlah parameter bobot
- $\sigma$  : Standar deviasi
- $\mu$  : *Mean* atau ekspektasi

Sedangkan parameter bias diinisialisasi dengan angka nol.

## V. IMPLEMENTASI DAN HASIL UJI COBA

### A. Lingkungan Hardware dan Software

Lingkungan perancangan sistem dibangun dari dua lingkungan yaitu lingkungan *software* dan lingkungan *hardware*. Spesifikasi lingkungan perancangan sistem secara lengkap dapat dilihat pada tabel 2.

**Tabel 3.** Daftar metode *data augmentation*.

Lingkungan	Spesifikasi	
Hardware	Processor	Intel® Core™ i3 2.13 GHz
	RAM	3 GB
	WebCam	5MP 640x320 piksel
Software	Sistem Operasi	Windows 7 Ultimate 32 Bit
	Tools	Netbeans IDE 8.0.2

### B. Dataset Uji Coba Training

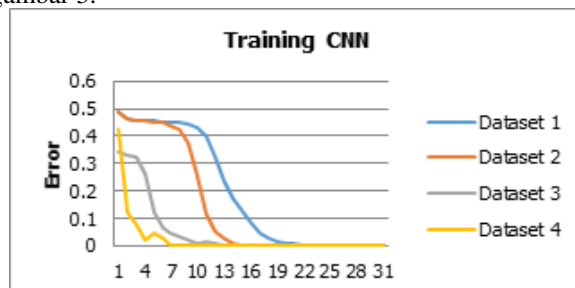
Dataset uji coba *training* dikumpulkan dari relawan berupa data gambar wajah dengan berbagai kondisi pencahayaan dan posisi wajah masing-masing individu. Dataset dibagi menjadi 6 bagian berdasarkan tujuan pengujian. Setelah dikumpulkan dataset dilakukan proses ekstraksi menggunakan metode *Extended Local Binary Pattern* dengan pengaturan radius diberi nilai 1 dan *neighbor* diberi nilai 15. Rincian keterangan daftar *dataset* disajikan pada tabel 3.

Tabel 4. *Dataset Uji Coba*.

Data	Perincian Tiap gambar	Kondisi
3 wajah laki-laki	10 gambar	Pencahayaan terang
3 wajah perempuan	10 gambar	Kurang pencahayaan
5 wajah laki-laki	36 gambar	Kurang pencahayaan
2 wajah laki-laki	70 gambar	Pencahayaan terang

### C. Uji Coba Training Model

Dataset yang telah dikumpulkan akan dilakukan uji coba *training* untuk mendapatkan model pengenalan wajah dengan benar sesuai *dataset* yang dimasukkan. Pada tahap *training* model, diperlukan input berupa *dataset* dan nama label. *Dataset* akan otomatis bersesuaian dengan nama label. Nama label yang bersesuaian direpresentasikan sebagai deretan target bipolar. Hasil akhir *training* Dataset 1 menghasilkan model 1, dataset 2 menghasilkan model 2, dataset 3 menghasilkan model 3, dan dataset 4 menghasilkan model 4. Kurva perubahan nilai *error* pada uji coba *training* dataset 1, 2, 3, dan 4 dengan memberikan nilai  $\alpha = 0.01$  pada setiap iterasi dapat dilihat pada gambar 3.

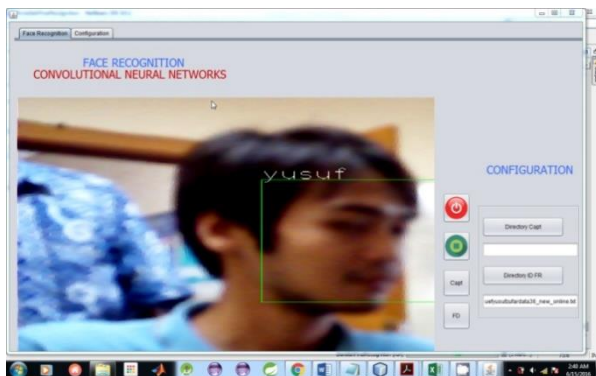


**Gambar 3.** Kurva konvergensi training dataset 1, 2, 3, dan 4.

Berdasarkan kurva *training* dataset 1,2, 3, dan 4 bahwa jaringan sudah berhasil konvergen ke suatu titik dan seiring berjalannya iterasi perubahan nilai *error* semakin lama semakin kecil sehingga nilai *error* mendekati konstan yaitu pada *error minimal* (*konvergen* dan stabil).

**D. Uji Coba Model**

Uji coba model dilakukan dengan menjalankan implementasi pengenalan wajah secara *real-time*. Proses pengenalan dijalankan dengan menangkap gambar wajah yang terdeteksi, dirubah kedalam *grayscale* 48x48. Gambar *grayscale* akan dilakukan preprocessing ELBP menjadi ekstraksi fitur gambar. Hasil ekstraksi akan masuk model dan dilakukan klasifikasi data. Keluaran klasifikasi model berupa nama label yang akan ditampilkan pada layar monitor tepatnya di atas kiri lokasi *bounding box* terdeteksi wajah.



**Gambar 4.** Hasil pengenalan wajah secara *real-time*.

Perhitungan tingkat akurasi seluruh model disajikan pada tabel 4.

**Tabel. 4.** Akurasi model 1, 2, 3, dan 4.

Model	Kondisi perekaman	Akurasi
1	Pencahayaan terang	91.67%
2	Kurang pencahayaan	80.15%
3	Kurang pencahayaan	88.82%
4	Pencahayaan terang	89.30%

Dari hasil perhitungan diambil kesimpulan bahwa model dapat melakukan klasifikasi dengan tingkat akurasi target minimal > 60%. Secara keseluruhan akurasi seluruh model didapatkan rata-rata akurasi:

$$\begin{aligned}
 \text{Akurasi} &= \frac{m_1 + m_2 + m_3 + m_4}{4} \\
 &= \left( \frac{91.67 + 80.15 + 88.82 + 89.30}{4} \right) \% \\
 &= 87.48\%
 \end{aligned}$$

**VI. KESIMPULAN DAN SARAN**

Dari hasil pembahasan pada bab sebelumnya, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Konstruksi model *Convolutional Neural Networks* dengan kedalaman 7 layer model konvolusi sebagai membangun jaringan antara lain *input layer*, *convolutional layer C1*, *pooling layer P2*, *convolutional layer C3*, *pooling layer P4*, *hidden layer H* dan *output layer F* berhasil mengklasifikasikan gambar wajah dengan rata-rata tingkat akurasi lebih dari 87%.

2. Penggunaan ekstraksi *Extended Local Binary Pattern* mampu mengatasi pengaruh intensitas cahaya pada gambar sehingga gambar yang terkena gangguan berupa intensitas cahaya dapat menghasilkan ekstraksi pola fitur yang hampir sama dengan gambar yang mendapatkan pencahayaan rendah dan konfigurasi inialisasi parameter bobot dengan menggunakan persebaran terdistribusi normal standar dapat mempercepat konvergensi dan kestabilan dibandingkan melakukan inialisasi secara acak.

Saran yang diberikan untuk perbaikan dan kelanjutan dari penelitian ini adalah:

1. Memilih *hardware* kamera digital yang memiliki resolusi tinggi dan memiliki fitur *autofocus* agar didapatkan gambar yang jelas walaupun obyek bergerak sehingga dapat meningkatkan kinerja jaringan sebelum masuk model baik untuk deteksi dan pengenalan.
2. Melakukan eksplorasi terhadap kedalaman jaringan *Convolutional Neural Networks*, jenis *pooling layer*, fungsi aktivasi, dan jenis metode klasifikasi.

**DAFTAR PUSTAKA**

- [1] T. Dunstone and N. Yager. *Biometric System and Data Analysis*: Springer, 2009.
- [2] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11), 2278–2324.
- [3] Gonzalez, R.C and Rafael E.W. *Digital Image Processing*. Prentice-Hall. Inc.. United State, America. 2008.
- [4] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. *Face Description with Local Binary Patterns: Application to Face Recognition*. vol. 28 no. 12, pp. 2037-2041, December 2006.
- [5] Mac Developer Library. *Performing Convolution Operations*[Online]. Available: <https://developer.apple.com/library/mac/documentation>.
- [6] Hubel, D. and Wiesel, T. (1968). *Receptive fields and functional architecture of monkey striate cortex*. *Journal of Physiology (London)*, 195, 215–243.
- [7] Fukushima, K. (1980). *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*. *Biological Cybernetics*, 36, 193–202.
- [8] Serre, T., Wolf, L., Bileschi, S., and Riesenhuber, M. (2007). *Robust object recognition with cortex-like mechanisms*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3), 411–426. Member-Poggio, Tomaso.