

Pengenalan Ekspresi Wajah Menggunakan Arsitektur *Xception* Pada Algoritma *Convolutional Neural Network* (CNN) *Facial Expression Recognition Using Xception Architecture On Convolutional Neural Network Algorithm*

1st Fadel Anfasha Putra
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
fadelfanfashap@student.telkomuniversity.ac.id

2nd Budhi Irawan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
budhiirawan@telkomuniversity.ac.id

3rd Ashri Dinimaharawati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ashridini@telkomuniversity.ac.id

Abstrak—Secara alami manusia menggunakan ekspresi wajah untuk berinteraksi dan memperlihatkan emosi mereka. Terkadang, kita sulit untuk mengidentifikasi emosi seseorang, maka dari itu perlu diciptakan sebuah sistem yang dapat mendeteksi emosi seseorang menggunakan kecerdasan buatan. Sistem tersebut dibuat karena memiliki banyak kegunaan salah satunya untuk interaksi antara manusia dengan sistem. Terdapat tujuh kategori ekspresi dasar manusia yaitu marah, sedih, senang, jijik, takut, terkejut, netral. Dengan menggunakan kecerdasan buatan lebih spesifiknya menggunakan pendekatan *Deep Learning* dengan algoritma *Convolutional Neural Network* (CNN), sistem yang akan dibuat dapat mengklasifikasi ekspresi wajah. Pada penelitian ini akan menggunakan citra digital yang akan diinput kedalam sistem, kemudian sistem akan dilatih menggunakan data tersebut dengan mengekstraksi fitur dari citra digital yang diinput kemudian dari fitur tersebut akan di klasifikasi menurut ciri yang terdapat pada masing-masing ekspresi wajah. Untuk membuat sistem tersebut, digunakan dataset FER2013 sebagai bahan mentah untuk melatih sistem. Pada penelitian sebelumnya dengan menggunakan dataset FER2013 dan algoritma CNN, sistem dapat mencapai akurasi uji 65.2%. Sehingga, tujuan dari penelitian ini yaitu untuk membandingkan dan meningkatkan akurasi pada sistem yang akan dibangun. Karena semakin besar akurasi yang dimiliki sistem, semakin efektif dalam mendeteksi emosi seseorang melalui wajahnya. Hasil uji akurasi pada penelitian ini mencapai 66.92%.

Kata Kunci— *ekspresi wajah, deep learning, FER2013, CNN, akurasi*

Abstract—Naturally, human uses facial expression to interact and show their emotion. At times it's hard to identify someone emotion, therefore there should be a system made that can detect someone emotion using artificial intelligence. The system were made because of its benefits, such as human interaction with the system itself. There are seven categories in human emotion, anger, sadness, joy/happiness, disgust, fear, surprise and neutral. By using artificial intelligence, with using *Deep Learning* approach with *Convolutional Neural Network* algorithm in specific, the system made can classifies facial expression. This research will use digital imaging that will be inputted into the system, then the system will be trained to use the data by extracting features from the digital imaging input and classified the features by each facial expression characteristics. To make the system, FER2013 dataset was used as the raw material and train the system. On previous research that used FER2013 and CNN algorithm, the system could reach accuracy up to 62.5%. Hence, the aim of this research is to compare and increase system accuracy that will be built. The higher accuracy of the system would make the system to effectively detect human emotion thorough their face. The accuracy test on this study could achieve up to 66.92% in accuracy.

Keywords— *accuracy, CNN, deep learning, facial expression, FER2013*

I. PENDAHULUAN

Manusia secara alami dan intuitif menggunakan ekspresi wajah untuk berkomunikasi dan menunjukkan emosinya dalam berinteraksi sosial.

Ekspresi wajah merupakan salah satu bentuk manusia untuk mengekspresikan suatu hal. Ekspresi dapat dilihat dengan berbagai cara, diantaranya dari cara bicara, gerak tubuh, dan ekspresi pada wajah. Dari berbagai cara tersebut ekspresi wajah yang paling

mudah untuk menentukan emosi atau ekspresi seseorang [1].

Ekspresi yang bisa diklasifikasikan adalah marah, takut, sedih, senang, netral, jijik, dan terkejut [2]. Pendeteksian ini dilakukan dengan melihat fitur pada wajah yaitu mata, bibir, hidung, dan alis. Selain itu, ekspresi juga akan menghasilkan emosi yang berbeda dari orang yang berbeda [3].

Kecerdasan buatan dapat digunakan untuk membantu proses identifikasi ekspresi pada wajah. Salah satu algoritma kecerdasan buatan yang dapat digunakan adalah Deep Learning. Deep Learning merupakan sebuah teknik dari machine learning yang dimana sebuah komputer diajarkan untuk melakukan sesuatu hal natural yang biasa dilakukan oleh manusia, contohnya seperti mendeteksi wajah, mendeteksi objek, dan masih banyak lainnya. Deep Learning dapat memproses unstructured data seperti teks serta gambar, dapat mengotomatisasi proses ekstraksi fitur tanpa melakukan proses pelabelan manual, memberikan hasil akhir yang berkualitas, mengurangi biaya operasional, serta melakukan manipulasi data dengan lebih efektif [4].

Penelitian terkait ekspresi emosi pada wajah terus dilakukan agar tercipta sebuah sistem pengenalan wajah menjadi subjek yang semakin populer, serta dapat diandalkan, karena sistem pengenalan wajah lebih akurat, presisi dan aman [5]. Pada penelitian sebelumnya hasil yang didapatkan yaitu 65,2% untuk akurasi uji dengan model Xception, 61,42% untuk akurasi uji dengan model InceptionV3, dan 54% untuk akurasi uji dengan model Simple CNN [6]. Pada penelitian ini penyusun melakukan pengenalan ekspresi wajah dengan menggunakan metode Convolutional Neural Network (CNN) untuk ekstraksi fitur dari dataset [7]. Arsitektur Xception dengan Augmentasi pada dataset FER2013 agar memiliki variasi data yang lebih banyak sehingga model dapat mengenali ekspresi pada wajah dengan lebih baik.

II. KAJIAN TEORI

A. Facial Expression Recognition

Pengenalan ekspresi pada wajah (*Facial Expression Recognition*) dapat ditemukan dalam berbagai bidang seperti interaksi komputer manusia (*Human Computer Interaction*), identifikasi biometrik, video kamera pengintai, analisis emosi manusia, dan masih banyak lagi. Pengenalan ekspresi pada wajah terdiri dari tiga bagian utama yaitu prapemrosesan set gambar wajah, mengekstraksi fitur wajah, dan klasifikasi wajah [8].

B. Deep Learning

Deep Learning merupakan bagian dari *machine learning* yang dimana algoritmanya ini

terinspirasi dari struktur dan fungsi dari susunan *artificial neural network* (ANN). Manfaat dari *deep learning* yaitu, dapat memproses *unstructured data* seperti teks serta gambar, dapat mengotomatisasi proses ekstraksi fitur tanpa melakukan proses pelabelan manual, memberikan hasil akhir yang berkualitas, mengurangi biaya operasional, serta melakukan manipulasi data dengan lebih efektif. *Deep Learning* itu sendiri terbagi menjadi empat jenis yaitu *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), *Long Short-Term Memory Network* (LSTM), *Self Organizing Maps* (SOM) [4].

C. Convolutional Neural Network

Sebagai salah satu algoritma *Deep Learning*, *Convolutional Neural Network* (CNN) adalah algoritma yang dikembangkan dari *Multi Layer Perceptron* (MLP) yang biasanya digunakan untuk mengolah data berupa suara atau gambar. CNN menggunakan metode *Supervised Learning* untuk mengklasifikasi data yang terlabel. CNN dapat digunakan juga untuk mendeteksi dan mengenali objek pada sebuah image. CNN dapat terdiri dari puluhan hingga ratusan lapisan yang tiap lapisannya dalam belajar mendeteksi gambar. Lapisan-Lapisan CNN terdiri dari 3 *neuron* yaitu lebar, tinggi, dan kedalaman (*width, height, depth*) [9].

D. Convolutional Layer

Convolutional layer memiliki peran penting dalam operasi CNN. *Layer* parameter berfokus pada penggunaan kernel. Dalam dimensi spasial, kernel ini biasanya kecil, tetapi tersebar di seluruh kedalaman *input*. Ketika data mengenai *convolutional layer*, *layer* menggulung setiap *filter* yang melintasi dimensi spasial *input* untuk menghasilkan peta aktivasi 2D. Setiap kernel akan memiliki peta aktivasi yang sesuai, yang akan ditumpuk sepanjang dimensi *depth* untuk membentuk volume *output* penuh dari *convolutional layer* [10]. Secara signifikan, *Convolutional layer* juga mampu mengurangi kompleksitas model melalui optimasi *output* yang dapat dioptimalkan menggunakan tiga parameter *depth*, ukuran *stride*, dan opsi penggunaan *zero padding* [11]. *Convolutional layer* digunakan untuk mengekstrak gambar untuk mendapatkan fitur dari gambar tersebut [12].

E. Depthwise Separable Convolution

Depthwise Separable Convolution adalah konvolusi yang terdiri dari dua bagian, yaitu *Depthwise Convolution* dan *Pointwise Convolution*. Dinamakan *Depthwise Separable Convolution* karena konvolusinya tidak hanya hanya berhubungan dengan

dimensi spasial, melainkan dimensi kedalamannya juga [13].

F. *Adaptive Moment Estimation* (Adam)

Adam adalah algoritma optimisasi yang cara kerjanya yaitu dengan memperbaharui nilai *learning rate* pada setiap jaringan bobot secara individual. Adam berasal dari kata *adaptive moment estimation*. Adam memiliki banyak kelebihan, oleh karena itu algoritma ini banyak digunakan. Adam memiliki keunggulan untuk menentukan batas pembaharuan nilai *learning rate* [14].

G. *Stochastic Gradient Descent*

Stochastic Gradient Descent (SGD) adalah algoritma optimisasi yang cara kerjanya yaitu dengan melakukan pembaharuan pada bobot pada setiap data pelatihan. SGD melakukan pembaharuan setelah menghitung nilai kerugian dari pelatihan tersebut. Karena bobot yang terlalu sering diperbaharui membuat kurva menjadi tidak stabil [15].

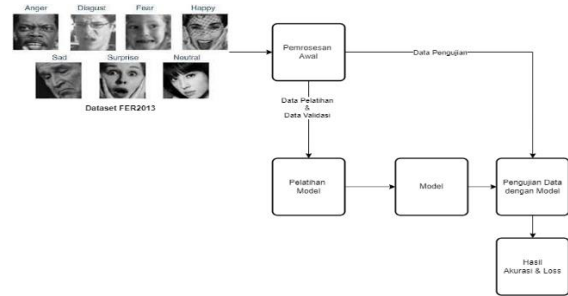
H. Arsitektur *Xception*

Xception adalah model yang dinamakan berasal dari model sebelumnya yaitu *Inception*, sedangkan *Xception* sendiri adalah berasal dari kata *Extreme Inception*. Arsitektur pada model berbasis pada konsep *Depthwise Separable Convolution*. *Depthwise Separable Convolution* ini memberikan kelebihan komputasi yang lebih hemat pada model [6].

III. METODE

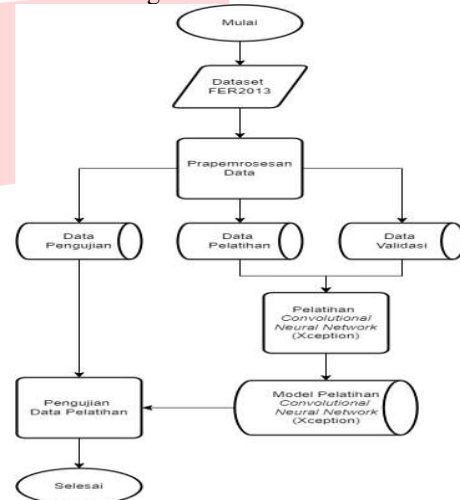
A. Gambaran Umum Sistem

Pada gambar 3.1, sistem pengenalan ekspresi pada wajah menggunakan model arsitektur yang berbasis *Convolutional Neural Network* (CNN) yaitu *Xception*. *Dataset* pada awalnya akan dibagi menjadi tiga menurut kegunaannya, yaitu ada data pelatihan, data validasi, dan data pengujian. Seluruh jenis data akan dilakukan pemrosesan awal terlebih dahulu. Kemudian untuk data pelatihan dan data validasi akan digunakan untuk proses pelatihan model *Xception*. Setelah model dilatih kemudian model di evaluasi dengan data uji, data validasi, dan data pelatihan kembali agar dapat dilihat seluruh akurasi dan loss dari data latih, data validasi, dan data uji.



GAMBAR 3.1 GAMBARAN UMUM SISTEM

B. Perancangan Sistem



GAMBAR 3.2 PERANCANGAN SISTEM

Pada Gambar 3.2, sistem dirancang menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur *Xception*. Data yang digunakan dalam sistem yaitu *Dataset FER2013* yang dibagi menjadi 3 jenis *dataset* menurut kegunaannya yaitu data pelatihan, data validasi, dan data uji. Sebelum data digunakan oleh sistem untuk pelatihan, data terlebih dahulu di *preprocessing*. Setelah itu sistem kemudian melakukan pelatihan model dengan data latih dan data validasi dan setelah model dilatih, model akan di evaluasi menggunakan data uji.

1. Data

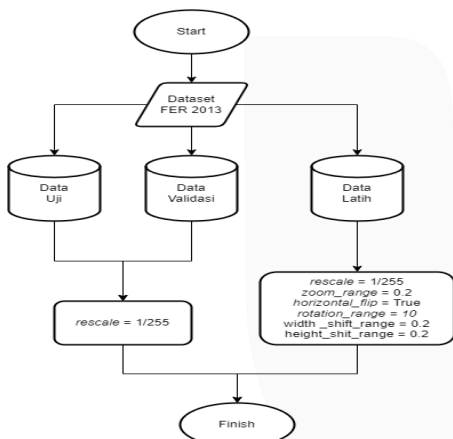
TABEL 3.1 JUMLAH KELAS EKSPRESI PADA DATASET FER 2013

Emosi	Train	Test	Validasi
Marah	3995	491	467
Jijik	436	55	56
Takut	4097	528	496

Senang	7215	879	895
Sedih	4830	594	653
Terkejut	3171	416	415
Netral	4965	626	607
Total	28709	3589	3589

Pada Tabel 3.1, peneliti menggunakan *dataset* FER2013 yang memiliki total data 35.887 data. Data ini dibagi menjadi tiga bagian menurut penggunaannya dengan perbandingan 80% data pelatihan dengan jumlah 28.709 data, 10% data validasi dengan jumlah 3.589 data, dan 10% data pengujian dengan jumlah 3.589 data. FER2013 memiliki tujuh kelas emosi dari range 0 - 6 yang masing-masing mewakili ekspresi wajah. (0=marah, 1=jijik, 2=takut, 3=senang, 4=sedih, 5=kaget, 6=netral).

2. Pemrosesan Awal

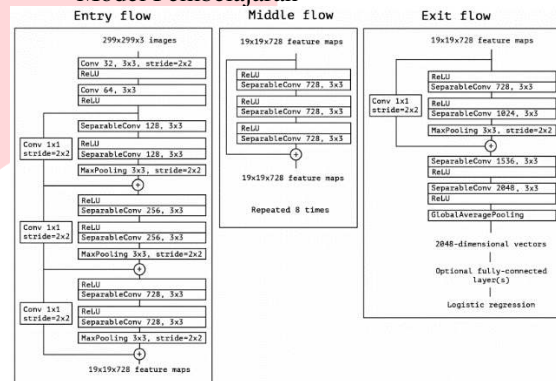


GAMBAR 3.3 PEMROSESAN AWAL

Pada Gambar 3.3, *dataset* FER2013 yang sudah di bagi menjadi menjadi tiga bagian menurut kegunaannya yaitu data latih, data validasi, dan data uji dimuat terlebih dahulu dan dilakukan pemrosesan awal untuk meningkatkan hasil agar lebih optimal ketika data di proses. Proses ini menggunakan pustaka *Tensorflow* dengan *Application Programming Interface (API) Keras* dengan modul *ImageDataGenerator*. Dalam penelitian ini, pemrosesan awal dijadikan sebagai parameter pengujian, yaitu pelatihan dengan *dataset* yang di Augmentasi dan pelatihan dengan *dataset* yang tidak di Augmentasi. Berikut adalah augmentasi yang digunakan pada *dataset* dan penjelasannya:

- a. *rescale* = 1/255, menormalisasi nilai piksel pada gambar dengan cara membagi tiap nilai piksel dengan nilai terbesar pada piksel.
- b. *zoom_range* = 0.2, memperbesar citra sebesar 20%.
- c. *horizontal_flip* = *True*, membalik gambar secara horizontal.
- d. *rotation_range* = 10, memutar gambar sebesar 10°.
- e. *width_shift_range* = 0.2, menggeser gambar secara horizontal sebesar 20%
- f. *height_shift_range* = 0.2, menggeser gambar secara vertikal sebesar 20%

3. Model Pembelajaran



GAMBAR 3.4 ARSITEKTUR XCEPTION

Pada Gambar 3.4, penyusun menggunakan algoritma *Deep Learning* yaitu CNN (*Convolutional Neural Network*) dengan model arsitektur *Xception*. Model dibangun terlebih dahulu dengan menyusun lapisan *neural network* yang digunakan pada model *Xception* menggunakan pustaka *Tensorflow* dengan *Application Programming Interface (API) Keras*. Terdapat dua jenis lapisan konvolusi pada arsitektur *Xception* yaitu *Convolution2D* dan *Separable Convolution2D*. Pada kedua lapisan konvolusi terdapat parameter-parameter seperti *filters*, *kernel size*, dan *strides*. Setelah model dibangun, model kemudian di *compile* dengan parameter-parameter yang telah ditentukan untuk pengujian seperti *epoch*, *optimizer*, *learning rate*, dan *batch size*.

Dalam tahap ini, model yang telah dibangun kemudian dilakukan proses pelatihan menggunakan data latih dan data validasi. Model dilatih dengan menggunakan *Model Fitting*, berikut potongan *code* untuk *Model Fitting* dapat dilihat pada lampiran 1. Pada *model.fit* ada beberapa parameter yang perlu ditentukan dalam proses training, berikut ini penjelasan parameternya:

1. *training_set* adalah inputan data latih yang sudah dilakukan prapemrosesan.
2. *validation_data* adalah input data validasi untuk mengevaluasi model setiap *epoch*.

3. *epoch* adalah jumlah pelatihan yang harus diterapkan pada set data.
4. *step_per_epoch* adalah jumlah total data latih dibagi dengan *batch_size*.
5. *validation_steps* adalah jumlah total data validasi dibagi dengan *batch_size*.

Model yang telah melalui proses pelatihan, kemudian akan di evaluasi menggunakan data latih, data validasi, dan data uji kembali untuk mendapatkan hasil *Accuracy* dan *Loss* dari model yang sudah dilatih. Kemudian dengan menggunakan *Confusion Matrix* akan di hitung nilai *Precision*, *Recall*, dan *Accuracy* pada model. *Precision* adalah menghitung persentase ketepatan data yang diprediksi positif yang pada aktualnya positif dengan membandingkan dengan seluruh data yang diprediksi positifnya, sedangkan *Recall* adalah menghitung persentase ketepatan data yang di prediksi positif dan pada aktualnya positif dengan membandingkan dengan seluruh data aktualnya yang di prediksi positif.

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian

Skenario pengujian pada model yang telah dirancang yaitu dengan cara mengkombinasikan nilai parameter-parameter yang digunakan dalam setiap pelatihan model. Parameter-parameter yang digunakan dalam pengujian model yaitu *epoch*, *optimizer*, *learning rate*, dan *batch size*, dan *data augmentation*. Untuk nilai parameter yang digunakan dalam pengujian, dapat dilihat pada tabel 4.1 berikut.

TABEL 4.1
PARAMETER PENGUJIAN

Parameter	Nilai Parameter
<i>Epoch</i>	60, 80, 100
<i>Optimizer</i>	Adam, <i>Stochastic Gradient Descent</i> (SGD)
<i>Learning Rate</i>	0.005, 0.0005
<i>Batch Size</i>	32, 128
Augmentasi Data	Dengan Augmentasi (<i>rescale</i> = 1/255, <i>zoom_range</i> = 0.2, <i>horizontal_flip</i> = True, <i>rotation_range</i> = 0.2, <i>width_shift_range</i> = 0.2, <i>height_shift_range</i> = 0.2), Tanpa Augmentasi (<i>rescale</i> = 1/255)

TABEL 4.2
SKENARIO PENGUJIAN DENGAN OPTIMIZER ADAM
DENGAN AUGMENTASI

<i>Epoch</i>	<i>Optimizer</i>	<i>Learning Rate</i>	<i>Batch Size</i>
60	Adam	0,005	32
			128
		0,0005	32
			128

80		0,005	32
			128
100		0,0005	32
			128
	0,005	32	
		128	

Pada tabel 4.2 berisi *scenario* pengujian dengan *dataset* yang di augmentasi dengan *rotation range* = 10, *width shift range* = 20%, *height shift range* = 20%, *horizontal flip* = true dengan optimizer Adam dan parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*.

TABEL 4.3
SKENARIO PENGUJIAN DENGAN OPTIMIZER SGD
DENGAN AUGMENTASI

<i>Epoch</i>	<i>Optimizer</i>	<i>Learning Rate</i>	<i>Batch Size</i>
60	SGD	0,005	32
			128
		0,0005	32
			128
80	SGD	0,005	32
			128
		0,0005	32
			128
100	SGD	0,005	32
			128
		0,0005	32
			128

Pada tabel 4.3 berisi *scenario* pengujian dengan *dataset* yang di augmentasi dengan *rotation range* = 10, *width shift range* = 20%, *height shift range* = 20%, *horizontal flip* = true dengan optimizer SGD dan parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*.

TABEL 4.4
SKENARIO PENGUJIAN DENGAN OPTIMIZER ADAM
TANPA AUGMENTASI

<i>Epoch</i>	<i>Optimizer</i>	<i>Learning Rate</i>	<i>Batch Size</i>
60	Adam	0,005	32
			128
		0,0005	32
			128
80	Adam	0,005	32
		0,0005	32

100		0,005	128
			32
			128
		0,0005	32
			128
			128

Pada tabel 4.4 berisi *scenario* pengujian dengan *dataset* tanpa di augmentasi hanya *rescale* = 1/255 dengan *optimizer* Adam dan parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*.

TABEL 4.5
SKENARIO PENGUJIAN DENGAN OPTIMIZER SGD
TANPA AUGMENTASI

Epoch	Optimizer	Learning Rate	Batch Size
60	SGD	0,005	32
			128
		0,0005	32
			128
80		0,005	32
			128
		0,0005	32
			128
100	0,005	32	
		128	
	0,0005	32	
		128	

Pada tabel 4.5 berisi *scenario* pengujian dengan *dataset* tanpa di augmentasi hanya *rescale* = 1/255 dengan *optimizer* SGD dan parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*.

B. Hasil Pengujian

Pada pengujian ini, parameter-parameter yang digunakan yaitu data yang diaugmentasi dan tanpa diaugmentasi, *epoch*, *learning rate*, *batch size*, dan *optimizer*. Pengujian dilakukan satu persatu dengan nilai parameter yang berbeda-beda.

TABEL 4.6
HASIL PENGUJIAN OPTIMIZER ADAM DENGAN
AUGMENTASI

Epoch	Optimizer	Learning Rate	Batch Size	Validation Accuracy	Training Accuracy	Test Accuracy
60	Adam	0.005	32	65,39%	72%	66,04%
			128	65,09%	71,68%	65,59%
		0.0005	32	64,75%	71,44%	65,65%

80		0.005	128	62,64%	69,3%	64,14%
			32	66,20%	76,31%	66,92%
		0.0005	128	65,51%	72,20%	66,12%
			32	65,45%	69,55%	66,23%
100		0.005	128	63,69%	69,55%	64,7%
			32	66,29%	77,52%	67,62%
		0.0005	128	65,56%	78,90%	66,59%
			32	66,45%	77,23%	66,68%
			128	65,73%	79,29%	67,04%

Pada tabel 4.6 dapat dilihat hasil pengujian menggunakan *dataset* yang di augmentasi dengan *optimizer* Adam dan parameter-parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*. Dari tabel tersebut dapat dilihat hasil evaluasi model yang terbaik diperoleh dengan parameter *epoch* = 80, *learning rate* = 0,005, dan *batch size* = 32 dengan hasil akurasi pelatihan 76,31%, akurasi validasi 66,2%, dan akurasi uji 66,93%.

TABEL 4.7
HASIL PENGUJIAN OPTIMIZER SGD DENGAN
AUGMENTASI

Epoch	Optimizer	Learning Rate	Batch Size	Validation Accuracy	Training Accuracy	Test Accuracy
60	SGD	0.005	32	60,71%	61,53%	61,52%
			128	54,64%	54,01%	55,59%
		0.0005	32	46,92%	45,62%	47,39%
			128	34,19%	30,62%	33,8%
80		0.005	32	61,33%	64,05%	61,74%
			128	56,7%	57,89%	57,79%
		0.0005	32	49,99%	49,28%	50,52%
			128	38,31%	34,72%	37,48%
100	0.005	32	63,81%	68,6%	63,81%	
		128	57,48%	58,31%	57,95%	
	0.0005	32	51,91%	50,62%	52,74%	
		128	36,97%	34,51%	36,75%	

Pada tabel 4.7 dapat dilihat hasil pengujian menggunakan *dataset* yang di augmentasi dengan *optimizer* SGD dan parameter-parameter lainnya seperti *epoch*, *learning rate*, dan *batch size*. Dari tabel tersebut dapat dilihat hasil evaluasi model yang terbaik diperoleh dengan parameter *epoch* = 100,

learning rate = 0,005, dan batch size = 32 dengan hasil akurasi pelatihan 68,6%, akurasi validasi 63,81%, dan akurasi uji 63,81%.

TABEL 4.8
HASIL PENGUJIAN OPTIMIZER ADAM TANPA AUGMENTASI

Epoch	Optimizer	Learning Rate	Batch Size	Validation Accuracy	Training Accuracy	Test Accuracy
60	Adam	0.005	32	58,4%	96,8%	58,23%
			128	56,73%	92,11%	58,15%
		0.0005	32	56,37%	95,29%	56,65%
			128	54,81%	97,08%	54,47%
80	Adam	0.005	32	60,99%	97,89%	60,18%
			128	58,54%	97,06%	59,21%
		0.0005	32	57,7%	98,43%	57,87%
			128	54,47%	94,98%	55,25%
100	Adam	0.005	32	59,6%	99,12%	60,57%
			128	60,27%	99,28%	60,99%
		0.0005	32	57,65%	98%	58,18%
			128	53,8%	98,44%	54,64%

Pada tabel 4.8 dapat dilihat hasil pengujian menggunakan dataset tanpa di augmentasi dengan optimizer Adam dan parameter-parameter lainnya seperti epoch, learning rate, dan batch size. Dari tabel tersebut dapat dilihat hasil evaluasi model yang kebanyakan mengalami overfitting dikarenakan hasil akurasi pelatihan lebih besar daripada akurasi validasi dan akurasi uji dengan selisih yang sangat jauh.

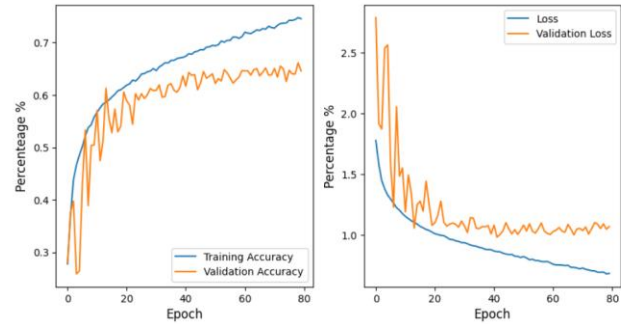
TABEL 4.9
HASIL PENGUJIAN OPTIMIZER SGD TANPA AUGMENTASI

Epoch	Optimizer	Learning Rate	Batch Size	Validation Accuracy	Training Accuracy	Test Accuracy
60	SGD	0.005	32	51,94%	99,78%	51,94%
			128	46,48%	99,82%	46,17%
		0.0005	32	44,08%	98,48%	44,11%
			128	39,93%	55,48%	39,98%
80	SGD	0.005	32	52,61%	99,65%	51,85%
			128	45,64%	99,78%	46,09%
		0.0005	32	44,8%	99,68%	43,38%

Epoch	Learning Rate	Batch Size	Training Accuracy		Test Accuracy	
			Value	%	Value	%
100	0.005	128	63,78%	45,52%	42,1%	
		32	52,88%	99,78%	53,05%	
	0.0005	128	46,64%	99,83%	45,25%	
		32	45,22%	99,63%	44,86%	
		128	41,6%	41,6%	40,62%	

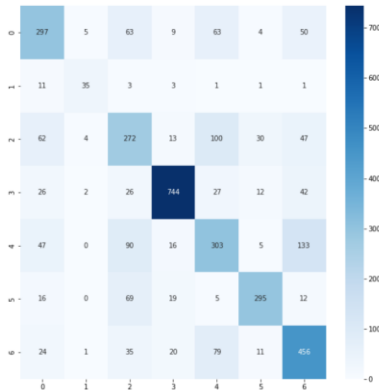
Pada tabel 4.9 dapat dilihat hasil pengujian menggunakan dataset tanpa di augmentasi dengan optimizer SGD dan parameter-parameter lainnya seperti epoch, learning rate, dan batch size. Dari tabel tersebut dapat dilihat hasil evaluasi model yang kebanyakan mengalami overfitting dikarenakan hasil akurasi pelatihan lebih besar daripada akurasi validasi dan akurasi uji dengan selisih yang sangat jauh.

Dari seluruh pengujian yang dilakukan peneliti, model terbaik yang di peroleh yaitu menggunakan dataset yang di augmentasi dengan parameter optimizer Adam, epoch = 80, learning rate = 0,005, dan batch size = 32. Hasil yang diperoleh yaitu akurasi pelatihan 76,31%, akurasi validasi 66,2%, dan akurasi uji 66,92%.



GAMBAR 4.1
GRAFIK AKURASI DAN LOSS DARI MODEL TERBAIK

Pada gambar 4.1 yaitu grafik akurasi dan loss dari model terbaik, bisa kita simpulkan bahwa model tidak mengalami overfitting karena garis grafik antara train_accuracy dan validation_accuracy tidak memiliki selisih yang terlalu jauh dan model juga tidak mengalami underfitting yang dimana untuk kasus underfitting ditandai dengan garis validation_accuracy lebih besar daripada garis train_accuracy.



GAMBAR 4.2
CONFUSION MATRIX DARI MODEL TERBAIK

Pada gambar 4.2 *confusion matrix* diatas dapat dilihat bahwa kelas tertinggi dimiliki oleh kelas nomor 3(senang), dimana memiliki nilai tertinggi dari data prediksi yang sesuai dengan data aktual. Hal ini bisa terjadi karena pada kelas senang memiliki data lebih banyak yang di prediksi oleh model. Sedangkan untuk kelas terendah dimiliki oleh kelas nomor 1(jijik) karena hanya sedikit data prediksi yang sesuai dengan data aktual. Hal ini bisa terjadi karena pada kelas jijik, data yang di prediksi hanya sedikit oleh model.

TABEL 4.10
TABEL PERHITUNGAN *PRECISION*, *RECALL*, DAN *ACCURACY*

Emotion Class	Precision	Recall
Angry	$\frac{297}{297+194} = 0,6049$	$\frac{297}{297+186} = 0,6149$
Disgust	$\frac{35}{35+20} = 0,6364$	$\frac{35}{35+12} = 0,7447$
Fear	$\frac{272}{272+256} = 0,5151$	$\frac{272}{272+286} = 0,4874$
Happy	$\frac{744}{744+135} = 0,8464$	$\frac{744}{744+80} = 0,9029$
Sad	$\frac{303}{303+291} = 0,5101$	$\frac{303}{303+275} = 0,5242$
Surprise	$\frac{295}{295+121} = 0,7091$	$\frac{295}{295+63} = 0,8240$
Neutral	$\frac{456}{456+170} = 0,7284$	$\frac{456}{456+285} = 0,6153$
Accuracy		$\frac{2402}{3589} = 0,6692 = 66,92\%$

Pada table 4.10 *Precision*, *Recall*, dan *Accuracy*, perhitungan akurasi dari data uji secara manual ini memiliki hasil 66,92% dan hasil *test_accuracy* yang didapat dari evaluasi model yang dibangun 66,92%. Hal ini membuktikan bahwa hasil perhitungan secara manual dan hasil yang didapat dari evaluasi model menggunakan data uji pada pengujian memiliki hasil yang mirip. Perhitungan pada tabel dapat menggunakan persamaan *Confusion Matrix* yang sudah di tuliskan pada BAB II.

TABEL 4.11
PERBANDINGAN AKURASI DENGAN PENELITIAN SEBELUMNYA

Penelitian	Model	Dataset	Akurasi Uji
Jaymon <i>et al</i>	<i>Xception</i>	FER2013	65.2%
Jaymon <i>et al</i>	<i>InceptionV3</i>		61.42%
Jaymon <i>et al</i>	<i>Simple CNN Model</i>		54%
Model yang diajukan	<i>Xception</i>		66,92%

Pada tabel diatas, dapat dilihat bahwa akurasi uji yang dimiliki penelitian sebelumnya dijadikan perbandingan dengan hasil akurasi uji yang didapatkan. Penelitian sebelumnya menggunakan tiga model yang berbeda dengan hasil akurasi uji yang berbeda. Namun salah satu model yang digunakan pada penelitian sebelumnya, sama dengan model yang digunakan pada penelitian ini yaitu model *Xception* [16].

Hasil akurasi uji yang dimiliki Jaymon *et al* yaitu untuk model *Xception* 65,2%, untuk model *InceptionV3* 61,42%, dan *Simple CNN Model* 54%. Sedangkan hasil akurasi uji yang diperoleh dari model yang diajukan yaitu 66,92%. Dari perbandingan akurasi uji tersebut, hasil yang terbaik merupakan model yang penyusun ajukan yaitu *Xception* dengan akurasi uji 66,92% dengan parameter sebagai berikut: *epoch* 80, *optimizer* Adam dengan *Learning Rate* 0.005, dan Augmentasi yang digunakan seperti yang sudah dijelaskan pada BAB III.

TABEL 4.12
DETEKSI EKSPRESI PADA CITRA MENGGUNAKAN MODEL TERBAIK

Jenis Ekspresi	Jumlah Prediksi Benar oleh Model Terbaik
Marah	4 dari 10 citra
Jijik	0 dari 10 citra
Takut	1 dari 10 citra
Senang	4 dari 10 citra
Sedih	1 dari 10 citra
Terkejut	1 dari 10 citra
Netral	7 dari 10 citra

Pada Tabel 4.12 adalah tabel prediksi menggunakan model terhadap 10 citra dari tiap-tiap ekspresi wajah. Dari 10 citra dengan ekspresi marah, model dapat memprediksi empat citra yang memiliki ekspresi marah, dari 10 citra ekspresi jijik model tidak dapat memprediksi ekspresi jijik satupun, dari 10 citra ekspresi takut model dapat memprediksi hanya satu citra yang memiliki ekspresi takut, dari 10 citra ekspresi senang model dapat memprediksi empat citra yang memiliki ekspresi senang, dari 10 citra ekspresi sedih model dapat memprediksi hanya satu citra yang memiliki ekspresi sedih, dari 10 citra dengan ekspresi terkejut model tidak dapat

memprediksi ekspresi terkejut satupun, dari 10 citra ekspresi netral model dapat memprediksi tujuh ekspresi dengan ekspresi netral.

Hasil yang diprediksi oleh model belum maksimal dikarenakan beberapa faktor yaitu model yang dicapai dalam pengujian belum mencapai akurasi yang maksimal. Hal ini terjadi karena dataset yang digunakan tidak memiliki jumlah data yang merata pada setiap kelas ekspresi, sehingga menyebabkan adanya ketimpangan dalam proses pembelajaran model dimana model lebih mudah mengenali ekspresi dengan data yang lebih banyak.

V. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Sistem yang dibangun dapat mengenali tujuh ekspresi dasar pada wajah.
2. Hasil Akurasi yang dicapai yaitu 76,31% untuk akurasi latih, 66,2% untuk akurasi validasi, dan 66,92% untuk akurasi uji.
3. Akurasi yang digunakan sebagai perbandingan dengan penelitian sebelumnya yaitu akurasi uji, dimana hasil penelitian sebelumnya oleh Jaymon *et al* mencapai akurasi uji sebesar 65.2% dengan model *Xception*, 61.42% dengan model *InceptionV3*, dan 54% dengan model *Simple CNN Model*. Sedangkan hasil akurasi uji yang didapatkan dari penelitian memiliki nilai yang lebih baik yaitu 66,92%.

REFERENSI

- [1] J. S. Wirtjes, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network (CNN)," *Repository Universitas Sumatera Utara*, pp. 6-11, 2019.
- [2] Y. Afriansyah, R. A. Nugrahaeni dan A. L. Prasasti, "Facial Expression Classification for User Experience Testing Using K-Nearest Neighbor," *IEEE*, 2021.
- [3] U. D. Rosiani, P. Choirina, S. Sumpeno dan M. H. P, "Menuju Pengenalan Ekspresi Mikro: Pendeteksian Komponen Wajah Menggunakan Discriminative Response Map Fitting," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, pp. 204-206, 2018.
- [4] R. Setiawan, "Mengenal Deep Learning Lebih Jelas," dicoding, 9 October 2021. [Online]. Available: <https://www.dicoding.com/blog/mengenal-deep-learning/>. [Diakses 28 June 2022].
- [5] D. L. Z. Astuti dan Samsuryadi, "Kajian Pengenalan Ekspresi Wajah menggunakan Metode PCA dan CNN," *Annual Research Seminar (ARS)*, p. 293, 2019.
- [6] N. Jaymon, S. Nagdeote, A. Yadav dan R. Rodrigues, "Real Time Emotion Detection Using Deep Learning," *International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies*, p. 6, 2021.
- [7] F. A. Isman, A. L. Prasasti dan R. A. Nugrahaeni, "Expression Classification For User Experience Testing Using Convolutional Neural Network," *IEEE*, 2021.
- [8] F. Mahmud dan D. M. A. Mamun, "Facial Expression Recognition System Using Extreme Learning Machine," *International Journal of Scientific & Engineering Research*, p. 26, 2017.
- [9] S. Ilahiyah dan A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *Justindo*, p. 52, 2018.
- [10] K. O'Shea dan R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv:1511.08458*, pp. 5-7, 2015.
- [11] N. Ibrahim, S. Sai'dah, B. Hidayat dan S. Darana, "Klasifikasi Grade Telur Ayam Negeri secara non-Invasive menggunakan Convolutional Neural Network," *Elkomika*, pp. 299-300, 2021 .
- [12] M. Zaki, A. L. Prasasti dan M. W. Paryasto, "Alexnet Architecture and Fuzzy Analysis on Talent Judge Decision Prediction Based on Facial Expression," *Jurnal Riset Informatika*, p. 343, 2022.
- [13] C.-F. Wang, "A Basic Introduction to Separable Convolutions," *Towards Data Science*, 14 August 2018. [Online]. Available: <https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728>. [Diakses 1 September 2022].
- [14] A. Gupta, "A Comprehensive Guide on Deep Learning Optimizers," *Analytics Vidhya*, 7 October 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/#h2_2. [Diakses 2022 September 2022].
- [15] J. Moolayil, *Learn Keras for Deep Neural Networks : A Fast-Track Approach to Modern Deep Learning with Python*, 2019.
- [16] Alzubaidi dan dkk, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big*

Data, pp. 17-38, 2021.

