

# EVALUASI OPTIMIZER ADAM DAN RMSPROP PADA ARSITEKTUR VGG-19 KLASIFIKASI EKSPRESI WAJAH MANUSIA

Fauzi Ramadhan\*<sup>1)</sup>, Julan Hernadi<sup>2)</sup>

1. Universitas Ahmad Dahlan, Yogyakarta, Indonesia
2. Universitas Ahmad Dahlan, Yogyakarta, Indonesia

## Article Info

**Kata Kunci:** *Optimizer Adam; Optimizer RMSProp; VGG-19; Ekspresi Wajah*

**Keywords:** *Optimizer Adam; Optimizer RMSProp; VGG-19; Facial Expressions*

## Article history:

Received 11 November 2024

Revised 15 Desember 2024

Accepted 14 Januari 2025

Available online 1 Maret 2025

## DOI :

<https://doi.org/10.29100/jupi.v10i2.6197>

\* Corresponding author.

Fauzi Ramadhan

E-mail address:

[fauzira1912@gmail.com](mailto:fauzira1912@gmail.com)

## ABSTRAK

Ekspresi wajah merupakan salah satu bentuk komunikasi non-verbal utama yang memungkinkan pengenalan dan pemahaman emosi manusia. Dalam konteks perkembangan Artificial Intelligence (AI), klasifikasi emosi melalui ekspresi wajah menjadi bidang penelitian yang penting. Salah satu model pembelajaran mesin populer yang digunakan untuk klasifikasi ekspresi wajah adalah model VGG-19 yang digunakan adalah versi modifikasi dengan penambahan lapisan fully connected dan dropout untuk mengurangi overfitting. Penelitian ini bertujuan untuk mengevaluasi performa dua optimizer populer yaitu Adam dan RMSProp pada model VGG-19 dengan membandingkan akurasi dan loss validasi dari kedua optimizer menggunakan dataset yang terdiri atas kumpulan citra ekspresi wajah dengan berbagai emosi. Hasil penelitian menunjukkan bahwa optimizer Adam mampu mencapai rasio akurasi tertinggi sebesar 91.26% dengan batch size 256 dan epoch 25, serta menghasilkan loss validasi sebesar 1.0241, sedangkan optimizer RMSProp menghasilkan rasio akurasi 84.60% dengan batch size 32 dan epoch 50, serta menghasilkan loss validasi 1.0910. Analisis lebih lanjut menunjukkan bahwa optimizer Adam tidak hanya lebih efisien dalam mencapai akurasi yang lebih tinggi, tetapi juga lebih stabil dalam konvergensi dibandingkan dengan RMSProp, meskipun memerlukan batch size yang lebih besar. Hal ini menunjukkan bahwa Adam lebih efektif dalam mengatasi variasi dalam gradien yang sering terjadi pada dataset besar dan kompleks. Data ini menunjukkan bahwa meskipun optimizer Adam membutuhkan batch size yang lebih besar, secara konsisten memberikan performa lebih baik dalam klasifikasi emosi dibandingkan dengan RMSProp.

## ABSTRACT

Facial expressions are a primary form of nonverbal communication that allow for the recognition and understanding of human emotions. In the context of the development of Artificial Intelligence (AI), emotion classification through facial expressions has become an important area of research. One popular machine learning model used for facial expression classification is the VGG-19 model, which in this study is a modified version with added fully connected layers and dropout to reduce overfitting. This research aims to evaluate the performance of two popular optimizers, Adam and RMSProp, on the VGG-19 model by comparing the accuracy and validation loss of both optimizer using a dataset consisting of images of facial expressions with various emotions. The result of the study showed that the Adam optimizer achieved the highest accuracy ratio of 91.26% with a batch size of 256 and 25 epoch, while producing a validation loss of 1.0241. In comparison, the RMSProp optimizer achieved an accuracy ratio of 84.60% with a batch size of 32 and 50 epoch, with a validation loss of 1.0910. Further analysis indicated that the Adam optimizer was not only more efficient in achieving higher accuracy but also more stable in convergence compared to RMSProp, despite requiring a larger batch size. This demonstrates that Adam is more effective in handling gradient variations, which often occur in large and complex datasets. These findings suggest that, although the Adam optimizer requires a larger batch size, it consistently delivers better performance in emotion classification compared to RMSProp.

## I. PENDAHULUAN

Ekspresi wajah manusia merupakan sumber informasi non-verbal yang kaya memungkinkan pengenalan emosi seperti kebahagiaan, kesedihan, dan lainnya [1]. Pengklasifikasian perasaan dari ekspresi wajah manusia telah menjadi fokus eksplorasi yang penting, terutama dalam bidang kecerdasan buatan. Hal ini disebabkan karena ekspresi wajah dapat menjadi petunjuk yang signifikan dalam memahami emosi seseorang. Awalnya, klasifikasi emosi dari ekspresi wajah manusia dilakukan secara manual, namun metode tersebut memiliki keterbatasan seperti membutuhkan waktu dan upaya yang besar serta rentan terhadap kesalahan. Salah satu tantangan utama dalam deteksi emosi adalah ekspresi wajah dapat dipengaruhi oleh faktor-faktor seperti aksesoris wajah, pencahayaan, dan variasi gaya. Pendekatan konvensional sering kali terbatas dalam mengoptimalkan ekstraksi fitur secara baik sehingga sulit dalam klasifikasi. Oleh karena itu dibutuhkan metode klasifikasi yang menggunakan teknik yang lebih canggih.

Seiring dengan kemajuan teknologi klasifikasi emosi dari ekspresi wajah kini dapat dilakukan melalui berbagai aplikasi, terutama dengan menggunakan teknologi kecerdasan buatan [2]. Salah satu teknik kecerdasan buatan yang banyak digunakan untuk tujuan ini adalah deep learning. Deep learning merupakan metode pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk memahami hubungan kompleks antara input dan output [3]. Deep learning telah terbukti berhasil dalam berbagai tugas klasifikasi, termasuk klasifikasi emosi dari ekspresi wajah manusia.

Berbagai arsitektur dalam deep learning telah banyak digunakan untuk keperluan klasifikasi, salah satunya adalah model VGG-19 yang merupakan salah satu arsitektur jaringan saraf konvolusional yang mendalam (deep convolution neural network) [4]. VGG-19 memiliki karakteristik unik seperti kedalaman jaringan yang terdiri dari 19 layer dengan 16 layer konvolusional dan 3 fully connected layer [5]. Selain itu model VGG-19 juga menggunakan kernel berukuran kecil ( $3 \times 3$ ) yang memungkinkan ekstraksi fitur secara lebih efektif dan mendetail [5]. Model ini telah digunakan dalam tugas pengenalan pola visual termasuk deteksi emosi [6]. Meskipun VGG-19 memiliki potensi yang besar dalam pengenalan pola visual, hasil akhirnya sangat bergantung pada metode optimasi (optimizer) untuk memaksimalkan kinerja model.

Dua optimizer populer yang banyak digunakan dalam proses training pada pembelajaran mesin yaitu metode Adam (adaptive moment estimation) dan metode RMSProp. Kedua optimizer ini memiliki kemiripan dalam mekanisme adaptasi learning rate secara dinamis selama proses training [7]. Adaptasi learning rate penting untuk memastikan model dapat belajar dengan efisien dari data yang kompleks dan bervariasi, seperti halnya data yang diperoleh dari ekspresi wajah manusia. Karakteristik yang membedakan kedua optimizer adalah Adam menggunakan bias correction pada momen pertama dan kedua, sementara RMSProp menggunakan decay rate untuk menjaga stabilitas training [8][9]. Walaupun metode Adam ini mengadopsi keunggulan metode RMSProp dan metode AdaGrad [10][11], namun masih perlu di uji cobakan pada arsitektur dan jenis data yang berbeda. Pada awal penemuannya, metode Adam sudah diterapkan pada regresi logistik untuk pengenalan pola dataset MNIST, multilayer neural network untuk dataset yang sama, convolution neural network (CNN) dengan beberapa layer konvolusi, pooling, dan unit taklinear pada data citra [10].

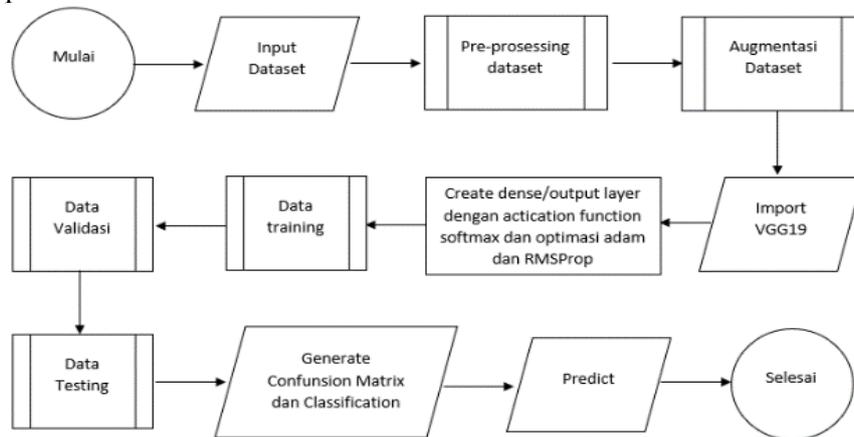
Penelitian sebelumnya menunjukkan keberhasilan penggunaan metode RMSProp pada arsitektur ResNet-152 dalam mendeteksi penyakit paru-paru pada manusia, dengan model terbaik mencapai accuracy 92%, precision 94%, recall 92%, f1-score 93%, dan specificity 96.75% [12]. Model ini menggunakan optimizer RMSProp dengan learning rate 0.001 dan batch size 8, serta menunjukkan nilai AUC tertinggi sebesar 0.946. Keberhasilannya menunjukkan bahwa optimizer RMSProp, dengan kemampuannya menyesuaikan learning rate secara adaptif, sangat efektif untuk dataset mini-batch. Meskipun banyak penelitian telah melakukannya, studi yang membandingkan kinerja optimizer berdasarkan kombinasi batch size dan epoch pada arsitektur VGG-19 masih terbatas. Oleh karena, penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja optimasi Adam dan RMSProp berdasarkan kombinasi hyperparameter pada model VGG-19 dalam mengklasifikasikan emosi dari ekspresi wajah manusia.

Penelitian ini akan menggunakan dataset yang sudah tervalidasi yaitu *FER2013* yang bersumber dari kaggle (<https://kaggle.com/datasets/deadskull7/fer2013>). Dalam penelitian ini dilakukan penyesuaian hyperparameter pada masing-masing optimizer dan melakukan evaluasi kinerja dari masing-masing optimizer menggunakan confusion matriks seperti akurasi, presisi, recall, dan F1-score. Dengan memahami kelebihan dan keterbatasan masing-masing metode optimasi, penelitian ini diharapkan dapat memberikan wawasan baru dalam pemilihan optimizer terbaik untuk tugas klasifikasi emosi wajah, serta memberikan solusi praktis pada berbagai masalah yang terkait dengan pengenalan wajah (*face-recognition*). Hasil penelitian ini diharapkan dapat meningkatkan akurasi dan efisiensi dalam klasifikasi emosi. Kecepatan dan ketepatan proses rekognisi citra wajah akan sangat banyak dibutuhkan pada berbagai bidang kehidupan bahkan saat ini sajah sudah banyak device yang

mengaplikasikan pengenalan wajah untuk proses masuk (log-in) dan kontrol pada berbagai portal, misalnya imigrasi, bandara, stasiun, dan lain-lain.

## II. METODE PENELITIAN

Metode penelitian dilakukan untuk mengetahui perbandingan antara optimizer Adam dan RMSProp dengan menggunakan arsitektur VGG-19 klasifikasi emosi ekspresi wajah manusia yang akan dibahas memiliki beberapa tahapan dapat dilihat pada Gambar 1.



Gambar. 1. Algoritma Penelitian

Algoritma tersebut melakukan input dataset, pra-prosesing dataset, augmentasi dataset, import VGG-19, membuat output/dense layer dengan fungsi aktivasi ReLU dan softmax, menggunakan optimizer Adam dan RMSProp, melakukan data training, melakukan data testing, dan evaluasi dengan menggunakan confusion matrix dan classification report.

### A. Pengumpulan Data

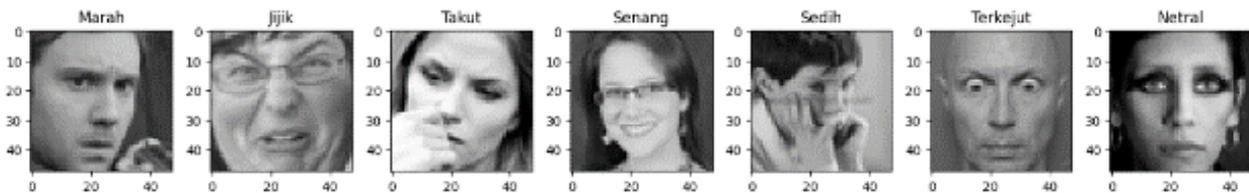
Dataset yang digunakan pada penelitian ini adalah dataset *FER2013* yang berisi citra wajah dalam skala *grayscale* dengan dimensi  $(48 \times 48)$  piksel. Dataset tersebut memiliki total 35.887 gambar ekspresi wajah yang di klasifikasikan ke dalam tujuh kelas emosional seperti marah, jijik, takut, senang, sedih, terkejut, dan netral. Dataset *FER2013* digunakan karena keberagaman dalam ekspresi wajah yang memungkinkan penelitian ini untuk melatih model deep learning pada berbagai variasi emosi yang relevan. Ukuran dataset yang besar juga memberikan keunggulan dalam hal stabilitas dan akurasi pelatihan, serta memungkinkan evaluasi model yang lebih kredibel. Relevansi *FER2013* sebagai dataset standar dalam penelitian klasifikasi emosi wajah memudahkan perbandingan hasil penelitian ini dengan studi-studi sebelumnya dan berkontribusi pada literatur yang ada. Rincian dapat dilihat pada Tabel 1.

TABEL I  
 DATASET FER2013

Kelas	Data Training	Data Test	Jumlah
Marah	3.993	960	4.953
Jijik	436	111	547
Takut	4.103	1.018	5.121
Senang	7.164	1.825	8.989
Sedih	4.938	1.139	6.077
Terkejut	3.205	797	4.002
Natural	4.982	1.216	6.198
<b>Total</b>	<b>28.821</b>	<b>7.066</b>	<b>35.887</b>

Berdasarkan Tabel 1, dataset pelatihan terdiri dari 28.821 data citra, sementara dataset uji terdiri dari 7.066 data citra. Jumlah total citra dalam setiap tahap adalah kombinasi dari semua kelas emosi yang ada dalam dataset tersebut. Pembagian dataset dengan proporsi 80% untuk training dan 20% untuk testing didasarkan pada prinsip bahwa model deep learning memerlukan data pelatihan yang cukup besar untuk dapat belajar dengan baik. Dengan memberikan 80% data untuk pelatihan, model memiliki kesempatan yang lebih besar untuk mengenali berbagai pola dan variasi dalam dataset, yang diharapkan dapat meningkatkan akurasi dan kemampuan generalisasi model.

Data testing yang berjumlah 20% digunakan untuk mengevaluasi performa model setelah pelatihan. Ukuran ini dianggap cukup representatif untuk memberikan gambaran tentang bagaimana model akan bekerja pada data yang belum pernah dilihat sebelumnya.



Gambar. 2. Ekspresi Wajah Manusia

### B. Preprocessing Data

Gambar dalam kumpulan dataset tidak bisa langsung digunakan, sehingga perlu dilakukan pembuatan data latihan yang lebih optimal untuk tahap selanjutnya. Data citra di diskretisasi ke dalam skala *grayscale* 0-255 yang berukuran  $48 \times 48$  pixels. Data ini dinormalisasi ke dalam interval  $[0, 1]$  dengan formula sebagai berikut.

$$r_{ij} = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}} = \frac{x_{ij} - 0}{255 - 0} = \frac{x_{ij}}{255}$$

Selanjutnya data dalam bentuk pixels direshape menjadi array satu dimensi dengan panjang 2304, masing-masing bernilai pada rentang 0 hingga 1, sehingga diperoleh array satu dimensi dengan panjang 2304 sebagai hasil dari reshape. Pada proses reshaping, setiap gambar dalam dataset *FER2013* direpresentasikan sebagai array satu dimensi yang terdiri dari 2304 nilai, dimana nilai setiap merepresentasikan pixels dalam gambar berukuran  $48 \times 48$  pixels. Proses normalisasi kemudian mengubah tipe data menjadi *float*. Normalisasi citra merupakan suatu metode yang mengubah nilai keseluruhan dari data menjadi kisaran antara 0 hingga 1, dengan tujuan untuk memastikan konsistensi dalam nilai data. Teknik ini bertujuan mempermudah pemrosesan data dan analisis lebih lanjut.

### C. Augmentasi Data

Augmentasi data adalah teknik khusus untuk membuat data training baru secara artificial dari data yang sudah ada, sehingga model dapat belajar dari variasi data yang lebih luas dan meningkatkan performa [13]. Pada penelitian ini menggunakan *rotation range* 10 derajat, *zoom range* 10%, *height shift range* 10%, dan *horizontal flip*. Teknik augmentasi tersebut digunakan karena beberapa alasan yang relevan dengan tujuan penelitian

#### 1) Rotation range

*Rotation range* 10 derajat digunakan untuk memperkenalkan variasi orientasi wajah yang mungkin terjadi dalam kondisi nyata. Ekspresi wajah dapat sedikit berubah orientasi tergantung pada posisi kepala, sehingga dengan memutar kepala dalam batas yang wajar (10 derajat), model dapat belajar untuk mengenali emosi meskipun wajah tidak sepenuhnya menghadap kamera.

#### 2) Zoom range

*Zoom range* 10% diterapkan untuk mensimulasikan variasi jarak antara wajah dan kamera. Dalam kondisi nyata, jarak dapat bervariasi, dan dengan memperbesar atau memperkecil gambar sebesar 10%.

#### 3) Height shift range

*Height shift range* 10% digunakan untuk menangani variasi posisi vertikal wajah dalam gambar. Pergerakan kepala ke atas atau ke bawah dapat mengubah posisi wajah dalam frame, dan dengan mensimulasikan pergeseran vertikal.

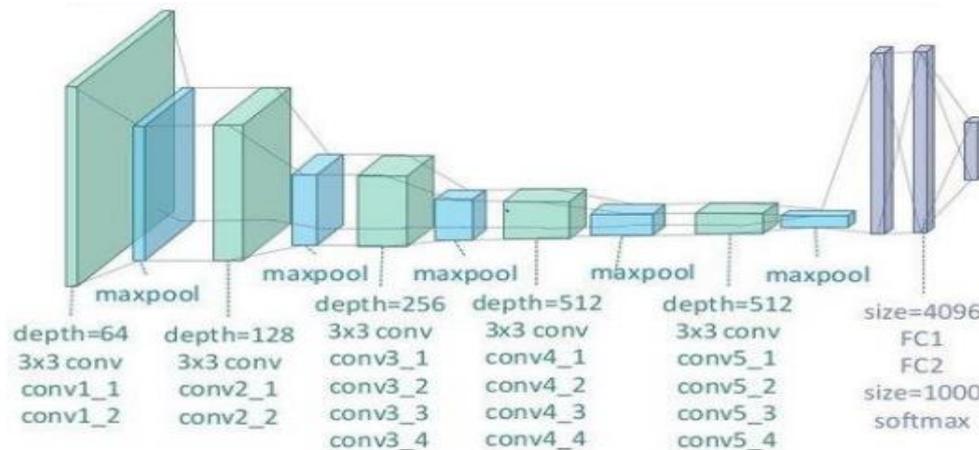
#### 4) Horizontal flip

*Horizontal flip* diterapkan untuk menangani asimetri dalam ekspresi wajah. Beberapa ekspresi mungkin terlihat sedikit berbeda jika wajah menghadap ke kiri atau ke kanan.

Penggunaan teknik-teknik augmentasi membantu dalam meningkatkan kinerja model dengan memperluas ruang variasi data training, sehingga model tidak hanya bergantung pada pola-pola spesifik yang ada dalam model dataset *FER2013*. Augmentasi data meningkatkan kemampuan generalisasi model, memungkinkan model untuk performa lebih baik pada data uji yang memiliki variasi serupa dengan data yang telah di-augmentasi.

### D. Arsitektur Model VGG-19

VGG-19 adalah salah satu model *convolutional neural network* yang terkenal dalam bidang pengolahan citra. VGG-19 memiliki 19 lapisan yang terdiri 16 lapisan konvolusional dan 3 lapisan *fully connected* dengan 4096 node pada 2 lapisan pertama dan 7 untuk lapisan terakhir [5].



Gambar. 3. Arsitektur Model VGG-19  
(<https://images.app.goo.gl/5mduN1ZwGeAhGXd49>)

VGG-19 menerima input berupa gambar *RGB* yang berukuran  $224 \times 224$  pixels. Pada Gambar 3 tersebut diproses oleh 16 *convolution layers* dengan ukuran *kernel*  $3 \times 3$  dan *stride* 1. *Convolution layers* menggunakan *padding* untuk menjaga resolusi spasial gambar. Setiap *convolution layers* diikuti oleh fungsi aktivasi ReLU untuk menambahkan non-linearitas. Jumlah filter pada *convolutin layers* meningkat secara bertahap dari 64, 128, 256, hingga 512. *Fully connected layers* juga menggunakan fungsi aktivasi ReLU, kecuali pada lapisan terakhir yang menggunakan aktivasi *softmax* untuk menghasilkan probabilitas kelas. Arsitektur VGG-19 digunakan dibandingkan dengan arsitektur lain karena beberapa alasan. Pertama, VGG-19 terkenal dengan kesederhanaan desainnya. Menggunakan hanya kernel  $3 \times 3$  dan *stride* 1, arsitektur tersebut menjaga resolusi spasial gambar melalui *padding*, memungkinkan ekstraksi fitur yang sangat baik dari data gambar. Hal ini memungkinkan model untuk menangkap berbagai fitur penting dari citra, termasuk tepi, tekstur, dan bentuk kompleks yang relevan untuk pengenalan pola visual. Kedua, peningkatan bertahap jumlah filter dari 64 hingga 512 membantu model untuk belajar representasi yang semakin abstrak dan kompleks seiring dengan kedalaman jaringan, yang penting untuk tugas pengenalan pola yang rumit seperti klasifikasi emosi wajah. Ketiga, arsitektur VGG-19 telah terbukti berhasil dalam berbagai tugas pengenalan pola visual, termasuk dalam kompetisi ImageNet, dimana model VGG-19 menunjukkan performa yang sangat baik dalam klasifikasi gambar.

Untuk memperoleh model dengan performa terbaik dilakukan penyesuaian parameter pada setiap model yang dilatih dan pengujian dengan variasi parameter yang berbeda. Ukuran *kernel* yang digunakan adalah  $3 \times 3$ , ukuran *pooling*  $2 \times 2$ , *padding* 'same', dan ukuran *stride* 1. Fungsi akitvasi yang diterapkan adalah *ReLU* untuk setiap *convolution layers*, sedangkan *softmax* digunakan untuk *output layer* karena bersifat kategorikal. Algoritma optimasi yang digunakan adalah *Adam* dan *RMSProp* dengan *learning rate* 0.0001. *Batch size* yang digunakan pada penelitian ini adalah 32, 64, 128, dan 256. Jumlah *Epoch* yang digunakan adalah 25 dan 50

Keunggulan dan keberhasilan VGG-19 dalam tugas pengenalan pola visual membuatnya sangat cocok untuk klasifikasi emosi wajah. Kemampuan VGG-19 dalam menangkap detail-detail halus dalam gambar dan mengkonsolidasi informasi tersebut ke dalam representasi yang bermakna memberikan dasar yang kuat bagi model untuk mengenali berbagai ekspresi wajah secara akurat. Selain itu, kesederhanaan dan efisiensi arsitektur membuatnya mudah diimplementasikan dan dioptimalkan, memberikan fleksibilitas dalam penyesuaian parameter untuk berbagai tugas klasifikasi visual.

Pada penelitian ini, model dilatih menggunakan notebook Kaggle yang menyediakan GPU, khususnya jenis NVIDIA Tesla P100. Framework deep learning yang digunakan adalah Tensorflow versi 2.15, yang dikenal dengan fleksibilitas dan performanya dalam pengembangan dan pelatihan model deep learning. Library yang digunakan dalam penelitian ini termasuk *pandas* (versi 2.1.4) untuk manipulasi data, *matplotlib* (versi 3.7.5) dan *seaborn* (versi 0.12.2) untuk visualisasi, *os* dan *cv2* untuk pengolahan file dan pemrosesan gambar, *sklearn* untuk evaluasi model, *tabulate* untuk menampilkan tabel hasil. Penggunaan perangkat keras dan perangkat lunak ini membantu memastikan bahwa eksperimen dapat direplikasi oleh penelitian lain, memberikan kontribusi bahwa berharga untuk komunitas ilmiah dalam bidang klasifikasi emosi wajah menggunakan deep learning.

#### E. Optimizer

Optimasi dalam konteks *neural network* adalah proses penyesuaian parameter model seperti bobot dan bias untuk meminimalkan *loss* selama training model [14]. Tujuan dari algoritma optimasi adalah untuk menemukan bobot optimal yang dapat meminimalkan *loss* dan memaksimalkan akurasi. Selama proses training, parameter (bobot)

model diubah untuk mencoba dan meminimalkan *loss function*. *Loss function* mengukur seberapa akurat model dalam memprediksi data uji [15]. Algoritma optimasi menyatukan *loss function* dan parameter model untuk memperbarui model. Dalam istilah yang lebih sederhana, algoritma optimasi membentuk model yang paling akurat dengan memanfaatkan bobotnya. Proses optimasi melibatkan algoritma seperti *gradient descent* yang umum digunakan untuk menghitung gradien dari *loss* terhadap parameter dan mengupdate parameter tersebut untuk mengurangi *loss*.

$$W_{t+1} = W_t - \eta \cdot \frac{\partial(\text{loss})}{\partial(\theta)}$$

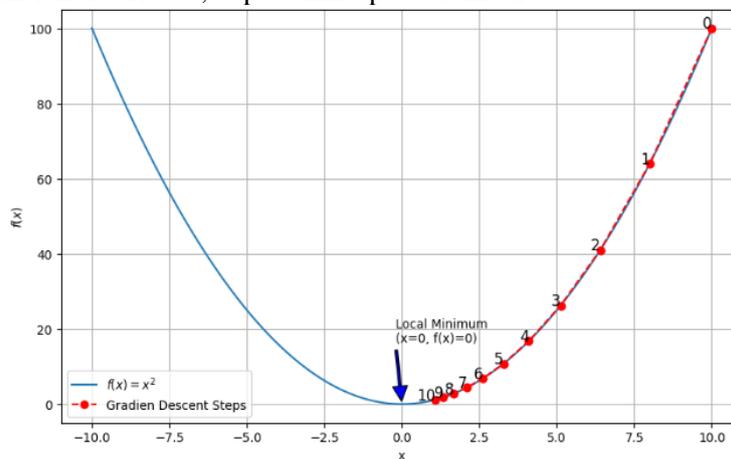
Dimana

$W$  : Vektor parameter model

$\eta$  : Learning rate

$\frac{\partial(\text{loss})}{\partial(\theta)}$  : Gradien dari *loss* terhadap parameter  $\theta$

Gradien menunjukkan arah tercepat untuk menurunkan nilai *loss* dengan mengurangi gradien dari parameter untuk bergerak menuju minimum lokal dari *loss*, dapat dilihat pada Gambar 4



Gambar. 4. Gradient Descent

Kekurangan pada gradient descent adalah melakukan perhitungan gradien untuk seluruh kumpulan data dalam satu pembaruan, maka perhitungan sangat lambat. Learning rate adalah ukuran langkah dalam penurunan gradien yang menentukan seberapa cepat atau lambat model bergerak menuju minimum lokal. Learning rate menunjukkan seberapa cepat atau lambat proses pembaruan bobot untuk mencapai bobot optimal. Salah satu algoritma optimasi yang populer pada jaringan saraf adalah *Adam* yang merupakan kombinasi dari *RMSProp* dan *Momentum*. *Momentum* menggabungkan gradien dari langkah sebelumnya ke dalam pembaruan, sehingga memberikan efek inersia yang membantu mengatasi rintangan seperti *noise* data atau titik pelana.

$$v_t = \beta v_{t-1} + (1 - \beta)g_t$$

Dimana

$v_t$  : Momentum

$\beta$  : Koefisien Momentum

$g_t$  : Gradien dari *loss* terhadap parameter

Momentum digunakan untuk memperbarui parameter model dalam jaringan saraf.

$$W_{t+1} = W_t - \eta \cdot v_t$$

Dimana

$W$  : Bobot

$\eta$  : Learning rate

$v_t$  : Momentum

### 1) Optimizer Adam

Adam optimizer adalah algoritma optimasi yang sering digunakan dalam *deep learning*. Adam optimizer memperbarui bobot dan learning rate dapat berubah – ubah selama proses training. Adam optimizer menggabungkan kelebihan dari *Momentum* dan *RMSProp optimizer* yang dapat memecahkan masalah gradien renggang untuk mencapai kinerja yang baik. Adam optimizer menggunakan rata – rata momen gradien pertama dan kedua untuk menyesuaikan parameter learning rate [11]. Berbeda dengan *RMSProp optimizer* yang hanya menggunakan momen pertama saja. Adam optimizer menggunakan dua variabel yaitu momen pertama ( $m$ ) dan momen kedua ( $v$ ). Momen pertama adalah rata – rata gradien, sedangkan momen kedua adalah rata -rata gradien kuadrat. Kedua momen tersebut diperbarui secara otomatis pada setiap waktu ( $t$ ). Langkah-langkah optimizer Adam sebagai berikut.

1. Inisialisasi :

$$\theta_0; m_0 = 0; v_0 = 0; \eta; \beta_1 = 0.9; \beta_2 = 0.99; \epsilon = 10^{-6}$$

2. Update Momen :

Hitung gradien

$$g_t = \nabla_0 L(\theta_t)$$

Update momen pertama (rata-rata eksponensial dari gradien)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) \cdot g_t$$

Update momen kedua (rata-rata eksponensial dari gradien kuadrat)

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) \cdot g_t^2$$

3. Bias Correction :

Karena  $m_t$  dan  $v_t$  dimulai dari 0, memiliki bias terhadap nilai awal.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

4. Update Parameter :

$$\Delta\theta = -\frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

$$\theta_{t+1} = \theta_t + \Delta\theta$$

Langkah – langkah tersebut memungkinkan algoritma Adam untuk mengoptimalkan parameter model secara adaptif dengan memperhitungkan estimasi momen gradien pertama dan kedua, serta menyesuaikan learning rate dengan menggunakan koreksi bias untuk momen pertama dan kedua.

### 2) Optimizer RMSProp

RMSProp adalah algoritma optimasi yang merupakan perbaikan dari metode Adagrad dengan menyesuaikan learning rate secara adaptif berdasarkan rata – rata pergerakan dari kuadrat gradien [16]. Tujuan utamanya adalah untuk mengatasi masalah perubahan learning rate yang tidak sama selama training. RMSProp menghitung rata–rata eksponensial dari kuadrat gradien dan menggunakan nilai tersebut untuk menormalisasi gradien. Proses tersebut mencegah terjadinya bergerak yang terlalu lambat atau terlalu jauh. Langkah-langkah optimizer *RMSProp* sebagai berikut.

1. Inisialisasi :

$$\theta_0; \eta; \beta_1 = 0.9; E[g_0^2] = 0; \epsilon = 10^{-6}$$

2. Hitung gradien :

$$g_t = \nabla_0 L(\theta_t)$$

3. Update rata – rate kuadrat gradien :

$$E[g_t^2] = \beta E[g_{t-1}^2] + (1 - \beta) \cdot g_t^2$$

$E[g_t^2]$  adalah rata – rata eksponensial dari kuadrat gradien pada iterasi  $t$

4. Perbarui Parameter :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g_t^2] + \epsilon}} g_t$$

Pembaruan parameter dihitung dengan mengurangi learning rate dari gradien dan membaginya dengan akar kuadrat dari momen kedua yang telah disesuaikan dengan  $\epsilon$ . Parameter model diperbarui dengan menambahkan perubahan parameter dengan parameter sebelumnya

#### F. Confusion Matrix

Confusion matrix adalah alat evaluasi kinerja esensial dalam *machine learning*, digunakan untuk menilai akurasi model klasifikasi. Confusion matrix terdiri dari empat element utama yaitu True Positif (TP), True Negatif (TN), False Positif (FP), dan False Negatif (FN) [17]. Dengan menggunakan confusion matrix dapat menganalisis performa model, mengidentifikasi kesalahan dalam klasifikasi dengan mengupayakan peningkatan akurasi prediksi. Untuk model klasifikasi dengan jumlah kelas target atau kategori sebanyak  $n$ , confusion matrix berukuran  $n \times n$  dipakai. Matrix tersebut membandingkan nilai target sesungguhnya dengan hasil prediksi model, memberikan gambaran komprehensif mengenai performa model klasifikasi tersebut.

TABEL II  
DATASET FER2013

	Predicted Positives	Predicted Negative
Actual Positives Instances	TP	FN
Actual Negatives Instances	FP	TN

#### 1) Accuracy

Mengukur seberapa sering model membuat prediksi yang benar dari keseluruhan prediksi yang dilakukan.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

#### 2) Precisions

Mengukur seberapa banyak dari prediksi positif yang benar-benar positif.

$$\frac{TP}{TP + FP}$$

#### 3) Recall

Mengukur seberapa baik model dapat menemukan semua sampel positif dalam dataset.

$$\frac{TP}{TP + FN}$$

#### 4) F1-Score

Rata-rata harmonis dari presisi dan recall, memberikan kinerja yang mempertimbangkan kedua metrik tersebut secara seimbang.

$$2 \left( \frac{Precision \cdot Recall}{Precision + Recall} \right)$$

### III. HASIL DAN PEMBAHASAN

#### A. Analisa Parameter

Pada penelitian ini hyperparamter yaitu batch size dan epoch di uji dengan tujuan untuk mendapatkan hyperparameter dengan performa terbaik pada model VGG-19 klasifikasi emosi ekspresi wajah. Hyperparameter di uji dalam proses training model untuk mengklasifikasikan ekspresi wajah. Nilai batch size dan epoch yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.

TABEL III  
 HYPERPARAMETER

Parameter	Ukuran
Batch Size	32, 64, 128, 256
Epoch	25, 50

Dengan memfokuskan pada kasus yang dikerjakan dengan melakukan percobaan. Dalam proses ini, peneliti menggunakan beberapa hyperparameter yang dapat diuji untuk membandingkan kinerja pengoptimalan Adam dan RMSProp. Hyperparameter yang akan di uji adalah kombinasi antara ukuran batch size optimal dan epoch optimal. Kombinasi yang terbaik dapat dilihat dari nilai loss validasi yang terendah dari seluruh kombinasi dan rasio antara akurasi training dan akurasi validasi yang mendekati satu.

Dalam konteks penelitian ini, pemilihan learning rate sebesar 0.0001 dianggap optimal untuk digunakan dengan algoritma Adam dan RMSProp. Learning rate ini cukup rendah untuk memastikan stabilitas dan kehandalan selama proses training mencegah divergensi, dan memungkinkan model dilatih secara konsisten. Adam dan RMSProp adalah algoritma optimasi adaptif yang menyesuaikan learning rate secara dinamis berdasarkan gradien. Dengan learning rate yang rendah seperti 0.0001, pembaruan bobot dapat dilakukan dengan stabil, mengurangi risiko instabilitas. Model tersebut kemudian di uji menggunakan data uji yang belum di ketahui oleh sistem. Proses penyetelan hyperparameter dijelaskan secara detail.

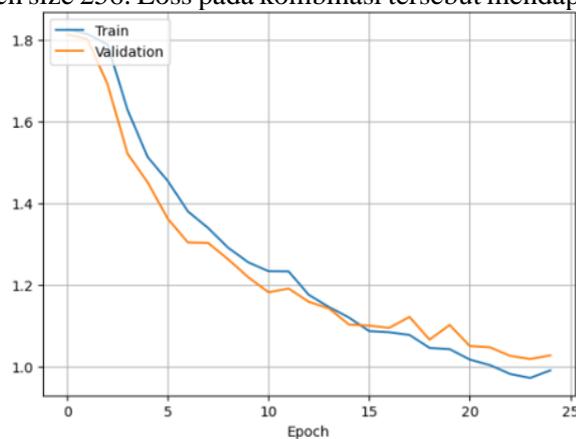
#### 1) Pengujian Pada Optimizer Adam

Pengujian optimizer Adam dilakukan berdasarkan kombinasi hyperparameter batch size dan epoch. Performa terbaik dicari berdasarkan kombinasi. Untuk memperoleh nilai tersebut, peneliti melakukan tes yang membandingkan kombinasi dari batch size dan epoch. Pengujian optimizer Adam untuk kombinasi hyperparameter yang terbaik dapat dilihat pada Tabel 4.

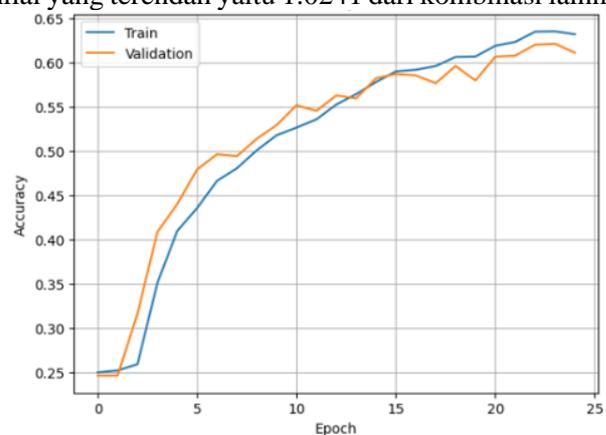
TABEL IV  
 HASIL TRAINING OPTIMIZER ADAM

Epoch	Batch Size	Akurasi	Validasi Akurasi	Loss
25	32	42.7482	40.9585	1.5041
	64	94.1447	67.1628	1.5548
	128	87.5300	65.5196	1.1752
	256	66.9999	61.1452	1.0241
50	32	46.5394	46.2106	1.3827
	64	86.7941	65.8326	1.2182
	128	86.5756	65.8958	1.0566
	256	82.3505	65.0320	1.0547

Berdasarkan Tabel 4 menunjukkan hasil kombinasi yang terbaik terdapat pada kombinasi antara epoch 25 dan batch size 256. Loss pada kombinasi tersebut mendapatkan nilai yang terendah yaitu 1.0241 dari kombinasi lainnya.



Gambar. 5. Loss Optimizer Adam



Gambar. 6. Akurasi Optimizer Adam

Berdasarkan Gambar 5 tersebut menunjukkan nilai loss. Hal tersebut menunjukkan bahwa model yang dilatih dengan kombinasi tersebut dapat mempelajari pola dalam data dengan baik. Rasio antara akurasi training dan akurasi validasi 91.26%, optimizer Adam mencapai konvergensi dalam 377.02 detik Efisiensi komputasi yang lebih

tinggi pada Adam disebabkan oleh penggunaan momen pertama dan kedua dari gradien untuk menyesuaikan learning rate secara dinamis. Ini memungkinkan proses konvergensi menjadi lebih cepat. Kombinasi hyperparameter yang optimal juga membantu dalam mengurangi jumlah iterasi per epoch, yang berkontribusi pada waktu pelatihan yang singkat. Hal tersebut menunjukkan bahwa kombinasi tersebut dapat menggeneralisasi dengan baik ke data yang belum di training oleh sistem.

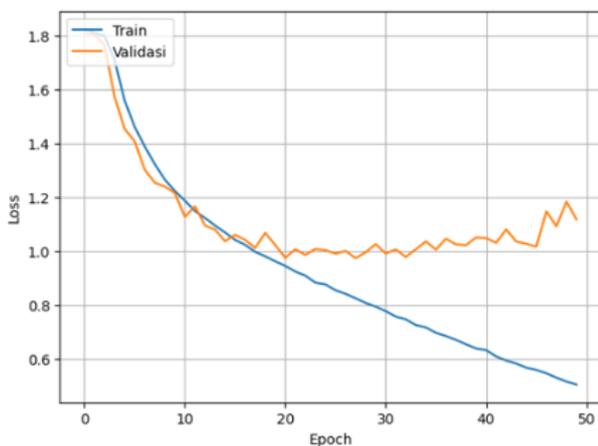
## 2) Pengujian Pada Optimizer RMSProp

Pengujian yang dilakukan dengan pengoptimalan RMSProp dilakukan dengan hyperparameter batch size dan epoch. Performa terbaik dicari berdasarkan kombinasi. Untuk memperoleh nilai tersebut, peneliti melakukan tes yang membandingkan kombinasi batch size dan epoch. Uji optimasi RMSProp untuk kombinasi hyperparameter terbaik ditunjukkan pada Tabel 5

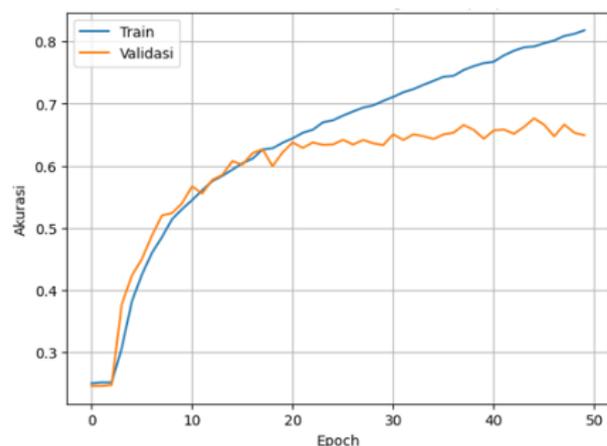
TABEL V  
HASIL TRAINING OPTIMIZER RMSPROP

Epoch	Batch Size	Akurasi	Validasi Akurasi	Loss
25	32	94.0785	65.7147	1.7071
	64	95.5833	66.6063	1.8600
	128	93.1346	66.2580	1.5125
	256	77.6621	63.8618	1.0500
50	32	76.0876	64.3633	1.0910
	64	84.8139	64.9565	1.1107
	128	97.9658	67.1914	1.3017
	256	92.8768	65.2549	1.4968

Berdasarkan Tabel 5 menunjukkan hasil kombinasi yang terbaik terdapat pada kombinasi antara epoch 50 dan batch size 32. Loss pada kombinasi tersebut mendapatkan yang terendah yaitu 1.0910 dari kombinasi lainnya.



Gambar. 7. Loss Optimizer RMSProp



Gambar. 8. Akurasi Optimizer RMSProp

Berdasarkan Gambar 7 tersebut menunjukkan nilai loss. Hal tersebut menunjukkan bahwa model yang dilatih dengan kombinasi tersebut dapat mempelajari pola dalam data dengan baik. Rasio antara akurasi training dan akurasi validasi mencapai 84.60%, optimizer RMSProp dengan kombinasi terbaiknya membutuhkan 857.64 detik untuk mencapai konvergensi. RMSProp membutuhkan lebih banyak waktu karena batch size yang lebih kecil dan jumlah epoch yang lebih besar, meningkatkan jumlah iterasi per epoch. Meskipun RMSProp menawarkan stabilitas yang lebih tinggi pada masalah dengan gradien yang tidak stabil, waktu konvergensi yang lebih lama menjadi kelemahan utama.

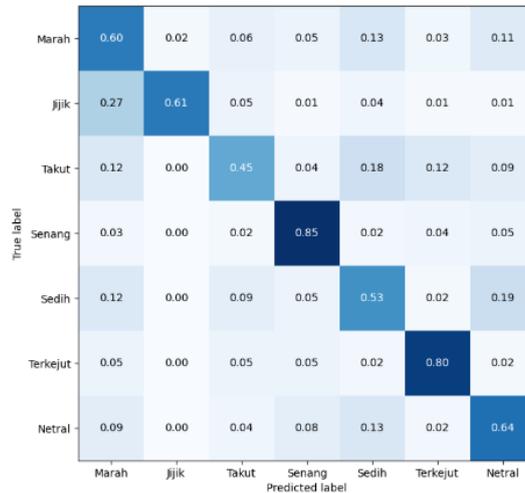
## B. Evaluasi

Di bawah ini adalah hasil evaluasi pengujian model terbaik VGG-19 menggunakan pengoptimalan Adam, dimana prediksi model dibandingkan dengan nilai sebenarnya dari data test. Dengan matrix *precision*, *recall*, *F1-score*, dan *support* untuk setiap kelas. Akurasi, Macro Avg, dan Weighted Avg dari keseluruhan performa model ditunjukkan pada Tabel 6.

TABEL VI  
HASIL TRAINING OPTIMIZER ADAM

Ekspresi	Precision	Recall	F1-Score	Support
Marah	0.54	0.60	0.57	985
Jijik	0.65	0.61	0.63	102
Takut	0.62	0.45	0.52	1043
Senang	0.83	0.85	0.84	1765
Sedih	0.54	0.53	0.54	1210

Terkejut	0.70	0.80	0.75	795
Netral	0.60	0.64	0.62	1278
Accuracy			0.66	7178
Macro Avg	0.64	0.64	0.64	7178
Weighted Avg	0.65	0.66	0.65	7178

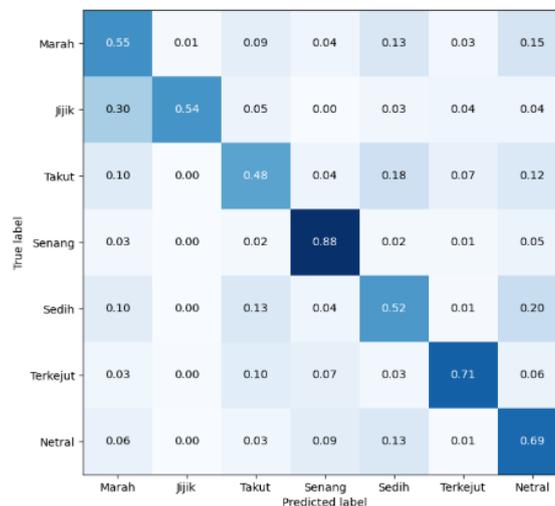


Gambar. 9. Normalized Confusion Matriks Optimizer Adam

Berikut hasil evaluasi pengujian model terbaik untuk estimasi VGG-19 menggunakan pengoptimalan RMSProp, membandingkan prediksi model dengan nilai sebenarnya dari data test. Dengan matrix *precision*, *recall*, *F1-score*, dan *support* untuk setiap kelas. Akurasi, Macro Avg, dan Weighted Avg dari keseluruhan performa model ditunjukkan pada Tabel 7.

TABEL VII  
 HASIL TRAINING OPTIMIZER RMSPROP

Ekspresi	Precision	Recall	F1-Score	Support
Marah	0.57	0.55	0.56	985
Jijik	0.75	0.54	0.63	102
Takut	0.56	0.48	0.52	1043
Senang	0.84	0.88	0.86	1765
Sedih	0.54	0.52	0.53	1210
Terkejut	0.80	0.71	0.75	795
Netral	0.57	0.69	0.63	1278
Accuracy			0.66	7178
Macro Avg	0.66	0.62	0.64	7178
Weighted Avg	0.66	0.66	0.66	7178



Gambar. 10. Normalized Confusion Matrik Optimizer RMSProp

Penelitian ini menggunakan model VGG-19 untuk klasifikasi emosi dengan pengoptimalan Adam dan RMSProp. Hasil menunjukkan bahwa kedua optimizer memiliki performa yang cukup baik, dengan akurasi keseluruhan sekitar 66%. Pengoptimalan Adam dengan epoch 25 dan batch size 256 menghasilkan performa terbaik, sementara pengoptimalan RMSProp dengan epoch 50 dan batch size 32 juga menunjukkan hasil yang baik. Pendekatan ini memiliki kelebihan seperti penggunaan model VGG-19 yang relatif dalam klasifikasi gambar, teknik augmentasi data untuk meningkatkan generalisasi model, dan optimizer adaptif yang membantu konvergensi lebih cepat dan stabil. Namun, kelemahannya termasuk risiko overfitting, akurasi validasi yang lebih rendah, dan kebutuhan sumber daya komputasi yang signifikan. Aspek yang dapat diperbaiki meliputi peningkatan dataset, fine-tuning pada model pretrained, penggunaan teknik regularisasi tambahan, model ensemble, dan eksplorasi optimizer lain.

Penelitian sebelumnya menggunakan arsitektur ResNet-152 dan dua optimizer Adam dan RMSProp untuk deteksi penyakit paru-paru, mencapai akurasi 92%. Kesamaan antara penelitian ini adalah penggunaan optimizer adaptif, sementara perbedaannya terletak pada arsitektur model, jenis dataset, dan hyperparameter yang digunakan. Faktor-faktor yang menyebabkan perbedaan hasil meliputi jenis dataset, arsitektur model, hyperparameter, dan teknik regularisasi. Penelitian ini memberikan kontribusi dengan menunjukkan performa VGG-19 dalam klasifikasi emosi wajah menggunakan Adam dan RMSProp, serta memberikan wawasan tentang pengaruh kombinasi hyperparameter pada performa model.

#### IV. KESIMPULAN

Penelitian ini membandingkan kinerja model VGG-19 dengan dua optimizer yaitu Adam dan RMSProp dalam kasus klasifikasi emosi ekspresi wajah. Hasil menunjukkan bahwa optimizer Adam menghasilkan performa terbaik dengan akurasi 91.26%, lebih unggul dibandingkan optimizer RMSProp dengan akurasi 84.60%. Model Adam juga menunjukkan performa yang lebih baik dalam metrik akurasi, akurasi validasi, loss validasi. Meskipun membutuhkan batch size yang lebih besar, optimizer Adam direkomendasikan untuk model VGG-19 dalam penelitian klasifikasi emosi ekspresi wajah karena menghasilkan akurasi yang lebih baik

#### DAFTAR PUSTAKA

- [1] D. Puspitasari and B. Putra Danaya, "Pentingnya Peranan Komunikasi Dalam Organisasi: Lisan, Non Verbal, Dan Tertulis (Literature Review Manajemen)," *J. Ekon. Manaj. Sist. Inf.*, vol. 3, no. 3, pp. 257–268, 2022, doi: 10.31933/jemsi.v3i3.817.
- [2] Y. Achmad, R. C. Wihandika, and C. Dewi, "Klasifikasi emosi berdasarkan ciri wajah menggunakan convolutional neural network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 11, pp. 10595–10604, 2019.
- [3] I. Arifin, R. F. Haidi, and M. Dzalhaqi, "Penerapan Computer Vision Menggunakan Metode Deep Learning pada Perspektif Generasi Ulul Albab," *J. Teknol. Terpadu*, vol. 7, no. 2, pp. 98–107, 2021, doi: 10.54914/jtt.v7i2.436.
- [4] S. Widodo, D. Setiawan, T. Ridwan, and R. Ambari, "Perancangan Deteksi Emosi Manusia berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16," *Syntax J. Inform.*, vol. 11, no. 01, pp. 01–12, 2022, doi: 10.35706/syji.v11i01.6594.
- [5] Weny Indah Kusumawati and Adisaputra Zidha Noorizki, "Perbandingan Performa Algoritma VGG16 Dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras," *J. Comput. Electron. Telecommun.*, vol. 4, no. 2, 2023, doi: 10.52435/complete.v4i2.387.
- [6] S. Cheng and G. Zhou, "Facial Expression Recognition Method Based on Improved VGG Convolutional Neural Network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 7, pp. 1–16, 2020, doi: 10.1142/S0218001420560030.
- [7] D. Learning, "Deep Learning - Goodfellow," *Nature*, vol. 26, no. 7553, p. 436, 2016.
- [8] Q. Tang, F. Shpilevskiy, and M. Lécuyer, "DP-AdamBC: Your DP-Adam Is Actually DP-SGD (Unless You Apply Bias Correction)," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 14, pp. 15276–15283, 2024, doi: 10.1609/aaai.v38i14.29451.
- [9] S. Vasudevan, "Mutual information based learning rate decay for stochastic gradient descent training of deep neural networks," *Entropy*, vol. 22, no. 5, 2020, doi: 10.3390/E22050560.
- [10] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [11] M. Reyad, A. M. Sarhan, and M. Arafah, "A modified Adam algorithm for deep neural network optimization," *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.
- [12] S. Asy Syifa and I. Amelia Dewi, "MIND (Multimedia Artificial Intelligent Networking Database Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru-Paru)," *J. MIND J. / ISSN*, vol. 7, no. 2, pp. 139–150, 2022, [Online]. Available: <https://doi.org/10.26760/mindjournal.v7i2.139-150>
- [13] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," 2017, [Online]. Available: <http://arxiv.org/abs/1712.04621>
- [14] A. Wibowo, P. W. Wiryanan, and N. I. Nuqoyati, "Optimization of neural network for cancer microRNA biomarkers classification," *J. Phys. Conf. Ser.*, vol. 1217, no. 1, 2019, doi: 10.1088/1742-6596/1217/1/012124.
- [15] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss Functions and Metrics in Deep Learning," pp. 1–53, 2023, [Online]. Available: <http://arxiv.org/abs/2307.02694>
- [16] I. H. Kartowisastro and J. Latupapua, "A Comparison of Adaptive Moment Estimation (Adam) and RMSProp Optimisation Techniques for Wildlife Animal Classification Using Convolutional Neural Networks," *Rev. d'Intelligence Artif.*, vol. 37, no. 4, pp. 1023–1030, 2023, doi: 10.18280/ria.370424.
- [17] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 13, pp. 1–26, 2020, doi: 10.1142/S0218001420520138.
- [18] L. Ciampiconi, A. Elwood, M. Leonardi, A. Mohamed, and A. Rozza, "A survey and taxonomy of loss functions in machine learning," vol. 1, no. 1, pp. 1–29, 2023, [Online]. Available: <http://arxiv.org/abs/2301.05579>
- [19] A. Mjihad, M. Saban, H. Azarmdel, and A. Rosado-Muñoz, "Efficient Extraction of Deep Image Features Using a Convolutional Neural Network (CNN) for Detecting Ventricular Fibrillation and Tachycardia," *J. Imaging*, vol. 9, no. 9, 2023, doi: 10.3390/jimaging9090190.
- [20] B. Putra, G. Pamungkas, B. Nugroho, and F. Anggraeny, "Deteksi dan Menghitung Manusia Menggunakan YOLO-CNN," *J. Inform. dan Sist. Inf.*, vol. 02, no. 1, pp. 67–76, 2021.

- [21] R. Yusuf and A. A. Huda, “Deteksi Emosi Wajah Menggunakan Metode Backpropagation,” *J. Autom. Comput. Inf. Syst.*, vol. 3, no. 2, pp. 103–114, 2023, doi: 10.47134/jacis.v3i2.60.
- [22] I. Gusmanda, J. Raharjo, and E. Suhartono, “Deteksi Penyakit Pneumonia Berbasis Citra XRay Menggunakan Cnn Arsitektur Vgg-19,” *e-Proceeding Eng.*, vol. 10, no. 6, pp. 5178–5181, 2023.
- [23] Y. Wang *et al.*, “Adapting Stepsizes by Momentumized Gradients Improves Optimization and Generalization,” no. M, pp. 1–40, 2021, [Online]. Available: <http://arxiv.org/abs/2106.11514>
- [24] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, “On Empirical Comparisons of Optimizers for Deep Learning,” no. 1, 2019, [Online]. Available: <http://arxiv.org/abs/1910.05446>
- [25] A. Chaudhary, K. S. Chouhan, J. Gajrani, and B. Sharma, *Deep learning with PyTorch*. 2020. doi: 10.4018/978-1-7998-3095-5.ch003.