

# Pengenalan Ekspresi Wajah Menggunakan Arsitektur Resnet Pada Algoritma *Convolutional Neural Network* (CNN)

## *Facial Expression Recognition Using ResNet Architecture On Convolutional Neural Network Algorithm*

1<sup>st</sup> Muhammad Farhan Muntazhar

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

farhanabrar@student.telkomuniversity.  
ac.id

2<sup>nd</sup> Budhi Irawan

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

budhiirawan@telkomuniversity.ac.id

3<sup>rd</sup> Ashri Dinimaharawati

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

ashridini@telkomuniversity.ac.id

**Abstrak**—Pada saat manusia berinteraksi antara satu dengan yang lainnya, komunikasi mereka terjadi antara verbal dan nonverbal, Komunikasi nonverbal ini terjadi melalui ekspresi manusia yang mengandung emosi. Ekspresi manusia menyimpan banyak sekali informasi yang mengatakan keadaan orang tersebut, karena ekspresi manusia juga merupakan respon secara spontan. Emosi dasar yang membentuk ekspresi ada enam yaitu marah, senang, sedih, jijik, bahagia, kaget. Model *machine learning* yang dibuat menggunakan algoritma *convolutional neural network* dengan model arsitektur *ResNet* dapat melakukan training dengan dataset untuk mengidentifikasi ekspresi dasar yang telah disebut diatas. Sistem yang akan dibuat ini akan menerima input berupa foto ekspresi wajah dan mencoba mendeteksi ekspresi pada wajah tersebut. Pada penelitian ini akan digunakan dataset FER 2013 yang berisi sekitar 30.000 data yang terbagi atas data *training* dan *testing* dan *val*. Kemudian data juga terbagi kedalam tujuh kategori emosi dasar. Pada penelitian ini penulis mengimplementasikan arsitektur *ResNet* 50 kepada dataset FER 2013 dan berhasil mencapai akurasi latih sebesar 72,06% akurasi validasi sebesar 64,34% dan akurasi test sebesar 64,47%. Model yang dibuat juga berhasil mendeteksi dengan benar ekspresi marah, jijik, takut, senang, netral sebanyak, sedih, kaget dari total dataset sebanyak 35.887 data.

**Kata kunci** — *Facial Expression Recognition, Convolutional Neural Network, Deep Learning*

**Abstract**—When humans interact with each other, the communications that happen between them are verbal and non-verbal. Non-verbal communications occur through facial expressions that contain emotions, human facial expressions contain a lot of information about their current state. This is because that facial expression is a spontaneous response to the event or circumstances that they are in. There are six basic emotions that create facial

*expressions: anger, disgust, fear, sadness, happiness, and surprise. A machine learning model with convolutional neural network-based algorithms and residual network architecture can train the dataset and identify human facial expressions. The system that I created will take the input in the photo of facial expressions and then try to detect the facial expression in the input photo. In this final project, the dataset that I use is FER 2013 with 30.000 pictures split into train, test, and validation. This dataset also splits the data into subfolders of seven facial expressions including neutral. In this final project, I implemented the convolutional neural network algorithm with residual network architecture to the FER 2013 dataset. The accuracy of train data is 72,06%, the accuracy of validation is 64,34% and the accuracy of test data is 64.47%. The model I create successfully detects the expression of anger, disgust, fear, happiness, neutral, sad, and surprise from the total dataset of 35.887photo.*

**Keywords**— *Facial Expression Recognition, Convolutional Neural Network, Deep Learning*

### I. PENDAHULUAN

Wajah adalah salah satu bagian tubuh yang pertama kali dilihat, mata sebagai panca indra untuk melihat dan mulut sebagai bagian tubuh yang salah satu fungsinya adalah berbicara, dua bagian tubuh ini menjadi salah satu alat *input* dan *output* manusia. Selain itu wajah juga merupakan tempat mengekspresikan perasaan manusia, banyak sekali informasi yang bisa didapatkan melalui ekspresi wajah manusia seperti saat sedang bahagia, sedih, marah, dan ekspresi lainnya. Bahkan gerak-gerik wajah dapat mengindikasikan jika seseorang sedang berbohong atau tidak.

Emosi adalah respon kompleks dari psikologi

manusia, kebalikan dari respon rasional yaitu respon intuitive. Emosi dibagi kedalam enam kategori yang merupakan emosi dasar yaitu: marah, senang, terkejut, jijik, sedih, dan takut. Emosi dapat dilihat secara langsung melalui ekspresi wajah manusia, misalnya jika seseorang sedang merasa bahagia maka mulut akan tersenyum atau tertawa, dan sebaliknya

jika sedang merasa sedih cenderung bibir mereka akan cemberut. Dari salah satu fitur wajah seperti mulut saja kita dapat menentukan ekspresi dan emosi seseorang, namun tidak hanya satu fitur yang menjadi penentu ekspresi seseorang tapi seluruh fitur wajah menjadi penentu ekspresi seseorang.

Ekspresi wajah dan emosi saling berkaitan, ekspresi wajah dapat dijadikan salah satu cara sebuah komputer untuk mengenali perasaan manusia dan hal ini dapat meningkatkan efektifitas interaksi antara komputer dan manusia. *Feedback* berupa ekspresi wajah ini dapat digunakan untuk pengembangan aplikasi atau sebuah sistem tertentu seperti *video game* [10] dan pembelajaran [15]. Pada anak-anak, pengenalan ekspresi wajah ini dapat membantu memahami perasaan mereka sehingga sebagai orang dewasa bisa mengambil tindakan yang tepat untuk emosi yang dirasakan anak tersebut.

## II. KAJIAN TEORI

### A. Facial Expression Recognition

Kegiatan manusia tidak akan lepas dari komunikasi antara sama lain, salah satu komunikasi non-verbal yang juga terjadi adalah ekspresi wajah, ekspresi wajah dapat menjadi indikator atau salah satu cara untuk memahami keadaan emosi seseorang seperti marah, sedih, senang dan lain sebagainya. *Facial Expression Recognition* (FER) menjadi salah satu bidang pada *computer vision* yang memiliki banyak kemungkinan untuk masa depan seperti mendeteksi keadaan mental, keamanan, deteksi kebohongan, music yang cocok untuk *mood* dan lainnya. [13]

### B. Deep Learning

Berbeda dengan machine learning yang belajar menggunakan metode statistical, deep learning belajar dari sebuah dataset yang besar. Deep learning itu sendiri adalah bagian dari machine learning yang mencoba mengikuti cara manusia mendapatkan pengetahuan. Dalam deep learning memiliki sebuah neural network yang mensimulasikan cara kerja otak manusia[1].

### C. Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) merupakan salah satu algoritma dalam *Deep Learning*, CNN dapat dideskripsikan sebagai sebuah konsep pendeteksi fitur secara hirarki, CNN dapat mempelajari fitur-fitur yang sangat abstrak dan mendeteksi objek secara efisien. Dibanding dengan Neural Network sederhana algoritma CNN lebih baik dalam mendeteksi gambar karena memang ditujukan untuk mendeteksi gambar. Cara kerja dari CNN

terbagi kedalam beberapa lapisan (layers), ada tiga lapisan utama yaitu Convolutional Layer, Pooling Layer, Fully Connected layer. Fungsi dari Convolutional Layer adalah untuk ekstraksi fitur yang akan digunakan, Pooling layer berfungsi untuk membuat filter berdasarkan rule yang sudah ditetapkan, Fully Connected Layer adalah Multi Layer Perceptron (MLP) yang merupakan bagian dari Artificial Neural Network yang terdiri dari berbagai neuron yang terhubung dengan menghubungkan beban (weight). [2]

#### 1. Batch Normalization

Untuk membuat proses dari *Neural Network* menjadi lebih cepat dan lebih stabil, dilakukan proses *Batch Normalization*. Pada lapisan ini, operasi standarisasi dan normalisasi dilakukan pada *input* lapisan dari *output* lapisan sebelumnya. Pada *Batch Normalization* terjadi dua proses, yang pertama adalah menormalisasikan input dan kemudian *rescaling*.

#### 2. Pooling Layer

*Pooling layer* dilakukan karena pada *convolutional layer* mengambil posisi sebuah fitur secara tepat yang berarti perbedaan posisi pada dari fitur akan menghasilkan *feature map* yang berbeda [15]. Ukuran dari operasi *pooling* atau filter lebih kecil dari ukuran *feature map*. Ukuran filter ini hampir selalu 2 x 2 dengan *strides* dua pixel. Maka ukuran dari *feature map* akan dibagi dua dan jumlah pixel dari *Feature map* akan menjadi seperempat nya. Dua fungsi *pooling* yang umum digunakan adalah:

- a. *Average Pooling*: menghitung nilai rata-rata untuk setiap *feature map*
- b. *Max Pooling*: menghitung nilai tertinggi dari setiap *feature map*

#### 3. Activation Function

*Activation Function* berfungsi sebagai penentu apakah sebuah neuron akan aktif atau tidak. Artinya fungsi ini akan menentukan apakah *input* dari sebuah neuron itu penting atau tidak terhadap proses dengan menggunakan model matematika sederhana []. Peran utama dari fungsi aktivasi ini adalah mengubah *input* berbobot yang telah dijumlahkan menjadi sebuah *output* yang akan masuk ke dalam *hidden layer* berikutnya atau langsung menjadi *output*. [14]

##### a. Rectified Linear Unit

*Rectified Linear Unit* atau ReLU secara sederhananya adalah fungsi yang akan memberikan output

secara langsung jika input nya positif, jika tidak maka outputnya akan bernilai nol.

##### b. Softmax

Aktivasi dari *softmax* ini merupakan bentuk lain dari algoritma *logistic regression* yang memiliki tugas untuk melakukan klasifikasi, standar klasifikasinya adalah klasifikasi biner. *Softmax* memberikan hasil lebih baik karena memiliki interpretasi probabilitas yang lebih baik, aktivasi ini juga melakukan

perhitungan untuk semua label. Sehingga untuk klasifikasi *multiclass* aktivasi ini sangat baik.

4. Optimizer

Optimizer adalah sebuah fungsi atau algoritma yang memodifikasi atribut-atribut dari neural network seperti bobot dan learning rate yang akan membantu mengurangi loss dan menaikkan akurasi dari pelatihan. [4]

a. Stochastic Gradient Descent (SGD)

Optimizer SGD melakukan iterasi pada setiap data latih, kemudian SGD akan menghitung loss dan memperbaharui weight. Hal ini akan menyebabkan kurva yang tidak stabil karena terlalu sering melakukan perubahan weight. Namun optimizer ini relatif cepat dibanding optimizer lain.[4]

b. Adaptive Moment Estimation (Adam)

Tidak seperti SGD yang menggunakan satu learning rate, Adam memperbaharui learning rate untuk setiap bobot network. Adam sangat adaptif terhadap pelatihan Deep Neural Network dan direkomendasikan untuk menjadi optimizer utama. Adam memiliki runtime yang lebih cepat, kebutuhan memori yang lebih kecil, dan membutuhkan penyesuaian yang lebih sedikit dibanding algoritma optimasi lainnya.[4]

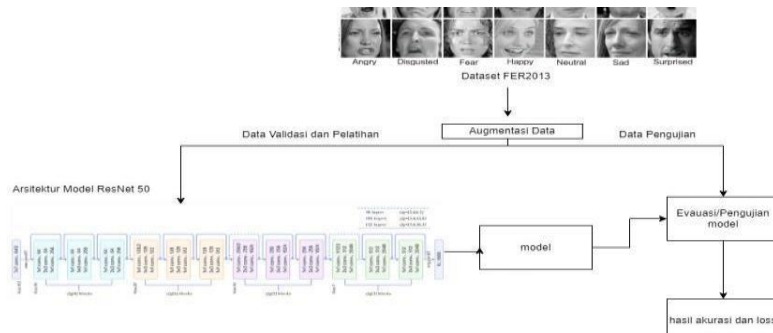
D. Residual Network

Kedalaman (*depth*) atau jumlah lapisan dari sebuah model sangat berpengaruh terhadap hasil dari model *Deep Learning*. Saat model *Deep Learning* yang lebih dalam mulai mengalami *convergence* maka akan muncul masalah degradasi yang artinya bertambahnya kedalaman model akan membuat akurasi menjadi jenuh dan meningkatkan error. Resnet memperkenalkan *Identity Shortcut Mapping* yang melewati satu lapisan atau lebih, yang berarti menambah lapisan tidak akan menghasilkan degradasi karena dapat menumpuk *identity mapping* yang menghasilkan performa yang sama. Hal ini mengindikasikan bahwa model yang lebih dalam tidak akan memiliki *error* yang melebihi lapisan pasangan yang lebih dangkal.[5]

III. METODE

A. Gambaran Umum Sistem

Sistem yang dibuat pada penelitian ini menggunakan algoritma Convolutional Neural Network (CNN) dengan arsitektur yang digunakan adalah Residual Network 50 layer atau bisa disebut ResNet50. Penulis juga melakukan beberapa modifikasi terhadap arsitektur ResNet50 agar lebih sesuai dengan dataset yang digunakan.

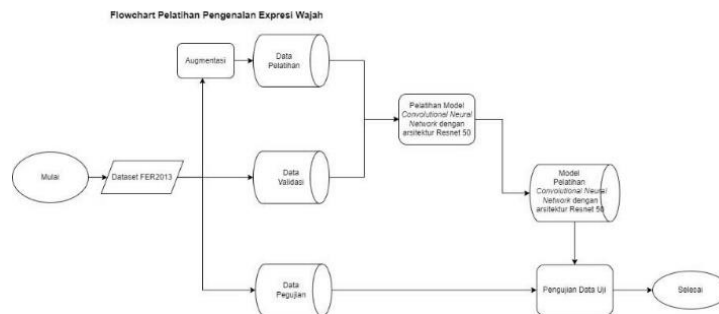


GAMBAR 3.1 GAMBARAN UMUM DESAIN SISTEM

Pada Penelitian ini input dataset akan masuk kedalam alur program seperti pada gambar 3.1. Alur akan dimulai dari pemecahan dataset yang akan dibagi menjadi data pelatihan, data validasi, dan data pengujian. Kemudian akan dilakukan proses augmentasi. Untuk data pelatihan dan validasi akan masuk ke proses pelatihan menggunakan arsitektur ResNet50 sedangkan untuk data pengujian akan

menevaluasi hasil dari pelatihan. Penulis kemudian akan membandingkan hasil akurasi dari arsitektur ResNet50 dengan ResNet50 modifikasi yang penulis kembangkan. Penulis juga akan membandingkan hasil akurasi terbaik dengan penelitian sebelumnya.

B. Perancangan Sistem



GAMBAR 3.2 PERANCANGAN SISTEM

Gambar 3.2 adalah flowchart untuk sistem yang dirancang pada penelitian ini. Sistem ini dirancang menggunakan algoritma CNN dengan arsitektur resnet dengan jumlah lapisan 50 layer. Pada sistem ini data yang digunakan sebagai input adalah dataset FER2013 yang telah dibagi menjadi 3 sub-folder yaitu train, val, dan test. Pada data train akan dilakukan augmentasi data terlebih dahulu, kemudian data train dan val untuk melakukan proses pelatihan menggunakan model pembelajaran resnet 50. Setelah itu akan dilakukan evaluasi model dengan data test.

## 1. Data

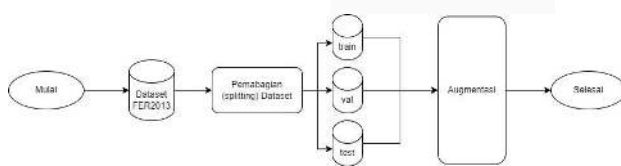
TABEL 3.1  
JUMLAH DATASET SETIAP KELAS

| Class (ekspresi) | Jumlah Data |
|------------------|-------------|
| Angry            | 4.953       |
| Disgust          | 547         |
| Fear             | 5.121       |
| Happy            | 8.989       |
| Neutral          | 6.198       |
| Sad              | 6.077       |
| Surprise         | 4.002       |

Tabel 3.1 berisi jumlah dataset FER2013 untuk setiap kelasnya dengan jumlah data paling banyak pada kelas *happy* atau senang dan paling sedikit pada *disgust* atau jijik.

## 2. Pemrosesan Awal

Proses selanjutnya adalah preproses, proses ini dilakukan sebelum masuk ke dalam proses pelatihan.



GAMBAR 3.3  
FLOWCHART PREPROCESSING

Penulis menggunakan *dataset Facial Expression Recognition 2013 (FER2013)* yang dapat diakses pada [kaggle.com](https://www.kaggle.com). Dataset ini berisi 35.887 gambar berukuran 48 x 48 pixel dengan format .jpg yang terbagi kedalam tujuh kelas yakni *angry, disgust, fear,*

*happy, neutral, sad, dan surprise*. Tujuh kelas ini merupakan ekspresi dasar wajah manusia.

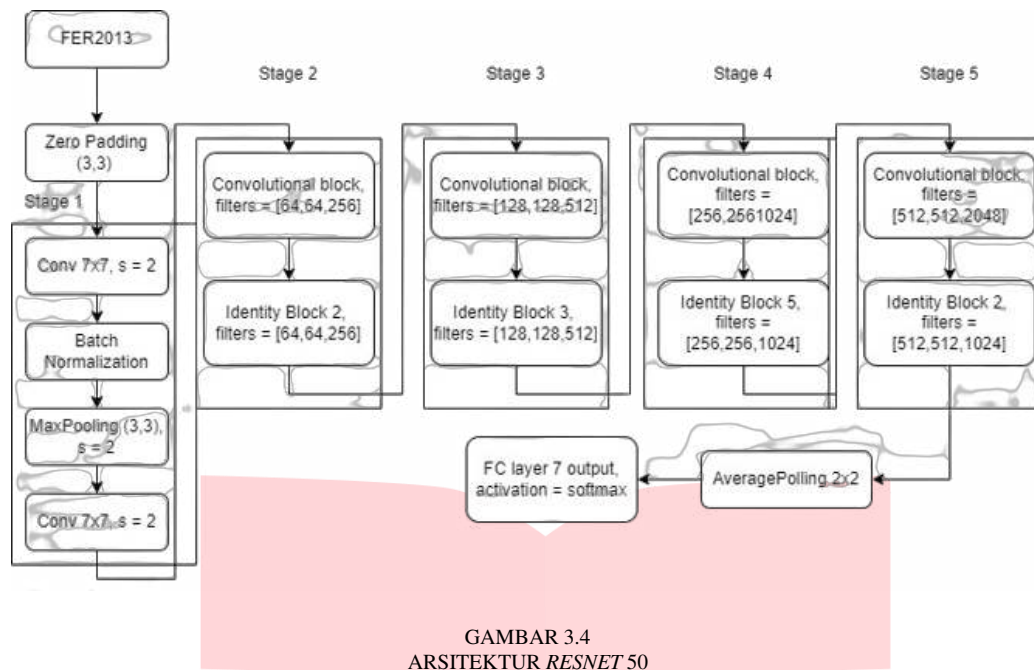
Gambar 3.2 merupakan augmentasi gambar untuk menambahkan fitur pada gambar yang didapat dari data akan lebih banyak pada proses pelatihan. Pada pelatihan *Neural Network* khususnya *Deep Learning*, dataset yang digunakan harus memiliki variasi fitur agar meningkatkan akurasi pada klasifikasi. Augmentasi

ini akan menambahkan fitur-fitur yang ada pada gambar dataset sehingga model akan memiliki banyak fitur yang bisa di ekstrak dan menjadi pertimbangan dalam proses klasifikasi. Pada penelitian ini penulis menggunakan augmentasi sebagai berikut:

- zoom range sebesar 0.2
- melakukan horizontal flip
- melakukan *width shift range* sebesar 0.2
- melakukan *height shift range* sebesar 0.2

## 3. Model Pembelajaran





GAMBAR 3.4  
ARSITEKTUR RESNET 50

Gambar 4.1 adalah flowchart dari sistem yang diimplementasikan dalam penelitian ini. Arsitektur *resnet 50* yang dibuat oleh penulis menggunakan parameter *filter*, *kernel size*, dan *stride* yang sama dengan arsitektur *state of the art* nya, namun untuk input size dan jumlah kelas pada output *fully connected* layer disesuaikan dengan *dataset FER2013*. Untuk input size nya sebesar  $48 \times 48$  sedangkan output kelasnya sebanyak 7 kelas. Sebelum data masuk kedalam proses pembelajaran menggunakan algoritma CNN dengan model pembelajaran *resnet 50*, *dataset* akan memasuki proses augmentasi data untuk memperbanyak variasi fitur dalam gambar sebagai berikut:

Augmentasi data: data akan di *zoom* sebesar 0.3 kemudian dibalik secara horizontal. Dilakukan juga *heightshift range* dan *width shift range* sebesar 0.2.

Setelah melalui augmentasi data, *dataset* akan memasuki proses pelatihan dengan model pembelajaran *resnet*

50, model pembelajaran ini dibagi kedalam 5 tahap sebagai berikut:

**Stage 1:** konvolusi yang memiliki 64 filter dengan ukuran kernel  $7 \times 7$  dan stride  $2 \times 2$ , kemudian masuk ke proses *Batch normalization* untuk mempercepat dan menstabilkan proses. Untuk fungsi aktifasinya adalah ReLU dan terakhir dilakukan *Max Pooling* dengan ukuran kernel  $3 \times 3$  dan stride  $2 \times 2$ .

**Stage 2:** konvolusi blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah : 64, 64, 256. Diikuti dengan 2 *identity* blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah: 64, 64, 256.

**Stage 3:** konvolusi blok dengan arsitektur

*bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah : 128, 128, 512. Diikuti dengan 3 *identity* blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah: 128, 128, 512.

**Stage 4:** konvolusi blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah : 256, 256, 1024. Diikuti dengan 5 *identity* blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah: 245, 256, 1024.

**Stage 5:** konvolusi blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah : 512, 512, 2048. Diikuti dengan 2 *identity* blok dengan arsitektur *bottleneck* dengan ukuran kernel  $1 \times 1$ ,  $3 \times 3$ , dan  $1 \times 1$ . Filter yang digunakan pada Conv2D secara berurut adalah: 512, 512, 2048.

Setelah itu akan masuk ke proses *Max Pooling* dengan ukuran  $2 \times 2$ , dan *flatten*. Pada bagian *fully connected layer* menggunakan output 7 klasifikasi dan aktifasi *softmax*.

#### IV. HASIL DAN PEMBAHASAN

##### A. Skenario Pengujian

Model yang telah dirancang akan menggunakan parameter yang berbeda-beda. Parameter yang digunakan adalah *epoch*, *batch size*, *learning rate*, dan *optimizer*. Penulis menggunakan *epoch* sebanyak 50, 80 dan 100, *learning rate* sebesar 0.1, 0.001, dan 0.0001, *batch size* sebesar 32, 64, dan 128. Sedangkan untuk optimizer penulis menggunakan SGD dan Adam. Selain *hyperparameter* yang telah disebutkan, penulis juga melakukan pengujian terhadap pemisahan

ratio *train*, *test*, dan *val* dataset yang berbeda. Penulis memisah dengan rasio *train* sebanyak 80 %, *test* 10 %, dan *val* 10 %. Kemudian skenario berikutnya

penulis memisah dataset dengan rasio *train* sebanyak 70 %, *test* 10 %, dan *val* 20 %.

TABEL 4.1  
SKENARIO PEMBAGIAN DATASET PERTAMA

| Class    | Train 80% | Val 10% | Test 10% |
|----------|-----------|---------|----------|
| Angry    | 3.995     | 491     | 491      |
| Disgust  | 436       | 55      | 55       |
| Fear     | 4.097     | 528     | 528      |
| Happy    | 7.215     | 897     | 897      |
| Neutral  | 4.965     | 626     | 626      |
| Sad      | 4.830     | 594     | 594      |
| Surprise | 3.171     | 416     | 416      |

Pada table 4.1 berisi jumlah dataset dengan rasio pembagian untuk *train* sebesar 80 %, *val* sebesar 10 %, dan *test* sebesar 10%.

TABEL 4.2  
SKENARIO PEMBAGIAN DATASET KEDUA

| Class    | Train 70 % | Val 20 % | Test 10% |
|----------|------------|----------|----------|
| Angry    | 3.714      | 990      | 491      |
| Disgust  | 410        | 110      | 55       |
| Fear     | 3.840      | 1.024    | 528      |
| Happy    | 6.741      | 1.797    | 897      |
| Neutral  | 4.684      | 1.239    | 626      |
| Sad      | 4.557      | 1.215    | 594      |
| Surprise | 3.001      | 800      | 416      |

Pada table 4.2 berisi jumlah dataset dengan rasio pembagian untuk *train* sebesar 70 %, *val* sebesar 20 %, dan *test* sebesar 10%.

Tabel 4.3 Skenario kombinasi parameter 1

| Epoch | Optimizer | Batch_Size | Learning_Rate |
|-------|-----------|------------|---------------|
| 50    | Adam      | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 64         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 128        | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
| 80    | Adam      | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 64         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 128        | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |

|     |     |       |
|-----|-----|-------|
| 100 | 64  | 0.1   |
|     |     | 0.01  |
|     |     | 0.001 |
|     | 128 | 0.1   |
|     |     | 0.01  |
|     |     | 0.001 |

Pada tabel 4.3 berisi kombinasi dari parameter *epoch*, *batch size* dan *learning rate* menggunakan optimizer *Adam*.

TABEL 4.4  
SKENARIO KOMBINASI PARAMETER 2

| Epoch | Optimizer | Batch_Size | Learning_Rate |
|-------|-----------|------------|---------------|
| 50    | SGD       | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 64         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 128        | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
| 80    | SGD       | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 64         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 128        | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
| 100   | SGD       | 32         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 64         | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |
|       |           | 128        | 0.1           |
|       |           |            | 0.01          |
|       |           |            | 0.001         |

Pada tabel 4.4 berisi kombinasi dari parameter *epoch*, *batch size*, dan *learning rate* menggunakan optimizer SGD

Setelah menentukan kombinasi parameter tahap berikutnya adalah melakukan pengujian untuk:

- a. Menganalisis hubungan tiap parameter.
- b. Membandingkan akurasi dari penelitian sebelumnya dengan akurasi milik penulis.

**B. Hasil Pengujian**

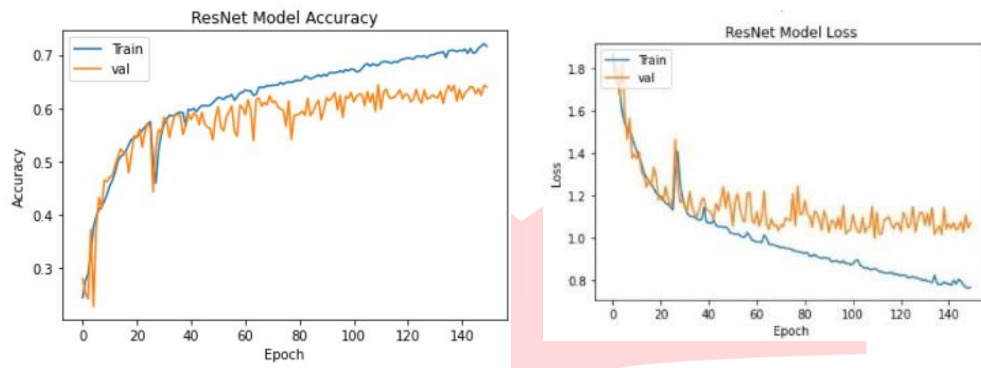
Penulis melakukan pengujian berdasarkan kombinasi parameter yang tertera pada tabel 4.3 dan 4.4. Penulis masih belum mendapatkan model dengan akurasi untuk melampui penelitian berikutnya. Berdasarkan analisis penulis terhadap parameter yang telah diuji, penulis mencoba melakukan pelatihan dengan kombinasi parameter baru seperti pada tabel 4.5.

|           |                 |
|-----------|-----------------|
| Parameter | Nilai Parameter |
|-----------|-----------------|

TABEL 4.5  
PENGUJIAN BEST

|               |                                       |
|---------------|---------------------------------------|
| Epoch         | 150                                   |
| Batch Size    | 32                                    |
| Learning Rate | 0.0005                                |
| Optimizer     | Adam                                  |
| Ratio Dataset | <i>Train 80 %, Val 10 %, Test 10%</i> |

PARAMETER  
MODEL



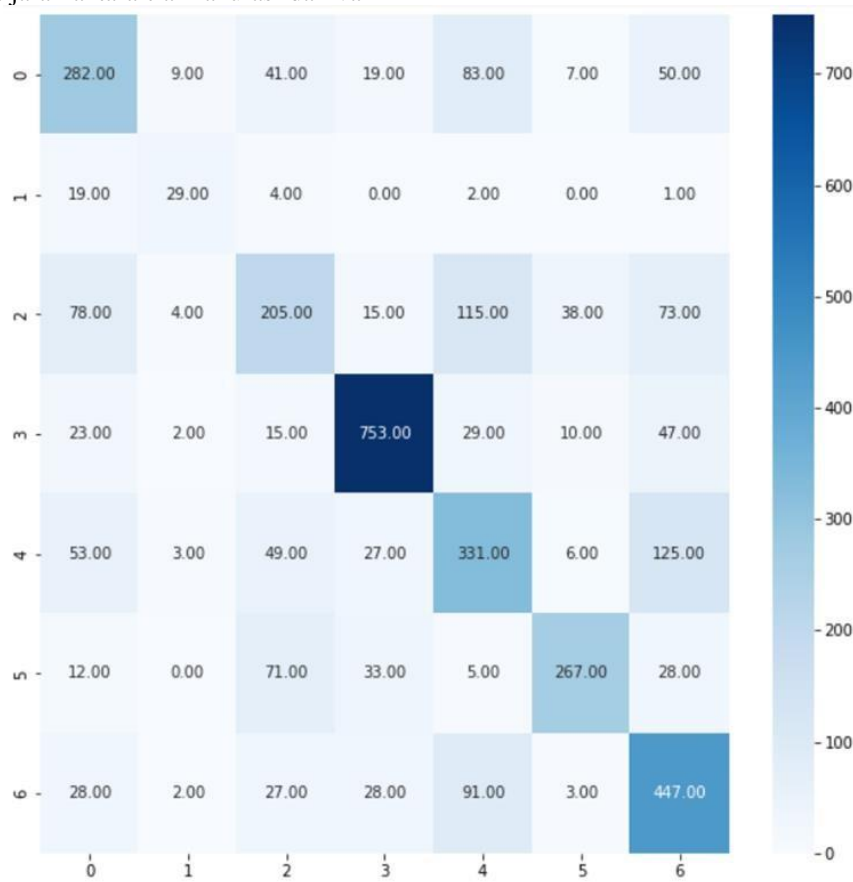
GAMBAR 4.5  
GRAFIK DARI AKURASI DAN LOSS





Pada gambar 4.5 ditunjukkan grafik dari Akurasi dan loss pada model pembelajaran yang menggunakan parameter seperti pada Tabel 4.5. Grafik validasi yang tidak terlalu menyimpang dari grafik train menandakan bahwa grafik tidak mengalami *overfitting* atau *underfitting*. *Overfitting* adalah kasus dimana train akurasi akan terus bertambah sedangkan validasi akurasi tidak bertambah sehingga jarak antara train akurasi dan val

akurasi jauh. Sedangkan untuk *underfitting* kebalikan dari *overfitting* yang berarti nilai val akurasi yang terus membesar sedangkan train akurasi tidak membesar. Penulis berhasil mencapai akurasi *train* sebesar 72.06%, akurasi validasi sebesar 64.34%, dan akurasi *test* sebesar 64.47%.

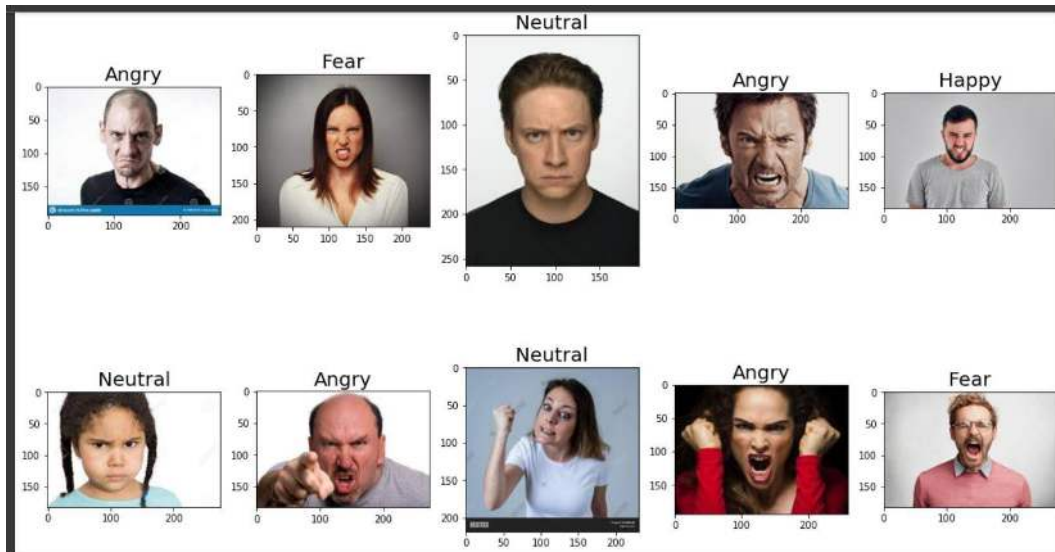


GAMBAR 4.6  
CONFUSION MATRIX

Pada gambar 4.6 dapat dilihat *confusion matrix* hasil prediksi dengan nilai sebenarnya, kolom yang terletak pada bagian diagonal adalah yang terprediksi benar sesuai dengan kelas ekspresinya. Untuk nilai tertinggi ada pada ekspresi dengan nomor 3 yaitu ekspresi senang, sedangkan untuk yang terendahnya ada di nomor 1 yaitu ekspresi jijik. Penulis

melakukan pengujian untuk setiap ekspresi wajah dengan 10 gambar tiap ekspresinya.

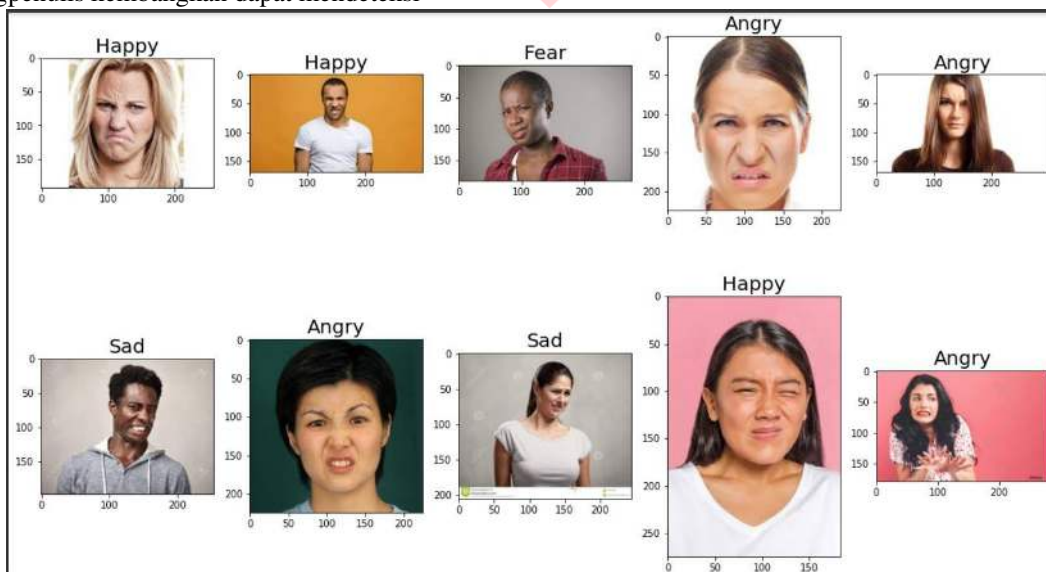
1. Angry



GAMBAR 4.7  
PENGUJIAN PADA EKSPRESI *ANGRY*

Pada gambar 4.7 penulis melakukan pengujian pada 10 gambar ekspresi marah. Dari 10 gambar, model yang penulis kembangkan dapat mendeteksi

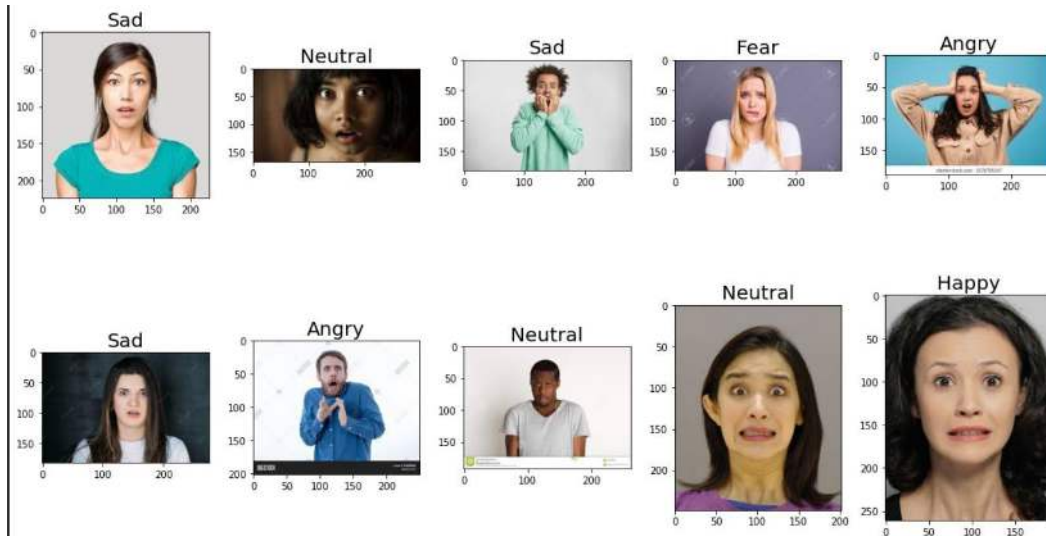
empat gambar dengan tepat.  
2.Disgust



GAMBAR 4.8  
PENGUJIAN PADA EKSPRESI *DISGUST*

Pada gambar 4.8 penulis melakukan pengujian pada 10 gambar ekspresi jijik. Dari 10 gambar, model yang penulis kembangkan masih

belum berhasil mendeteksi ekspresi jijik.  
3.Fear



GAMBAR 4.9  
PENGUJIAN PADA EKSPRESI FEAR

Pada gambar 4.9 penulis melakukan pengujian pada 10 gambar ekspresi takut. Dari 10 gambar, model yang penulis kembangkan hanya

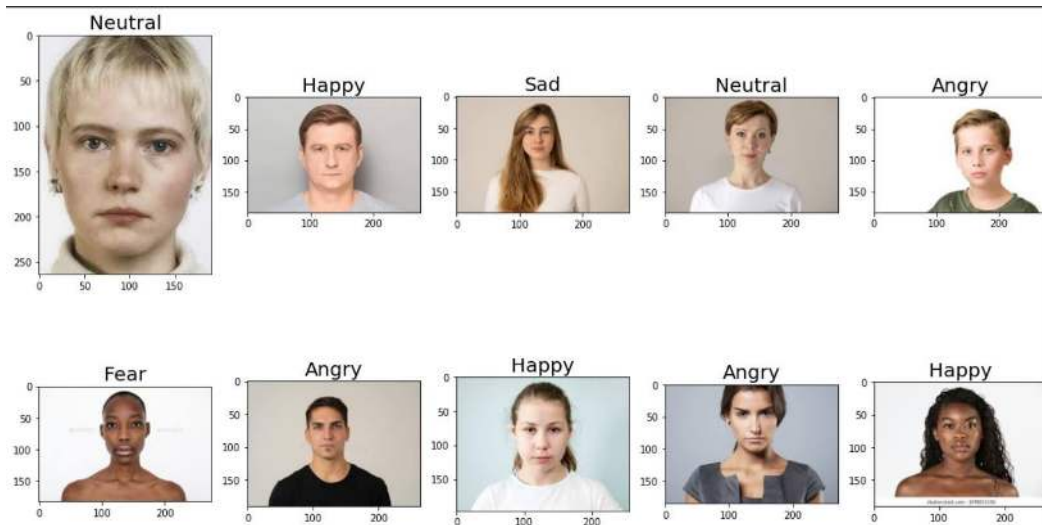
berhasil mendeteksi satu gambar dengan tepat.  
4.Happy



GAMBAR 4.10  
PENGUJIAN PADA EKSPRESI HAPPY

Pada gambar 4.10 penulis melakukan pengujian pada 10 gambar ekspresi senang. Dari 10 gambar,model yang penulis kembangkan hanya

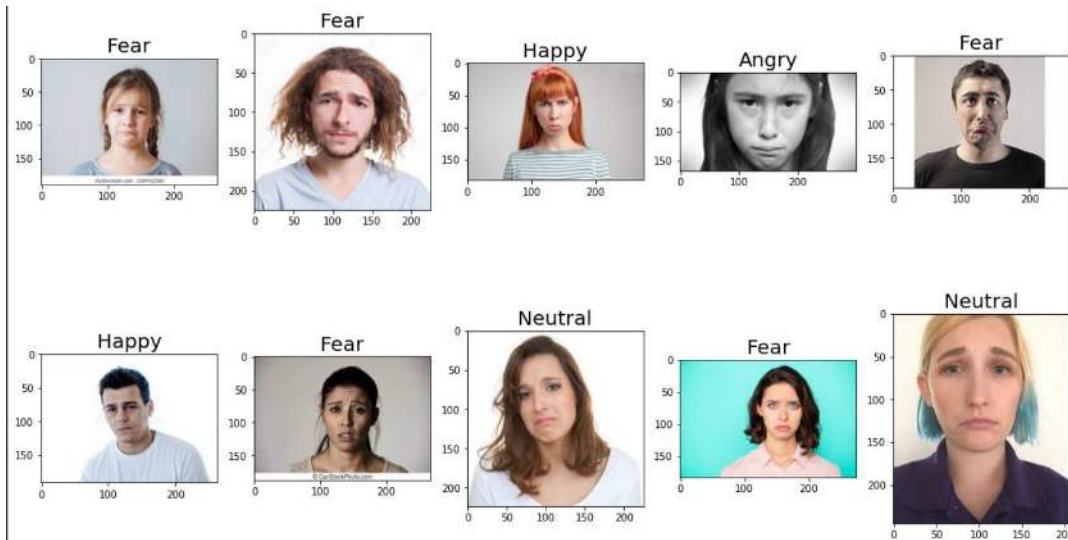
berhasil mendeteksi tiga gambar dengan tepat.  
5. 5. Neutral



GAMBAR 4.11 PENGUJIAN PADA EKSPRESI *NEUTRAL*

Pada gambar 4.11 penulis melakukan pengujian pada 10 gambar ekspresi netral. Dari 10 gambar, model yang penulis kembangkan hanya berhasil mendeteksi dua gambar dengan tepat.

6. Sad

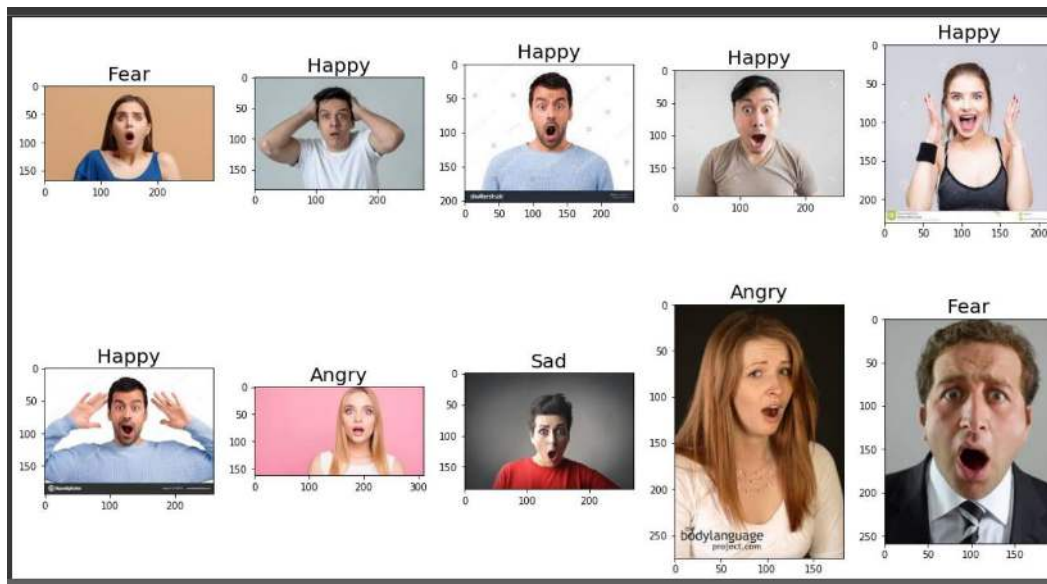


GAMBAR 4.12 PENGUJIAN PADA EKSPRESI *SAD*

Pada gambar 4.12 penulis melakukan pengujian pada 10 gambar ekspresi sedih. Dari 10 gambar, model yang penulis kembangkan masih belum berhasil mendeteksi ekspresi sedih.



## 7. Surprise



GAMBAR 4.13  
PENGUJIAN PADA EKSPRESI SURPRISE

Pada gambar 4.13 penulis melakukan pengujian pada 10 gambar ekspresi tekejut. Dari 10 gambar, model yang penulis kembangkan masih belum berhasil mendeteksi ekspresi tekejut.

Pengujian pada gambar baru masih berhasil mendeteksi semua gambar dengan benar, namun pada *confusion matrix* yang menjelaskan tentang korelasi antara nilai yang diprediksi dengan nilai sebenarnya banyak data yang diprediksi dengan benar. Hal ini terjadi karena pada gambar baru ini formatnya berbeda dengan dataset, sedangkan data yang diprediksi untuk menampilkan *confusion matrix* memiliki format yang sama dari dataset karena gambar itu berasal dari dataset tapi tidak digunakan pada proses pelatihan dan validasi.

## V. KESIMPULAN

Dari hasil pengujian dan analisis terhadap penelitian ini, dapat ditarik kesimpulan sebagai berikut:

1. Sistem yang dirancang dan dibangun masih belum bisa mendeteksi ekspresi wajah secara maksimal.
2. Hasil akurasi yang dicapai dengan kombinasi parameter *epoch* sebesar 150, *batch size* sebesar 32, *learning raet* sebesar 0.0005, menggunakan *optimizer Adam* dan rasio dataset *train* sebesar 80%, *val* sebesar 10 %, dan *test* sebesar 10% pada model ini untuk *train* akurasi sebesar 72,06% *val* akurasi sebesar 64,34% dan *test* akurasi sebesar 64,47%

## REFERENSI

- [1] Y. Afriansyah, R. A. Nugrahaeni dan A. L. Prasasti, "Facial Expression Classification for User Experience Testing Using K-Nearest Neighbor," dalam *2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, Bandung, 2021.
- [2] U. Astrashab, *Facial expression dataset image folders (fer2013)*, kaggle, 2020.
- [3] J. Brownlee, "A Gentle Introduction to Pooling Layers for Convolutional Neural Networks," *Machine Learning Mastery*, 05 07 2019. [Online]. Available: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>. [Diakses 20 08 2022].
- [4] R. Desmukh dan V. Jagtap, "A Comprehensive Survey on Techniwues for Facial Emotion Recognition," *International Journal of Computer Science and Information Security*, vol. 15, no. 3, pp. 219-224, 2017.
- [5] K. He, X. Zhang, S. Ren dan J. Sun, "Deep Residual Learning for Image Recognition," *arXiv*, pp. 1-12, 2015.
- [6] F. A. Isman, A. L. Prasasti dan R. A. Nugrahaeni, "Expression Classification For User Experience Testing Using Convolutional Neural Network," dalam *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, Bandung, 2021.
- [7] G. M, "Deep Learning Optimizer," *Toward Datascience*, 18 11 2020. [Online]. Available: <https://towardsdatascience.com/deep->

- learning-optimizers-436171c9e23f.  
[Diakses 23 08 2022].
- [8] T. M, I. M. N, R. I. V, C. T. K, M. J dan C. A, "Enhancing Game with Facial Expression Recognition as Dynamic Balancing," dalam *4th International Conferences on Computer Science and Computational Intelligence*, 2019.
- [9] A. P. a. M. P. M. Zaki, "ALEXNET ARCHITECTURE AND FUZZY ANALYSIS ON TALENT JUDGE DECISION PREDICTION BASED ON FACIAL EXPRESSION," *Jurnal Riset Informatika*, vol. 4, no. 4, pp. 341-346, 2022.
- [10] G. D. L. Mishra C, "Deep Machine Learning And Neural Network: an overview," *International Journal of Artificial Intelligence*, pp. 1-8, 2017.
- [11] J. Moolayil, *Learn Keras for Deep Neural Network*, Canada: Apress, 2019.
- [12] S. R, "Dicoding," Dicoding, 09 10 2021. [Online]. Available: <https://www.dicoding.com/blog/mengenal-deep-learning/>. [Diakses 17 12 2021].
- [13] A. Rastogi, "ResNet50," Dev Genius, 14 03 2022. [Online]. Available: <https://blog.devgenius.io/resnet50-6b42934db431>. [Diakses 04 08 2022].
- [14] Z. S, C. N dan H. C. H, "Facial Expression Versus Words: Unlocking Complex Emotional Response of Residence Towards Tourist," *Tourism Management*, pp. 1-3, 2021.
- [15] S. Singh dan F. Nasoz, "Facial Expression Recognition with Convolutional Neural Networks," *IEEE*, pp.324-328, 2020.
- [16] P. Utami, R. Hartanto dan I. Soesantri, "A Study on Facial Recognition in Assessing Teaching Skills: Dataset and Methods," dalam *The Fifth Information System Internationals Conference 2019*, 2019.
- [17] W. Y dan Q. W, "Facial Expression Recognition Based on Improved Deep Belief Network," dalam *AIP Conference Proceedings*, 2021.
- [18] Y. Zhong, S. Qiu, X. Luo, Z. Meng dan J. Liu, "Facial Expression Recognition Based on Optimzied ResNet," *2nd Wold Symposium on Artificial Intelligence*, pp. 84-91, 2020.



