

ANALISA PERBANDINGAN PERFORMA OPTIMIZER ADAM, SGD, DAN RMSPROP PADA MODEL H5

PERFORMANCE COMPARISON ANALYSIS OF OPTIMIZER ADAM, SGD, AND RMSPROP ON THE H5 MODEL

Doni Anggara¹⁾, Nana Suarna²⁾, Yudhistira Arie Wijaya³⁾

^{1), 2), 3)} Program Studi Teknik Informatika, STMIK IKMI Cirebon

Jl. Perjuangan No. 10B, Karyamulya, Kec. Kesambi, Kota Cirebon

Email : titanese.dot@gmail.com¹⁾, st_nana@gmail.com²⁾, yudhistira010471@gmail.com³⁾

Abstrak

Melakukan komunikasi tidak sebatas berbentuk verbal saja, bisa juga berkomunikasi nonverbal yaitu dengan menyampaikan informasi dari ekspresi wajah. Namun, permasalahan dalam analisa ekspresi wajah jika melakukan pendeteksian ekspresi wajah secara manual maka akan membutuhkan waktu yang cukup lama dan tidak selalu akurat, sedangkan jika melakukan pendeteksian menggunakan machine learning berbasis Python maka akan mempersingkat proses pendeteksian ekspresi wajah, oleh karena itu diperlukan suatu model yang memiliki tingkat accuracy yang mumpuni sehingga dapat mendeteksi dan mengklasifikasikan ekspresi wajah dengan cepat dan akurat. Tujuan utama dari penelitian ini yaitu untuk mengetahui optimizer mana yang terbaik diantara Adam, SGD, dan RMSprop untuk model klasifikasi dengan membandingkan performa hasil training dari setiap optimizer dimana hasil dari proses training menghasilkan file model dengan ekstensi h5. Model dengan metrik accuracy, validation accuracy, loss, waktu tempuh, dan size model terbaik di antara optimizer tersebut akan di nyatakan sebagai optimizer terbaik. Data yang digunakan berupa foto sebanyak 71.774 foto dengan 7 label ekspresi wajah yang diantaranya senang, sedih, terkejut, marah, takut, jijik, dan netral. Metode yang digunakan untuk mengukur performa model pada dataset yang diberikan yaitu evaluate() dari library Keras, classification_report dan precision_recall_fscore_support yang terdapat pada library sklearn.metrics. Dengan skenario pengujian 60 epochs dan learning rate sebesar 0.001, Optimizer Adam memiliki nilai accuracy lebih tinggi yaitu 68.61% disusul oleh SGD dengan nilai accuracy sebesar 57.68% dan accuracy RMSprop sebesar 54.83%.

Kata kunci: Adam, Deep learning, Ekspresi Wajah, Klasifikasi, Optimizer, RMSprop, SGD.

Abstract

Communicating is not limited to verbal communication alone; it can also be done through nonverbal means, such as conveying information through facial expressions. However, analyzing facial expressions manually can be time-consuming and not always accurate. On the other hand, utilizing machine learning based on Python can expedite the process of facial expression detection. Therefore, a model with high accuracy is needed to quickly and accurately detect and classify facial expressions. The main objective of this research is to determine the best optimizer among Adam, SGD, and RMSprop for a classification model by comparing the performance of each optimizer during training. The training process generates a model file with the extension h5, and the best optimizer is determined based on metrics such as accuracy, validation accuracy, loss, training time, and model size. The dataset used consists of 71,774 photos with 7 facial expression labels, including happiness, sadness, surprise, anger, fear, disgust, and neutral. The performance of the model on the given dataset is evaluated using the evaluate() function from the Keras library, as well as the classification_report and precision_recall_fscore_support functions from the sklearn.metrics library. With a testing scenario of 60 epochs and a learning rate of 0.001, Adam optimizer achieves the highest accuracy of 68.61%, followed by SGD with an accuracy of 57.68%, and RMSprop with an accuracy of 54.83%.

Keywords : Adam, Classification, Deep learning, Facial Expression, Optimizer, RMSprop, SGD.

1. PENDAHULUAN

Saat ini banyak teknik *machine learning* yang digunakan untuk menghasilkan performa yang lebih baik. Bahkan, teknik pembelajaran yang paling dalam atau disebut dengan teknik *deep learning* saat ini sudah mulai di implementasikan diberbagai sektor kehidupan [1]. Penelitian ini tentang *deep learning* yang berfokus pada analisa performa model h5 dengan 3 Optimizer yang berbeda menggunakan dataset ekspresi wajah. Proses *training* model dilakukan mengandalkan *Graphics Processing Unit* (GPU), karena GPU adalah komponen yang biasanya digunakan

untuk melakukan komputasi grafik atau gambar [2] yang mana itu berhubungan dengan penelitian ini yang menggunakan dataset berupa foto. Komputasi GPU memanfaatkan teknik paralel untuk mendapatkan kecepatan total yang lebih tinggi dibandingkan dengan komputasi yang dilakukan oleh *Central Processing Unit* (CPU) secara tradisional [2].

Tiga Algoritma yang bisa digunakan untuk klasifikasi citra diantaranya *Support Vector Machine* (SVM), *k-Nearest Neighbors* (k-NN), dan *Convolutional Neural Network* (CNN) [3]. Penelitian ini menggunakan metode CNN, karena metode tersebut banyak digunakan pada permasalahan yang berkaitan dengan *image classification*. CNN juga memiliki tingkat *accuracy* yang relatif tinggi dan memiliki hasil yang signifikan dalam pengenalan citra [4].

Penelitian terdahulu yang relevan yaitu “Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca” yang di lakukan oleh Mohammad Farid Naufal. Pada penelitiannya, di temukan bahwa Algoritma CNN memiliki *accuracy* yang tinggi dibandingkan dengan SVM dan KNN pada performa model nya dengan nilai parameter *Accuracy* 0.942, *Precision* 0.943, *Recall* 0.942, dan *F1-Score* 0.942 [5]. Beberapa *optimizer* yang terdapat pada CNN untuk mengklasifikasikan citra diantaranya Adam, SGD, RMSprop [6], [7]. Terdapat juga *optimizer* lainnya seperti Adagrad, Adadelata, Adamax dan Nadam [8]–[11].

Pada penelitian CNN sebelumnya yang menggunakan dataset FER-2013 hanya di uji pada ekspresi *happy* dan *sad* menunjukkan performa *Adamx*, *Adam*, *N-Adam*, dan SGD pada namun *optimizer RMSprop* tidak di ikut sertakan [12]. Maka dilakukan penelitian ini untuk mengetahui perbandingan performa model ekspresi wajah pada Algoritma CNN pada *Optimizer Adam*, SGD, dan *RMSprop* pada ekspresi senang, sedih, terkejut, marah, takut, jijik, dan netral. Alat bantu yang di gunakan dalam penelitian ini di antaranya, *CUDA Toolkit* dan *Nvidia CuDNN* untuk akselerasi GPU, *Confusion Matrix* untuk menilai performa model, *Jupyter Notebook* sebagai aplikasi utama untuk melakukan penelitian ini.

2. DASAR TEORI

2.1 Ekspresi Wajah

Pada ekspresi wajah tersirat sebuah ungkapan dengan berbagai emosi dan perilaku antarpribadi yang mencerminkan karakter tertentu pada seseorang. Ekspresi wajah merupakan aspek penting dalam komunikasi nonverbal pada komunikasi sosial, terutama bagi mereka yang memungkinkan mengekspresikan dan memahami emosi dan mental yang tak terucapkan. Ekspresi wajah di klasifikasikan menjadi beberapa bagian yaitu, ketakutan, jijik, kemarahan, kebahagiaan, kejutan, kesedihan dan netral [13].

2.2 Convolution Neural Network

Convolutional Neural Network (CNN) adalah salah satu algoritma *deep learning* pada pengolahan citra digital. Arsitektur CNN yang sering digunakan di antaranya yaitu, *AlexNet*, *GoogLeNet*, *Resnet*, *VGGnet* dan *MobileNet* [14]. *Convolution layer* yang terdapat pada CNN terdiri dari beberapa neuron yang tersusun, sehingga membentuk filter yang memiliki *pixel* lebar dan tinggi. Di dalam filter ini terdapat *strides*, *padding*, *ReLU*, dan *pooling layer* [15].

2.3 CUDA Toolkit

CUDA Toolkit menyediakan fasilitas untuk membuat aplikasi dapat berakselerasi dengan GPU yang memiliki kinerja tinggi. *CUDA Toolkit* ini mencakup *library* yang dipercepat oleh GPU, alat *debugging* dan alat pengoptimalan, *compiler C/C++*, dan *runtime library* [16]. *CUDA Toolkit* yang digunakan yaitu versi 11.2.0

2.4 Nvidia CuDNN

Peneliti *Deep learning* di seluruh dunia mengandalkan *cuDNN* untuk melakukan akselerasi terhadap GPU berperforma tinggi. Ini memungkinkan para peneliti untuk fokus pada pelatihan jaringan saraf (*Neural Networks*) dan mengembangkan aplikasi perangkat lunak daripada menghabiskan waktu untuk penyyetelan kinerja GPU tingkat rendah. *cuDNN* mempercepat kerangka kerja (*Frameworks*) *deep learning* yang banyak digunakan, diantaranya

Caffe2, *Chainer*, Keras, MATLAB, MxNet, *PaddlePaddle*, *PyTorch*, dan *TensorFlow* [17]. Nvidia CuDNN yang digunakan yaitu versi 8.1.0.

2.5 *Confusion Matrix*

Confusion matrix adalah tabel yang menjelaskan hasil klasifikasi dari jumlah data uji yang benar dan jumlah data uji yang salah [18]. Juga merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya, *Confusion Matrix* berisi informasi yang membandingkan hasil klasifikasi yang dibuat oleh sistem dengan hasil klasifikasi yang seharusnya [19].

2.6 *Jupyter Notebook*

Jupyter Notebook adalah lingkungan kerja (*Frameworks*) atau *Integrated Development Environment* (IDE) yang sudah dikemas di dalam Anaconda [20]. Anaconda sendiri merupakan platform distribusi *Python open-source* paling populer di dunia, mudah digunakan dan terpercaya. *Jupyter Notebook* tersedia untuk Windows, Linux, dan MacOS [21].

3. METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini bersifat eksperimental dengan pendekatan kuantitatif. Model yang di hasilkan adalah file dengan ekstensi h5, dimana model tersebut bisa di gunakan untuk deteksi ekspresi wajah.

3.2 Metode Penelitian

Di bawah ini merupakan tahapan yang dilakukan pada penelitian ini. Tahapan analisa data bisa dilihat pada Gambar 1.



Gambar 1. Metode Penelitian

1. Coding

Tahapan awal penelitian yaitu penyusunan *script* menggunakan bahasa pemrograman Python untuk mengimplementasikan *library* dan melakukan analisa data.

2. Data

Dataset yang digunakan pada penelitian ini adalah data sekunder *open-access* yang di *upload* oleh akun Jonathan Oheix 4 tahun lalu dan Aaditya Singhal 2 tahun lalu di *website* Kaggle yang bisa di akses dengan link sebagai berikut <https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset> dan <https://www.kaggle.com/datasets/aadityasinghal/facial-expression-dataset>. Tahapan ini adalah mempersiapkan data *training* dan validasi yang didalamnya terdapat tujuh label ekspresi wajah. Data dibagi menjadi 2, yaitu data *training* sebanyak 57.530 dan data validasi sebanyak 14.244. Ukuran citra masing-masing sebesar 48 x 48 *pixel* dengan format warna *grayscale*.

3. Analisa Data

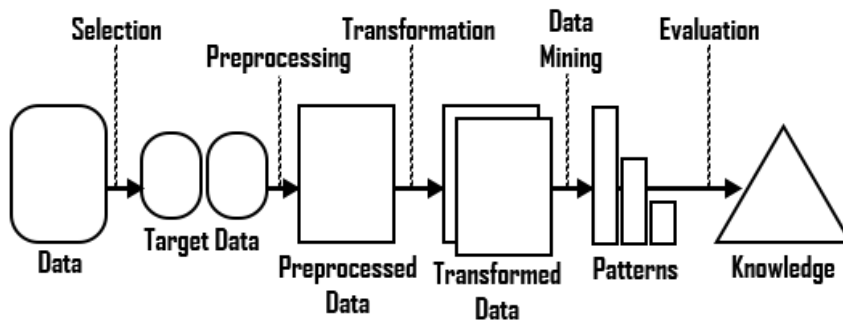
Setelah data dipersiapkan, data di analisa melalui tahapan *training* menggunakan arsitektur model yang telah di buat dan tahapan evaluasi. Tahapan ini di lakukan pada model yang menggunakan *optimizer* Adam, SGD, dan RMSprop.

3.3 Metode Analisa Data

Metode analisa data yang di gunakan yaitu *Knowledge Discovery In Database* (KDD) yang ilustrasinya dapat di lihat pada Gambar 2. Proses analisa data menggunakan Laptop Legion 5 dengan spesifikasi dilampirkan pada tabel 1.

Tabel 1. Spesifikasi Hardware yang digunakan

Spesifikasi	Keterangan
CPU	AMD Ryzen 7 5800H with Radeon Graphics 3.20 Ghz
RAM	16 GB
Penyimpanan	512 GB SSD
GPU	Nvidia RTX 3050 Ti, 4GB VRAM, TDP up to 95 Wat
Sistem Operasi	Windows 11 Home 64 Bit

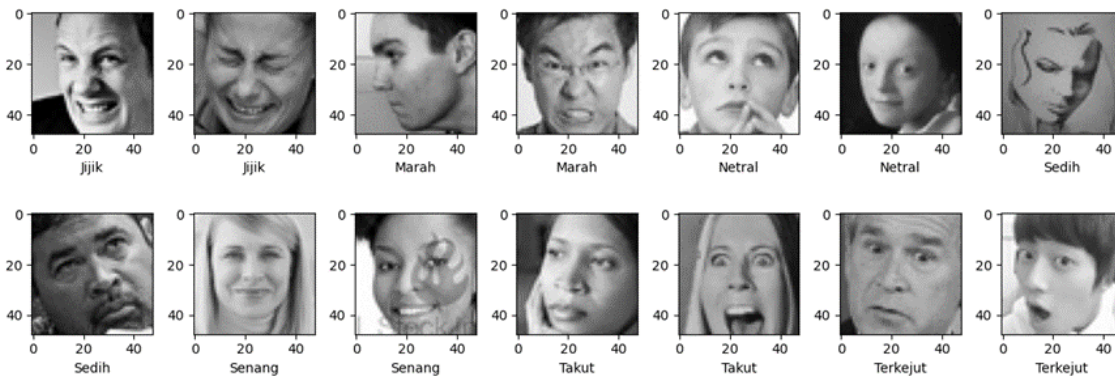


Gambar 2. Ilustrasi Proses Analisa Data (KDD)

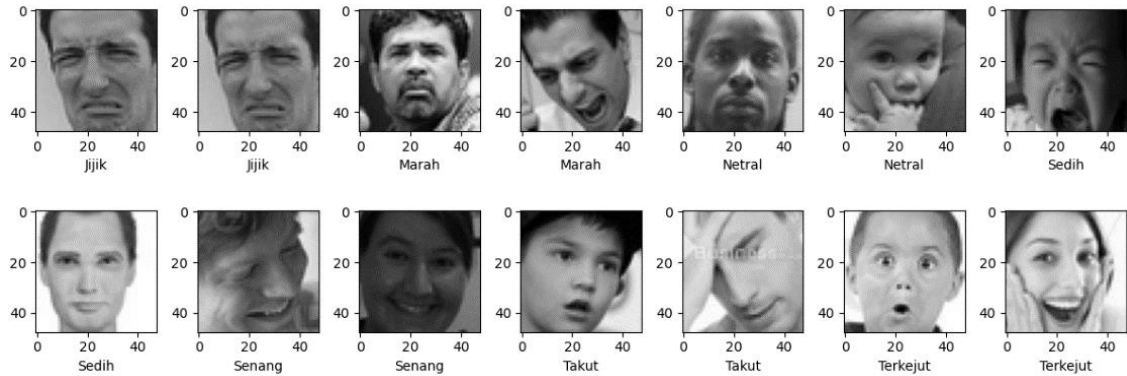
Fayyad dkk memperkenalkan proses *Knowledge Discovery in Database* (KDD) pada tahun 1996. Metode ini menggunakan teknik data mining untuk mengekstraksi pengetahuan berdasarkan ukuran dan ambang tertentu dalam basis data menggunakan proses *preprocessing*, *sampling*, atau transformasi data [22].

1. Selection

Tahapan ini memuat foto dari folder *training* dan *validation* yang di dalam nya terdapat folder dengan nama ekspresi wajah pada dataset yang akan di gunakan yaitu, senang, sedih, terkejut, marah, takut, jijik, dan netral.



Gambar 3. Sampel Dataset Training



Gambar 4. Sampel Dataset Validation

Tabel 2. Data Training Ekspresi Wajah

No.	Ekspresi	Jumlah	ID Ekspresi Wajah
1.	Jjikh	872	0
2.	Marah	7988	1
3.	Netral	9947	2
4.	Sedih	9768	3
5.	Senang	14379	4
6.	Takut	8200	5
7.	Terkejut	6376	6

Tabel 3. Data Validation Ekspresi Wajah

No.	Ekspresi	Jumlah	ID Ekspresi Wajah
1.	Jjikh	222	0
2.	Marah	1918	1
3.	Netral	2449	2
4.	Sedih	2386	3
5.	Senang	3599	4
6.	Takut	2042	5
7.	Terkejut	1628	6

2. Pre-processing

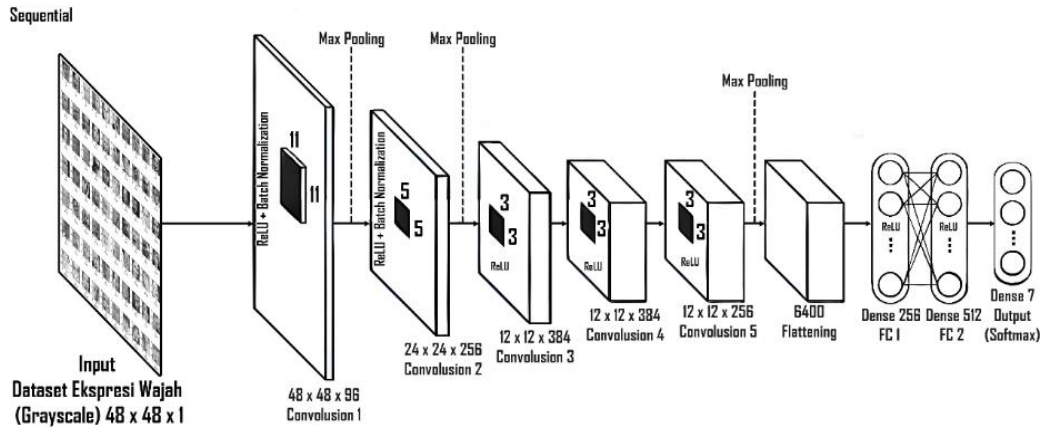
Pada tahap ini dilakukan Data *augmentation* pada setiap kelas, yaitu proses pembuatan data baru pada dataset yang sudah ada dengan menggunakan teknik transformasi seperti rotasi, pergeseran, *zoom*, dan *flipping* berdasarkan jumlah *batch* yang di gunakan. Tujuan dari data *augmentation* adalah untuk meningkatkan jumlah data yang tersedia sehingga dapat meningkatkan kinerja model *Deep learning* ketika data yang tersedia sangat sedikit dan tidak cukup untuk melatih model dengan baik. Data *augmentation* juga berguna untuk mengurangi *overfitting*. Dengan menambahkan variasi data, model dapat menyesuaikan dengan lebih baik dengan pola umum dalam data dan tidak hanya menyesuaikan dengan pola yang terlalu spesifik dalam data *training*.

Tabel 4. Data baru hasil Augmentation

Folder	Perhitungan (Jumlah data/jumlah batch)	Data baru
Training	57.530/128	± 450
Validation	14.244/128	± 111

3. Transformation

Tahap ini melakukan sebuah pemodelan arsitektur, yang menggunakan referensi dari arsitektur *AlexNet*, dikarenakan arsitektur CNN *AlexNet* berhasil menjadi pemenang pada ImageNet Competition untuk klasifikasi dan deteksi citra yang terdiri dari jutaan citra dengan puluhan ribu kelas [23], namun lapisan inputnya di sesuaikan agar sama dengan ukuran dataset yaitu 48 x 48 piksel. Ilustrasi arsitektur dapat dilihat pada Gambar 5.



Gambar 5. Ilustrasi Arsitektur

Tabel 5. Layer Arsitektur Model

Model Sequential			
No.	Layer (type)	Output Shape	Param #
1.	conv2d (Conv2D)	(None, 48, 48, 96)	11712
2.	activation (Activation)	(None, 48, 48, 96)	0
3.	batch_normalization (BatchNormalization)	(None, 48, 48, 96)	384
4.	max_pooling2d (MaxPooling2D)	(None, 23, 23, 96)	0
5.	conv2d_1 (Conv2D)	(None, 23, 23, 256)	614656
6.	activation_1 (Activation)	(None, 23, 23, 256)	0
7.	batch_normalization_1 (BatchNormalization)	(None, 23, 23, 256)	1024
8.	max_pooling2d_1 (MaxPooling2D)	(None, 11, 11, 256)	0
9.	conv2d_2 (Conv2D)	(None, 11, 11, 384)	885120
10.	activation_2 (Activation)	(None, 11, 11, 384)	0
11.	conv2d_3 (Conv2D)	(None, 11, 11, 384)	1327488
12.	activation_3 (Activation)	(None, 11, 11, 384)	0
13.	conv2d_4 (Conv2D)	(None, 11, 11, 256)	884992
14.	activation_4 (Activation)	(None, 11, 11, 256)	0
15.	max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 256)	0
16.	flatten (Flatten)	(None, 6400)	0
17.	dense (Dense)	(None, 256)	1638656
18.	activation_5 (Activation)	(None, 256)	0
19.	dropout (Dropout)	(None, 256)	0
20.	dense_1 (Dense)	(None, 512)	131584
21.	activation_6 (Activation)	(None, 512)	0
22.	dropout_1 (Dropout)	(None, 512)	0
23.	dense_2 (Dense)	(None, 7)	3591
Total Params		5.499.207	
Trainable Params		5.498.503	
Non-Trainable Params		704	

4. Data Mining

Tahap ini melakukan proses *training* dengan 60 *epochs* atau iterasi berdasarkan arsitektur model yang sebelumnya telah dibuat. Dimana *training* model Adam, SGD, dan *RMSprop* masing-masing di lakukan dengan tingkat *learning rate* sebesar 0.001.

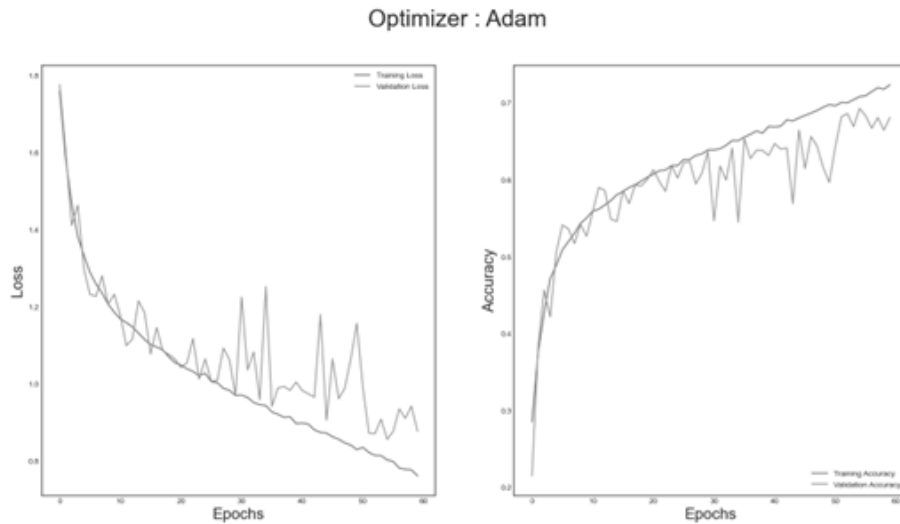
5. Evaluation

Di tahap *evaluation* ini, mengevaluasi performa model berdasarkan data hasil dari *training* pada proses *data mining* sebelumnya pada setiap *optimizer*, data di sajikan dalam bentuk *Plot* Grafik, Tabel performa *Confusion Matrix*, dan Tabel Evaluasi Performa. *Plot* Grafik menggambarkan proses pada saat *training* model, *Confusion Matrix* menampilkan performa

dalam memprediksi ekspresi wajah pada setiap kelas. Lalu, tabel evaluasi Performa untuk membandingkan performa dari masing-masing model.

4. PENGUJIAN DAN PEMBAHASAN

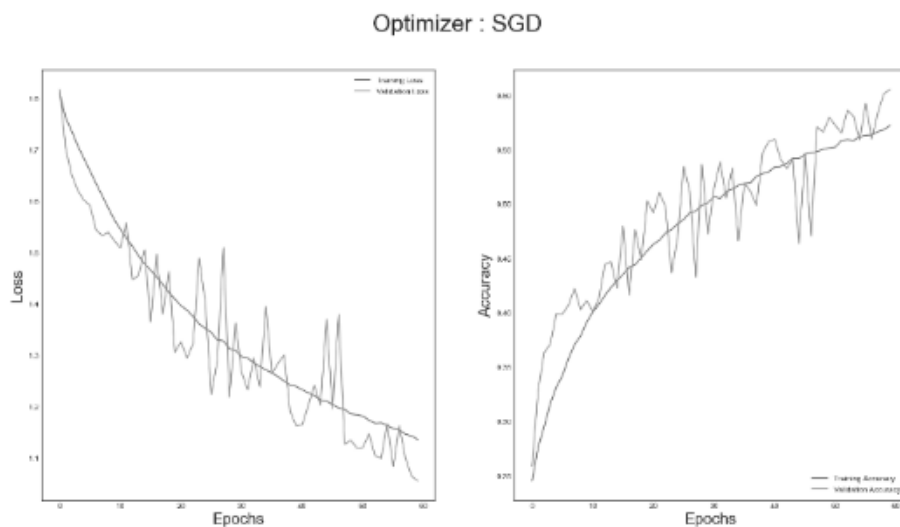
Berikut merupakan grafik history dari hasil proses pelatihan model dengan 60 epoch. Fokus dari grafik ini adalah metrik accuracy dan loss. Pada grafik loss mengukur tingkat kesalahan (loss). Pada grafik accuracy mengukur tingkat accuracy model pada data pelatihan dan data validasi.



Gambar 6. Plot Grafik – Adam

Terdapat peningkatan secara bertahap dalam accuracy dan validasi accuracy seiring bertambahnya epoch. Pada awal pelatihan, accuracy dan validasi accuracy cenderung rendah dan loss cukup tinggi. Namun, seiring berjalannya epoch, accuracy meningkat dan loss menurun, menunjukkan bahwa model semakin baik dalam mempelajari pola-pola dalam data pelatihan.

Pada akhir pelatihan, accuracy dan validation accuracy mencapai tingkat yang lebih tinggi, dan loss menurun ke angka yang lebih rendah. Dengan demikian, dapat disimpulkan bahwa model tersebut mengalami peningkatan performa selama proses pelatihan dengan meningkatnya accuracy dan penurunan loss.

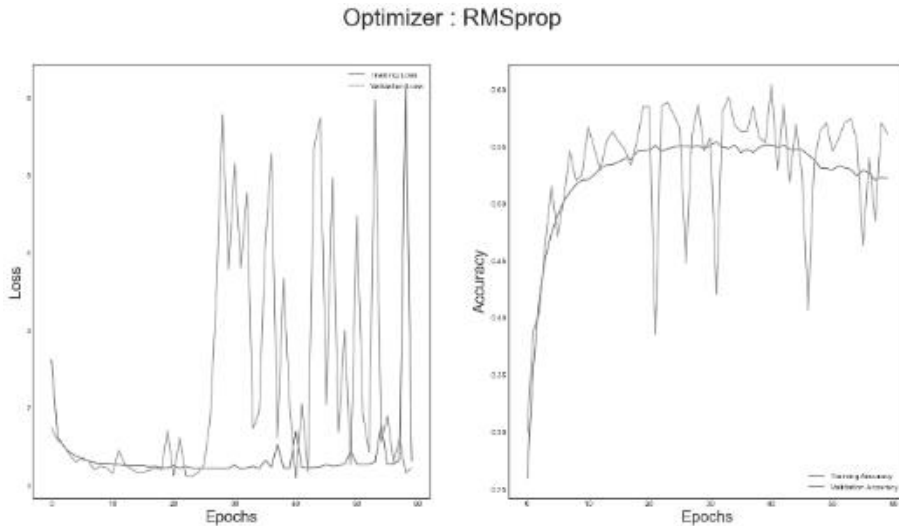


Gambar 7. Plot Grafik – SGD

Secara sekilas, grafik SGD terlihat mirip dengan Adam, bahwa accuracy dan validasi accuracy cenderung meningkat seiring dengan jumlah epoch yang bertambah.

Hal ini menunjukkan bahwa model secara bertahap mempelajari pola dalam data pelatihan dan dapat melakukan prediksi yang lebih baik pada data validasi.

Selain itu, *loss* dan *validation loss* juga cenderung menurun seiring dengan peningkatan *epoch*. Hal ini menunjukkan bahwa model semakin dekat dengan menemukan solusi yang optimal dan dapat menghasilkan prediksi yang lebih akurat.



Gambar 8. Plot Grafik - RMSprop

Pada *optimizer RMSprop*, terlihat banyak fluktuasi pada *loss* mulai dari *epoch* 25 sampai dengan *epoch* 60. Lalu pada bagian *Accuracy*, model secara bertahap mempelajari pola dalam data pelatihan namun terjadi fluktuasi yang signifikan pada *validation accuracy* mulai *epoch* 20 sampai *epoch* 60.

Setelah melakukan proses training, didapatkan data dari masing-masing *optimizer* yang di tampilkan dalam tabel *Confusion Matrix* yang digunakan untuk mengevaluasi kinerja model dalam melakukan klasifikasi dengan membandingkan prediksi model dengan nilai sebenarnya dari data uji atau validasi. Dimana terdapat metrik *precision*, *recall*, *F1-Score*, *support* terhadap masing-masing kelas. *Accuracy*, *Macro Avg*, dan *Weighted Avg* terhadap performa model secara keseluruhan.

Tabel 6. *Performa Model Optimizer Adam*

ID Ekspresi Wajah.	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>support</i>
0	0.80	0.50	0.61	222
1	0.62	0.59	0.60	1918
2	0.63	0.67	0.65	2449
3	0.53	0.60	0.56	2386
4	0.84	0.90	0.87	3599
5	0.66	0.37	0.47	2042
6	0.73	0.87	0.79	1628
Accuracy			0.68	14244
Macro avg	0.69	0.64	0.65	14244
Weighted avg	0.68	0.68	0.67	14244

Tabel 7. *Performa Model Optimizer SGD*

ID Ekspresi Wajah.	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>support</i>
0	1.00	0.03	0.06	222
1	0.53	0.51	0.52	1918
2	0.52	0.62	0.56	2449
3	0.48	0.51	0.50	2386
4	0.78	0.88	0.82	3599
5	0.51	0.24	0.32	2042
6	0.68	0.77	0.72	1628
Accuracy			0.61	14244
Macro avg	0.64	0.51	0.50	14244
Weighted avg	0.60	0.61	0.59	14244

Tabel 8. *Performa Model Optimizer RMSprop*

ID Ekspresi Wajah.	<i>precision</i>	<i>recall</i>	<i>F1-score</i>	<i>support</i>
0	0.00	0.00	0.00	222
1	0.43	0.56	0.49	1918
2	0.63	0.34	0.44	2449
3	0.38	0.65	0.48	2386
4	0.78	0.83	0.80	3599
5	0.44	0.18	0.26	2042
6	0.71	0.74	0.72	1628
Accuracy			0.56	14244
Macro avg	0.48	0.47	0.46	14244
Weighted avg	0.57	0.56	0.54	14244

Tahapan terakhir yaitu melakukan evaluasi model pada set data pelatihan dan set data pengujian.

```

1 train_loss, train_accu = model.evaluate(train_set)
2 test_loss, test_accu = model.evaluate(test_set)
3 print("Final train accuracy = {:.2f} , validation accuracy = {:.2f}".format(train_accu*100, test_accu*100))
4 print("Loss: ", scores[0])
    
```

Gambar 9. *Evaluasi Accuracy, Validation Accuracy, dan Loss*

Sehingga di dapatkan data *Accuracy*, *Validation Accuracy*, dan *Loss*. Ada juga metrik waktu tempuh selama proses *training* dan *size file* model H5.

Tabel 9. *Evaluasi Performa Model*

Keterangan	Adam	SGD	RMSprop
<i>Accuracy</i>	68.61	57.68	54.83
<i>Validation Acuracy</i>	68.20	60.61	56.22
<i>Loss</i>	0.87	1.05	1.22
<i>Waktu Tempuh</i>	0:54:83	0:56:22	0:58:54
<i>Size Model</i>	63.0 MB	21.0 MB	42.0 MB

Performa model ditunjukkan oleh Tabel 9, dimana terdapat beberapa metrik evaluasi yang dapat digunakan untuk memahami kinerja model:

1. *Accuracy*

Mencerminkan sejauh mana model dapat melakukan prediksi dengan benar. Model yang menggunakan optimizer Adam memiliki *accuracy* sebesar 68.61%, model dengan SGD memiliki *accuracy* 57.68%, dan model dengan RMSprop memiliki *accuracy* 54.83%. Model dengan *accuracy* yang lebih tinggi dianggap lebih baik dalam melakukan prediksi dengan benar.

2. *Validation Accuracy*

Mencerminkan sejauh mana model dapat melakukan prediksi dengan benar pada data yang belum pernah dilihat sebelumnya (data validasi). Model dengan optimizer Adam memiliki *accuracy* validasi sebesar 68.20%, model dengan SGD memiliki *accuracy* validasi 60.61%, dan model dengan RMSprop memiliki *accuracy* validasi 56.22%. Semakin mendekati *accuracy* dan *accuracy* validasi, semakin baik model dalam melakukan generalisasi pada data baru.

3. *Loss*

Menggambarkan sejauh mana model dapat mengurangi kesalahan prediksi selama proses training. Model dengan optimizer Adam memiliki loss sebesar 0.87, model dengan SGD memiliki loss 1.05, dan model dengan *RMSprop* memiliki loss 1.22. Semakin rendah nilai loss, semakin baik model dalam meminimalkan kesalahan prediksi.

4. Waktu Tempuh

Menunjukkan waktu yang diperlukan untuk melatih model dengan optimizer tersebut. Data berikut menunjukkan waktu dalam format jam:menit:detik. Waktu tempuh yang lebih singkat menunjukkan proses training yang lebih efisien.

5. *Size Model*

Menunjukkan ukuran file model setelah proses training selesai. Model dengan *optimizer* Adam memiliki ukuran 63 MB, model dengan SGD memiliki ukuran 21 MB, dan model dengan RMSprop memiliki ukuran 42 MB. Ukuran model yang lebih kecil dapat mengindikasikan efisiensi dalam penggunaan memori dan ruang penyimpanan.

5. KESIMPULAN

Setelah melakukan penelitian eksperimental, model yang menggunakan *optimizer* Adam memiliki *accuracy* sebesar 68.61%, model dengan SGD memiliki *accuracy* 57.68%, dan model dengan *RMSprop* memiliki *accuracy* 54.83%.

Dapat disimpulkan bahwa model yang menggunakan optimizer Adam memiliki kinerja yang lebih baik dengan metrik *accuracy*, *accuracy* validasi, loss, dan waktu tempuh yang lebih baik dibandingkan dengan model yang menggunakan SGD dan *RMSprop*. Namun untuk *size* model Adam lebih besar di banding *size* model SGD dan RMSprop.

DAFTAR PUSTAKA

- [1] T. Suprpti, D. A. Kurnia, D. Anggara, R. D. Rian, dan A. Setiawan, "Implementasi Model Algoritma Generative Adversarial Network (Gan) Pada Sistem Presensi Berbasis Deteksi Wajah (SIDEWA)," *Tematik : Jurnal Teknologi Informasi Komunikasi (e-Journal)*, vol. 9, no. 2, hlm. 231–236, Jan 2023, doi: 10.38204/tematik.v9i2.1048.
- [2] A. Abdul Razak, A. Muttaqin, dan M. Aswin, "Evaluasi Efisiensi Energi Komputasi FDTD Menggunakan Graphics Processing Unit," *Jurnal EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)*, vol. 13, no. 1, hlm. 1–5, Apr 2019, doi: 10.21776/jeccis.v13i1.557.

- [3] P. Ridho Aji, R. Basuki, dan A. Fetty Tri, "Implementasi Algoritma CNN Untuk Klasifikasi Citra Lahan Dan Perhitungan Luas," vol. 1, no. 1, 2020, doi: <https://doi.org/10.33005/jifosi.v1i1.5>.
- [4] I. Astuti, W. W. Ariestya, dan B. Solehudin, "Deteksi Objek Daun Semanggi Secara Real Time Menggunakan CNN-Single Shot Multibox Detector (SSD)," *Jurnal Ilmiah FIFO*, vol. 14, no. 1, hlm. 47–58, Mei 2022, doi: 10.22441/fifo.2022.v14i1.005.
- [5] M. F. Naufal, "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 8, no. 2, hlm. 311–318, Mar 2021, doi: 10.25126/jtiik.2021824553.
- [6] E. Rasywir, Y. Pratama, dan F. Fachruddin, "Eksperimen Pengujian Optimizer dan Fungsi Aktivasi Pada Code Clone Detection dengan Pemanfaatan Deep Neural Network (DNN)," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 2, hlm. 405–412, Sep 2022, doi: 10.47065/bits.v4i2.1776.
- [7] K. Wijaya dan E. P. Widiyanto, "Klasifikasi Kepemilikan Tanda Tangan Menggunakan Convolutional Neural Network dengan Arsitektur AlexNet," *MDP Student Conference*, vol. 2, no. 1, hlm. 133–143, Apr 2023, doi: 10.35957/mdp-sc.v2i1.4328.
- [8] D. Irfan, R. Rosnelly, M. Wahyuni, J. T. Samudra, dan A. Rangga, "PERBANDINGAN OPTIMASI SGD, ADADELTA, DAN ADAM DALAM KLASIFIKASI HYDRANGEA MENGGUNAKAN CNN," *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, vol. 5, no. 2, hlm. 244, Jun 2022, doi: 10.54314/jssr.v5i2.789.
- [9] D. D. Nur Cahyo, M. Anwar Fauzi, J. Tri Nugroho, dan K. Kusriani, "Analisis Perbandingan Optimizer pada Arsitektur NASNetMobile Convolutional Neural Network untuk Klasifikasi Ras Kucing," *J Teknol*, vol. 15, no. 2, hlm. 171–177, Jan 2023, doi: 10.34151/jurtek.v15i2.4025.
- [10] K. USMAN, N. K. C. PRATIWI, N. IBRAHIM, H. SYAHRAN, dan V. P. RAHADI, "Evaluasi Optimizer pada Residual Network untuk Klasifikasi Klon Teh Seri GMB Berbasis Citra Daun," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, no. 4, hlm. 841, Okt 2021, doi: 10.26760/elkomika.v9i4.841.
- [11] E. Rasywir, Y. Pratama, dan F. Fachruddin, "Eksperimen Pengujian Optimizer dan Fungsi Aktivasi Pada Code Clone Detection dengan Pemanfaatan Deep Neural Network (DNN)," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 2, hlm. 405–412, Sep 2022, doi: 10.47065/bits.v4i2.1776.
- [12] D. Alamsyah dan D. Pratama, "Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada FER-2013 Dataset," *Jurnal Teknologi Informasi*, vol. 4, no. 2, hlm. 350–355, Des 2020, doi: 10.36294/jurti.v4i2.1714.
- [13] H. Chaer, A. Rasyad, dan A. Sirulhaq, "Analisis Semiotika Ekspresi Wajah di dalam Ayat-ayat Agung al-Qur'an," *PALAPA*, vol. 7, no. 2, hlm. 234–250, Nov 2019, doi: 10.36088/palapa.v7i2.347.
- [14] F. Masykur, M. B. Setyawan, dan K. Winangun, "Epoch Optimization on Rice Leaf Image Classification Using Convolutional Neural Network (CNN) MobileNet," *CESS (Journal of Computer Engineering, System and Science)*, vol. 7, no. 2, hlm. 581–591, Jul 2022, doi: 10.24114/cess.v7i2.37336.
- [15] A. Hunaepi, M. Makhsun, dan S. Sarwani, "Deteksi Situs Pornografi Berdasarkan Gambar Menggunakan Metode Convolutional Neural Network," *J. Teknik Informatika*, vol. 12, no. 2, hlm. 137–148, 2019.
- [16] Nvidia Corporation, "CUDA," *Website*, 2023. <https://developer.nvidia.com/cuda-faq> (diakses 27 Januari 2023).
- [17] Nvidia Corporation, "Nvidia CuDNN," *Website*, 2023. <https://developer.nvidia.com/cudnn> (diakses 1 Februari 2023).
- [18] D. Normawati dan S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021. doi: <http://dx.doi.org/10.30645/j-sakti.v5i2.369>.
- [19] W. Nengsih, S. Informasi, C. Riau, J. Umban, S. No, dan R. Pekanbaru, "Analisa Akurasi Permodelan Supervised Dan Unsupervised Learning Menggunakan Data Mining,"

- SEBATIK*, vol. 23, no. 2, hlm. 285–291, Des 2019, Diakses: 18 Februari 2023. [Daring]. Tersedia pada: <https://jurnal.wicida.ac.id/index.php/sebatik/article/view/771>
- [20] Wenty Dwi Yuniarti, *Dasar-dasar pemrograman dengan python*. CV BUDI UTAMA, 2019. Diakses: 31 Januari 2023. [Daring]. Tersedia pada: https://books.google.co.id/books?hl=id&lr=&id=RZzODwAAQBAJ&oi=fnd&pg=PP1&dq=jupyter+notebook+adalah&ots=Mp3Bu7jngv&sig=GgadmOlq3wD_RMmM5ulIYA Y9nP0&redir_esc=y#v=onepage&q=jupyter%20notebook%20adalah&f=false
- [21] Anaconda Inc., “Anaconda 3,” 2023. <https://www.anaconda.com/products/distribution> (diakses 31 Januari 2023).
- [22] Nur Rina Utami, “Knowledge discovery in database metode generalized vector space model pada sistem temu kembali informasi terjemahan Kitab Mizanul Hikmah Berbahasa Indonesia,” Nov 2019. Diakses: 18 Februari 2023. [Daring]. Tersedia pada: <https://repository.uinjkt.ac.id/dspace/handle/123456789/50612>
- [23] W. Setiawan, “PERBANDINGAN ARSITEKTUR CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI FUNDUS,” *Jurnal Simantec*, vol. 7, no. 2, hlm. 48–53, Jun 2020, doi: 10.21107/simantec.v7i2.6551.