

KLASIFIKASI CITRA *DIABETIC RETINOPATHY* MENGGUNAKAN CNN (CONVOLUTIONAL NEURAL NETWORK) DENGAN ARSITEKTUR VGG-16

Alifidyah Nuril Hidayah, Firmansyah Adiputra, Meidya Koeshardianto

Sistem Informasi, Universitas Trunojoyo Madura

Jl. Raya Telang, Perumahan Telang Indah, Telang, Kec. Kamal, Kabupaten Bangkalan (69162), Indonesia
alifidyaha@gmail.com

ABSTRAK

Diabetic Retinopathy (DR) menjadi konsekuensi utama bagi penderita diabetes, namun dengan deteksi dini dan pengobatan yang tepat, risiko kehilangan penglihatan dapat dikurangi. Pemeriksaan *diabetic retinopathy* sendiri tidak dapat dilakukan dengan kasat mata karena berada di pembuluh darah bagian retina. *Diabetic retinopathy* dapat diidentifikasi melalui foto fundus oleh para ahli. Namun hal tersebut membutuhkan waktu yang cukup lama dan kurang akurat. Oleh sebab itu perlu adanya klasifikasi foto fundus secara sistematis agar dapat mendukung proses identifikasi. Salah satunya yakni melalui klasifikasi citra. Pada penelitian ini mengusulkan *Convolutional Neural Network* sebagai metode dalam mengklasifikasi citra *Diabetic Retinopathy* dengan menggunakan arsitektur VGG-16 dengan optimizer Adam. Di mana terdapat dua skenario uji coba. Uji coba pertama yakni proses klasifikasi tanpa menambahkan CLAHE dan augmentasi yang menghasilkan nilai akurasi training 75%. Sedangkan untuk hasil evaluasi menghasilkan akurasi sebesar 73%, presisi 67%, *recall* 73%, dan *f1-score* 69%. Pada skenario kedua yakni dengan menambahkan proses CLAHE dan augmentasi, yang menghasilkan akurasi training 78%. Sedangkan untuk hasil evaluasinya menghasilkan akurasi sebesar 76%, presisi 66%, *recall* 76%, dan *f1-score* 69%.

Kata kunci : *Diabetic Retinopathy, Convolutional Neural Network, VGG-16*

1. PENDAHULUAN

Menurut *World Health Organization* (WHO), Penderita diabetes adalah penyebab utama kebutaan, gagal ginjal, gagal jantung, stroke, dan amputasi anggota tubuh bagian bawah. Pada tahun 2019, penyakit ini menduduki peringkat keenam penyebab kematian di dunia. Sesuai dengan *International Diabetes Federation* (IDF), akan ada 783 juta penderita diabetes di seluruh dunia pada tahun 2045[1].

Diabetic Retinopathy (DR) merupakan komplikasi utama diabetes melitus yang mempengaruhi pembuluh darah retina dan dapat menyebabkan kehilangan penglihatan dan kebutaan[2].

DR merupakan masalah kesehatan masyarakat yang besar di seluruh dunia, terutama di negara-negara berpendapatan rendah dan menengah dimana akses terhadap pemeriksaan dan pengobatan terbatas. DR adalah penyebab utama kebutaan kelima dan gangguan penglihatan sedang dan berat pada orang dewasa berusia di atas 50 tahun [3].

Diabetic Retinopathy (DR) menjadi konsekuensi utama bagi penderita diabetes, namun dengan deteksi dini dan pengobatan yang tepat, risiko kehilangan penglihatan dapat dikurangi. Hal ini menekankan perlunya manajemen diabetes yang komprehensif dan pemeriksaan kesehatan mata secara teratur bagi penderita diabetes [1].

Pemeriksaan *diabetic retinopathy* sendiri tidak dapat dilakukan dengan kasat mata karena berada di pembuluh darah bagian retina. *Diabetic retinopathy* dapat diidentifikasi melalui foto fundus oleh para ahli. Namun hal tersebut membutuhkan waktu yang cukup

lama dan kurang akurat. Oleh sebab itu perlu adanya klasifikasi foto fundus secara sistematis agar dapat mendukung proses identifikasi. Salah satunya yakni melalui klasifikasi citra.

Adapun metode atau algoritma yang dapat digunakan untuk klasifikasi citra yakni CNN (*Convolutional Neural Network*). *Convolutional Neural Network* (CNN) merupakan variasi jaringan saraf tiruan yang terdiri dari bobot dan beberapa *hidden layer* yang disusun menjadi sebuah arsitektur[4].

Adapun kelebihan dari metode CNN dibanding metode lain yakni menghasilkan *computational cost* yang tinggi sehingga metode CNN lebih efisien dan relatif lebih cepat dibanding metode lain[5].

Kelebihan lain yang dimiliki metode ini yaitu kemampuan CNN dalam mengenali fitur yang abstrak dan beragam. Selain itu model ini mampu menangani deformasi gambar input seperti translasi, rotasi dan skala serta mengurangi sejumlah parameter bebas [6].

Metode CNN sendiri memiliki beberapa arsitektur seperti Dense-Net untuk mengklasifikasi *diabetic retinopathy* berdasarkan foto fundus dengan hasil rata-rata akurasi 79.96% [7].

Selanjutnya terdapat penelitian yang menggunakan arsitektur model EfficientNet dalam klasifikasi *diabetic retinopathy* berbasis pengolahan citra fundus dan *deep learning*, di mana yang menghasilkan nilai akurasi sebesar 82.096% [4].

Kemudian terdapat penelitian yang melakukan perbandingan arsitektur LeNet dan GoogleNet untuk mengklasifikasi *diabetic retinopathy*, di mana hasil uji coba menunjukkan nilai akurasi LeNet sebesar 95,5% dan GoogleNet sebesar 84,16% [8].

Selanjutnya terdapat penelitian yang menggunakan arsitektur VGG-16 yang mengklasifikasi *Diabetic Retinopathy* yang menghasilkan akurasi 96,24% [9]. Selanjutnya terdapat penelitian serupa yang menggunakan arsitektur VGG-16 untuk mengklasifikasi citra x-ray paru-paru yang menghasilkan akurasi sebesar 91,45%[10].

VGG-16 muncul pertamakali pada tahun 2014[11].

Di mana arsitektur ini memiliki 5 *convolutional layer* dan *pooling*, yang selanjutnya diakhiri oleh *fully connected layer*. Sedangkan parameter yang dimiliki arsitektur ini sebanyak 138 juta parameter [11].

Penelitian ini menggunakan metode *Convolutional Neural Network* dalam mengklasifikasi *Diabetic Retinopathy* dengan menggunakan arsitektur VGG-16. Di mana dalam penelitian sebelumnya telah dijelaskan bahwa VGG-16 cukup efektif dalam melakukan klasifikasi, segmentasi, maupun manipulasi pada citra.

2. TINJAUAN PUSTAKA

2.1. *Diabetic Retinopathy*

Diabetic Retinopathy (DR) adalah komplikasi mikrovaskuler jangka panjang dari diabetes pada mata. Tanpa deteksi dan pengobatan dini, DR akan berkembang dari kelainan ringan menjadi stadium lanjut. *Diabetic Retinopathy* dipersulit oleh *edema makula*, ablasi retina traksi, dan *glaukoma neovaskular* yang pada akhirnya menyebabkan gangguan penglihatan yang signifikan. Durasi diabetes yang lebih lama, kadar gula darah puasa yang tinggi, adanya riwayat hipertensi (darah tinggi), obesitas, hanya menjalani pengobatan insulin, adanya faktor keturunan dan status sosial ekonomi yang buruk merupakan faktor yang paling konsisten terkait dengan perkembangan DR pada pasien diabetes [2].

Ada dua jenis *Diabetic Retinopathy*:

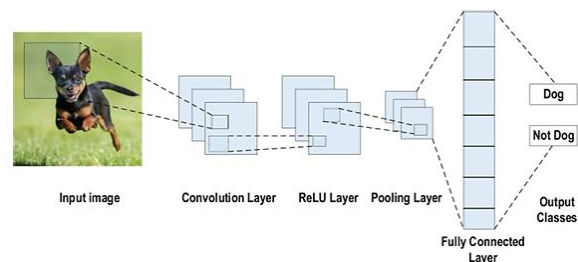
- Non-Proliferative Diabetic Retinopathy* (NPDR): Awalnya, penglihatan pada NPDR mungkin tidak terpengaruh, namun jika tidak ditangani, kondisi ini dapat berkembang ke tahap yang lebih parah. Berdasarkan tingkat keparahannya, NPDR dibagi lagi menjadi NPDR Sedang (ada bercak kapas dan perdarahan) dan NPDR berat (perdarahan intra retinal di empat kuadran mata - manik vena di dua kuadran atau kelainan mikrovaskuler intra retina di satu kuadran).
- Proliferative Diabetic Retinopathy* (PDR): Pada tahap lanjut ini, pembuluh darah baru tumbuh di permukaan retina, yang dapat menyebabkan perdarahan pada vitreous, zat seperti gel yang mengisi mata. PDR dapat menyebabkan kehilangan penglihatan atau kebutaan yang signifikan [1].

2.2. *Deep Learning*

Deep Learning berasal dari jaringan saraf konvensional tetapi kinerjanya jauh lebih unggul dari pendahulunya. Selain itu, *Deep Learning* menggunakan teknologi transformasi dan grafik secara bersamaan untuk membangun model pembelajaran *multi-layer*. Teknik *Deep Learning* yang paling baru dikembangkan telah memperoleh kinerja luar biasa yang baik di berbagai aplikasi, termasuk pemrosesan audio dan ucapan, pemrosesan data visual, pemrosesan bahasa alami, dan lain-lain [11].

2.3. *CNN (Convolutional Neural Network)*

CNN merupakan pendekatan yang paling sering digunakan untuk klasifikasi gambar, meningkatkan akurasi dengan meningkatkan kedalaman *hidden layer* dan secara efektif memperoleh lebih banyak parameter karakteristik. CNN sudah banyak berhasil dalam melakukan klasifikasi, bisanya mencakup klasifikasi wajah, objek, dan pelat nomor, namun klasifikasi gambar medis kurang umum karena kesulitan memperoleh gambar medis dan pemahaman yang buruk mengenai bagaimana penyakit muncul dalam berbagai gambar. Oleh karena itu, bantuan dokter biasanya penting untuk pengenalan citra medis guna mendiagnosis dan memberi label area fokus atau luka sebelum melanjutkan ke pelatihan model [12].



Gambar 1. Lapisan convolutional neural network [12]

Gambar di atas merupakan contoh arsitektur CNN, di mana pada *convolutional neural network* memiliki beberapa lapisan, yang akan dijelaskan sebagai berikut.

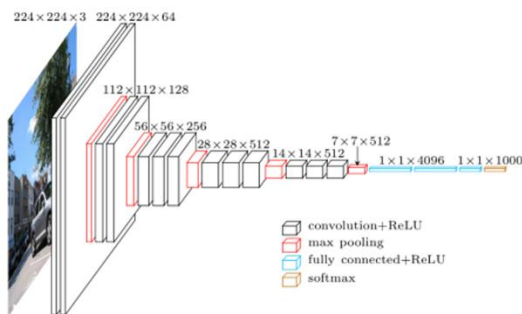
- Convolutional layer*, merupakan lapisan terpenting dalam arsitektur CNN. Di mana terdiri dari kumpulan filter konvolusional yang disebut kernel. *Input image* yang dinyatakan sebagai metrik N-Dimensional yang digabungkan dengan filter ini untuk menghasilkan *output feature map*.
- Pooling layer*, memiliki tugas utama yakni *sub-sampling* dari *feature map*. Di mana hal ini dihasilkan dengan mengikuti operasi *convolutional*. Dengan kata lain, pendekatan ini memperkecil *feature map*.
- Activation function*, berfungsi untuk memetakan input ke *output* dalam *neural network*. Ada beberapa *activation function* yang paling banyak digunakan dalam CNN, seperti Sigmoid, TanH, ReLU, Leaky ReLU, Noisy ReLU, dan Parametric ReLU.

- d. *Fully connected layer*, biasanya terletak di lapisan akhir setiap arsitektur CNN. Di dalam lapisan ini, setiap neuron terhubung ke semua neuron pada lapisan sebelumnya, yang disebut pendekatan *Fully Connected*. Ini berfungsi untuk pengklasifikasi CNN.
- e. *Loss function*, berfungsi untuk menghitung prediksi error yang dibuat di seluruh sampel training dalam CNN. Ada beberapa jenis *loss function*, seperti Cross-Entropy atau Softmax, Euclidean, dan Hinge [12].

2.4. VGG-16 (Visual Geometry Group)

VGG16 pertama kali muncul pada tahun 2014[9], model ini terdiri dari 5 layer konvolusi dan *pooling* lalu diakhiri dengan *fully connected layer* dan softmax. Model ini memiliki jumlah parameter yang banyak yaitu 138 juta parameter [10].

Adapun lapisan model VGG-16 dapat dilihat pada gambar berikut.



Gambar 2. Arsitektur model VGG-16 [10]

2.5. Augmentasi

Augmentasi citra merupakan pendekatan yang memungkinkan praktisi untuk memperluas variasi data yang tersedia dalam melatih model, tanpa perlu mengumpulkan data baru. Dengan menggunakan augmentasi, model akan menerima input citra yang telah mengalami transformasi, sehingga variasi citra input menjadi lebih beragam. Adapun tujuan dari augmentasi yakni selama proses pelatihan, model tidak melihat citra yang sama berulang kali. Augmentasi memungkinkan model untuk mengenali lebih banyak aspek dari data, sehingga menghasilkan model yang lebih mampu menggeneralisasi [13].

Augmentasi bertujuan melatih model pada data dalam jumlah besar untuk menghindari *over-fitting* [12].

2.6. Adam Optimizer

Adaptive moment estimation (Adam) diperkenalkan oleh Kingma dan Ba dan saat ini merupakan salah satu metode ukuran langkah paling populer di bidang jaringan saraf. Di mana mereka menunjukkan secara eksperimental, bahwa Adam *Optimizer* menyatu lebih cepat dengan jaringan saraf *multi-layer* atau jaringan saraf *convolutional*, daripada pengoptimal lainnya [14].

Metode ini dirancang dengan menggabungkan keunggulan dua metode yaitu: AdaGrad [15], yang bekerja dengan baik dengan gradien renggang, dan RMSProp [13], yang bekerja dengan baik dalam mode *online* dan *non-linear*. pengaturan stasioner; hubungan penting dengan metode ini dan metode optimasi stokastik lainnya. Beberapa keuntungan Adam yakni besaran pembaruan parameter tidak berubah terhadap penskalaan ulang gradien, *stepsize*-nya kira-kira dibatasi oleh *hyperparameter stepsize*, tidak memerlukan stasioner objektif, ia bekerja dengan gradien yang jarang, dan secara alami melakukan suatu bentuk *annealing* pada *stepsize* [14].

2.7. Evaluasi

Evaluasi merupakan tahap di mana kinerja model dinilai, yang menghasilkan nilai akurasi. Selain itu, juga dapat menampilkan hasil pengukuran metrik seperti *recall*, *accuracy*, *precision*, dan *F1-score*. Tahap ini juga berfungsi untuk mengidentifikasi kemungkinan bias maupun *overfitting*. Apabila hasil evaluasi belum memuaskan, maka langkah-langkah sebelumnya, seperti pemilihan algoritma atau pemrosesan data bisa diulang agar kinerja model meningkat [14].

2.8. Confusion Matrix

Confusion matrix merupakan tabel yang menunjukkan data uji yang diklasifikasikan dengan benar serta data uji yang diklasifikasikan secara salah. *Confusion matrix* juga digunakan untuk menilai kinerja model yang dihasilkan. Dalam *confusion matrix*, terdapat empat istilah, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [8].

2.9. Google Colaboratory

Google Colab adalah Jupyter Notebook gratis yang berjalan di cloud. Keuntungan utama yang dimiliki Google Colab dibandingkan *Editor Machine Learning* lainnya adalah tidak memerlukan setup apa pun dan notebook yang kita buat dapat diedit secara bersamaan oleh anggota tim – sama seperti kita mengedit dokumen di Google Dokumen. Google Colab mendukung populer *library machine learning* yang dapat dimuat di *notebook*.

Adapun fasilitas yang dapat digunakan oleh pengguna seperti:

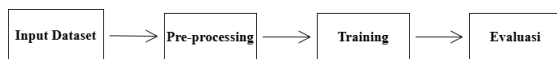
- a. menulis dan menjalankan kode Python.
- b. Mendokumentasikan kode.
- c. Membuat, mengunggah dan membagikan *notebook* Colab.
- d. Mengimpor *notebook* dari Google Drive.
- e. Menyimpan *notebook* ke Google Drive.
- f. Mengimpor dataset eksternal.
- g. Mengintegrasikan TensorFlow, Keras, PyTorch, OpenCV

Google Colab menyediakan mesin virtual untuk tiap pengguna dengan RAM dan Penyimpanan untuk diidentifikasi menggunakan berbagai perintah. Hal

yang membedakan Google Colab adalah menyediakan GPU dan memori yang dibagikan kepada banyak pengguna dengan masing-masing diberikan sesi terpisah. Hal ini memungkinkan pengguna untuk menggunakan sumber daya berdaya tinggi tanpa benar-benar membelinya untuk digunakan. Ia bekerja berdasarkan prinsip sewa dan model bayar per penggunaan [16].

3. METODE PENELITIAN

Metode penelitian kali ini menggunakan Model CNN (*Convolutional Neural Network*) dan Arsitektur VGG-16 dalam melakukan klasifikasi *diabetic retinopathy*. Adapun alur penelitian kali ini dapat dilihat pada gambar di bawah ini.



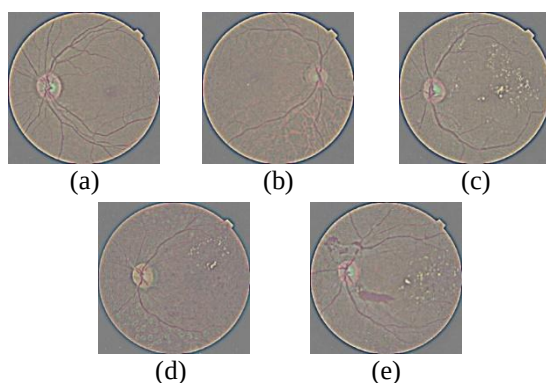
Gambar 3. Diagram blok penelitian

Pada penelitian ini perangkat yang digunakan yaitu CPU core i5 8300H, RAM 16GB, GPU NVIDIA GeForce GTX 1070 8GB. Adapun editor yang digunakan pada penelitian ini yakni Google Colaboratory.

3.1. Dataset

Dataset yang digunakan pada penelitian kali ini yakni berbentuk gambar. Di mana gambar terdiri dari gambar pemindaian retina terfilter gaussian untuk mendeteksi *diabetic retinopathy* sebanyak 3662. Dataset diambil dari Kaggle, di mana data aslinya tersedia di APTOS 2019 *Blindness Detection*. Semua gambar sudah disimpan ke dalam folder masing-masing sesuai dengan tingkat keparahan/tahapan *diabetic retinopathy* dan file train.csv sebagai data label ke dalam Google Drive. Di mana akan menemukan lima direktori dengan masing-masing gambar:

- 0 = *No_DR*
- 1 = *Mild*
- 2 = *Moderate*
- 3 = *Severe*
- 4 = *Proliferate_DR*



Gambar 4. Kelas diabetic retinopathy
a) *No_DR*, b) *Mild*, c) *Moderate*,
d) *Severe*, e) *Proliferate_DR*

Adapun klasifikasi tingkat *diabetic retinopathy* yakni sebagai berikut.

- a. *No_DR*, retina yang normal.
- b. *Mild*, adanya mikroaneurisma yang timbul.
- c. *Moderate*, mikroaneurisma yang timbul semakin banyak.
- d. *Severe*, vena intraretinal mengalami pendarahan.
- e. *Proliferate_DR*, adanya neovaskularisasi [9].

3.2. Pre-processing Data

Dalam *preprocessing* data ini, gambar-gambar pada dataset ini diubah ukurannya menjadi 224x224 piksel sehingga dapat dengan mudah digunakan dengan banyak model pembelajaran mendalam yang telah dilatih sebelumnya.

Pada tahap ini terdapat 2 skenario pengujian. Skenario pertama tanpa menerapkan CLAHE dan augmentasi. Sedangkan skenario kedua dengan menerapkan CLAHE untuk meningkatkan kontras gambar dan melakukan augmentasi dengan membuat objek *ImageDataGenerator*. Adapun augmentasi pada skenario kedua yang digunakan meliputi rotasi, pergeseran, *shear*, *zoom*, dan *flip horizontal*. Kemudian membagi data menjadi *training*, *testing*, dan validasi. Pada tahap ini data juga diacak dan dinormalisasi untuk meningkatkan performa model.

3.3. Persiapan Model

Langkah selanjutnya memuat model VGG-16 yang telah dilatih sebelumnya tanpa lapisan *fully connected*. Lalu membuat model *sequential* baru dengan menambahkan lapisan-lapisan di atas VGG-16. Adapun lapisan yang ditambahkan antara lain:

- a. *Dropout* untuk mencegah *overfitting*.
- b. *Flatten* untuk mengubah output VGG-16 menjadi vektor.
- c. *Dense layers* dengan fungsi aktivasi ReLU.
- d. *Dense layer* terakhir dengan fungsi aktivasi softmax untuk klasifikasi multi-kelas.

Kemudian dilakukan kompilasi model dengan *optimizer* 'adam', *loss function* 'categorical crossentropy', dan metrik 'categorical accuracy'.

3.4. Pelatihan Model

Langkah selanjutnya yakni melatih model, di mana pada skenario pertama menggunakan data pelatihan dan validasi. Sedangkan skenario kedua menggunakan data training yang diaugmentasi dan data validasi. Untuk kedua skenario ini menggunakan *EarlyStopping* untuk menghentikan pelatihan lebih awal jika *val_categorical_accuracy* tidak meningkat selama 10 *epoch* berturut-turut.

3.5. Evaluasi Model

Pada tahap ini akan dilakukan evaluasi model data pengujian, menghitung metrik seperti *accuracy*, *precision*, *recall*, dan *F1-Score*. Kemudian akan menampilkan laporan klasifikasi dan *confusion matrix*. Adapun rumus dari akurasi, presisi, *recall*, dan *F1-score* sebagai berikut [11]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

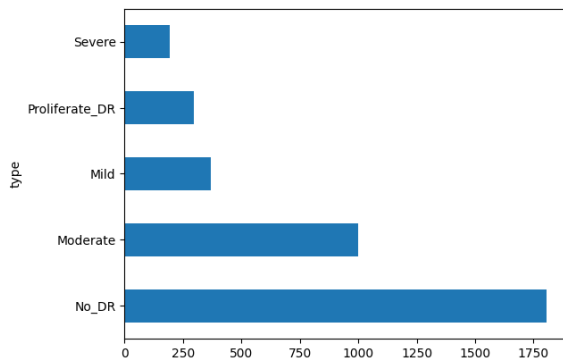
$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (4)$$

Di mana:

- TP (True Positive) = klasifikasi yang benar pada kelas positif
 TN (True Negative) = klasifikasi yang benar dari kelas negatif
 FP (False Positive) = prediksi yang salah pada kelas positif
 FN (False Negative) = prediksi yang salah dari kelas negatif

4. HASIL DAN PEMBAHASAN

Dataset yang digunakan yakni data gambar diabetic retinopathy dengan total 3662 gambar. Dataset diambil dari Kaggle. Semua gambar sudah disimpan ke dalam folder masing-masing sesuai dengan tingkat keparahan/tahapan *diabetic retinopathy* dan file train.csv sebagai data label ke dalam Google Drive. Berikut visualisasi data dari diabetic retinopathy.



Gambar 5. Visualisasi dataset diabetic retinopathy

Dari visualisasi di atas, dapat diketahui jumlah dari masing-masing kelas dataset yakni, *No_DR* sebanyak 1805 gambar, *Mild* sebanyak 370 gambar, *Moderate* sebanyak 999 gambar, *Severe* sebanyak 193 gambar, dan *Proliverate_DR* sebanyak 295 gambar.

4.1. Pre-processing Data

Pada tahap ini ukuran gambar dijadikan 224x224 piksel, lalu dilakukan pengacakan urutan data, dan normalisasi data. Selanjutnya dibuat 2 skenario untuk dijadikan uji coba nantinya.

- Skenario pertama, data langsung dibagi tanpa menambahkan proses augmentasi.
- Skenario kedua, menerapkan CLAHE dan ditambahkan augmentasi menggunakan ImageDataGenerator. Di mana augmentasi yang dilakukan terdiri dari rotasi dengan sudut 20 derajat, menggeser lebar dan tinggi gambar dengan maksimum 20%, dan memperbesar atau memperkecil gambar dengan maksimum 20%.

Setelah itu gambar dibagi menjadi set pelatihan dan set pengujian dengan rasio 80:20, yang bertujuan untuk melatih model dan mengevaluasi model. Selanjutnya membagi data pelatihan menjadi data pelatihan baru dan data validasi dengan rasio 80:20. Di mana hal ini berarti berarti 20% dari data pelatihan awal akan digunakan untuk validasi, dan 80% sisanya tetap sebagai data pelatihan. Adapun tujuannya yaitu untuk mengukur kinerja model dan melakukan penyesuaian *hyperparameter* selama proses pelatihan.

4.2. Persiapan Model

Pada tahap ini, model yang digunakan yaitu VGG-16 dengan tambahan lapisan seperti *dropout*, *flatten*, *dense layer* dengan aktivasi ReLU dan softmax. Adapun *optimizer* yang digunakan yakni Adam, dengan *loss function* 'categorical crossentropy', metrik 'categorical accuracy', dan *learning rate default* (0,001). Berikut lapisan klasifikasi model yang digunakan pada penelitian ini.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
dropout (Dropout)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 64)	1,605,696
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 5)	165

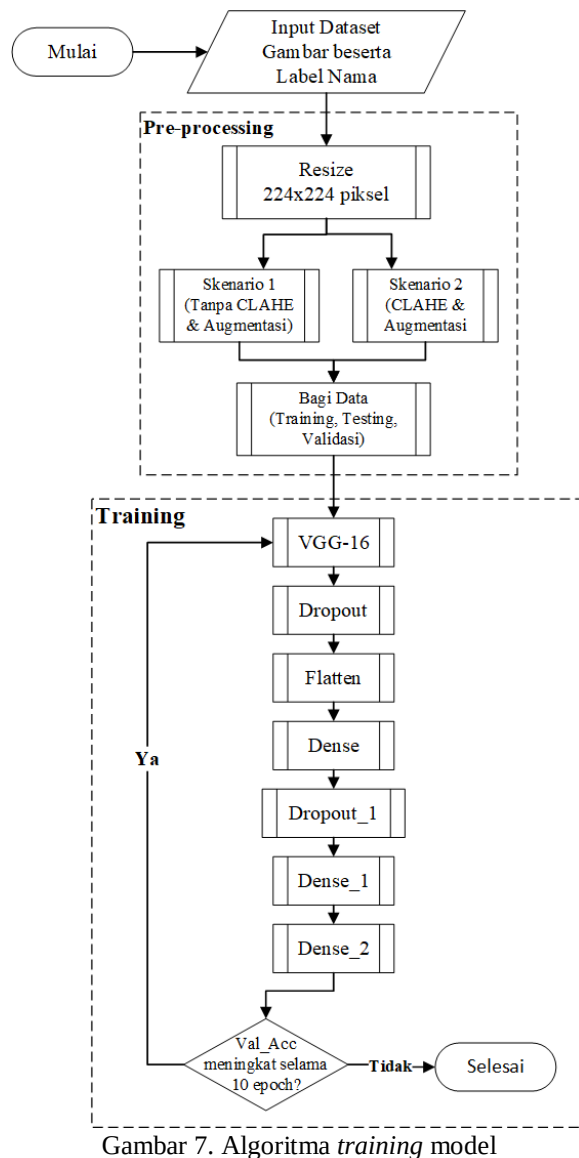
Total params: 16,322,629 (62.27 MB)
 Trainable params: 1,607,941 (6.13 MB)
 Non-trainable params: 14,714,688 (56.13 MB)

Gambar 6. Lapisan model

Pada gambar di atas dapat diketahui bahwa adanya penambahan *layer* pada klasifikasi model kali ini. Adapun layer tambahannya yakni *dropout*, *flatten*, dan *dense*. Di mana dapat dilihat total parameter dalam model sebanyak 16.322.629 parameter. Sedangkan jumlah parameter yang dapat dilatih (diperbarui) selama proses training sebanyak 1.607.941 parameter dan jumlah parameter yang tidak dapat dilatih (dibekukan) selama proses *training* sebanyak 14.714.688 parameter.

4.3. Training Model

Berikut algoritma pelatihan model pada penelitian ini.



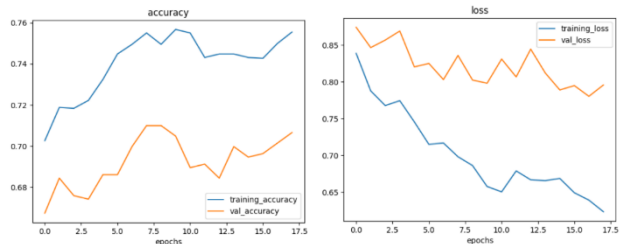
Gambar 7. Algoritma training model

Berdasarkan gambar di atas, dapat diketahui bahwa setelah data melalui proses *pre-processing*, selanjutnya akan dilatih menggunakan arsitektur VGG-16 dengan tambahan *layer* seperti pada gambar. Di mana pada tahap ini menggunakan *EarlyStopping* untuk menghentikan pelatihan lebih awal jika *val_categorical_accuracy* tidak meningkat selama 10 epoch berturut-turut. Selain itu, setelah training dihentikan, model akan dikembalikan ke bobot (*weights*) yang menghasilkan nilai terbaik dari metrik yang dipantau. Adapun jumlah epoch maksimum 50 kali, dengan *batch size* 64 (jumlah sampel data yang akan diproses dalam sekali iterasi).

Hasil dari tahap pelatihan kedua skenario yakni sebagai berikut.

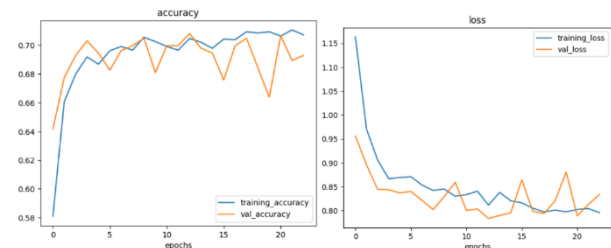
a. Skenario pertama

Pada skenario pertama, jumlah epoch terhenti setelah *epoch* ke 18 dengan hasil akurasi sebesar 0,7465 (75%) dan *loss* 0,7193 (72%). Adapun visualisasi proses *training* sebagai berikut.



Gambar 8. Kurva akurasi dan loss skenario 1

b. Skenario kedua



Gambar 9. Kurva akurasi dan loss skenario 2

Sedangkan pada skenario kedua, jumlah epoch terhenti setelah *epoch* ke 23 dengan hasil akurasi sebesar 0,7784 (78%) dan *loss* 0,6881 (69%). Adapun visualisasi proses *training* sebagai berikut.

4.4. Evaluasi Model

Pada tahap ini dilakukan penghitungan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-Score*. Kemudian akan menampilkan laporan klasifikasi dan *confusion matrix*. Berikut hasil evaluasi masing-masing skenario.

a. Metrik Evaluasi

Tabel 1. Perbandingan metrik evaluasi

Metrik Evaluasi	Skenario 1	Skenario 2
Accuracy	0,7299	0,7613
Precision	0,6671	0,6590
Recall	0,7299	0,7613
F1-Score	0,6854	0,6883

Dapat dilihat hasil metrik evaluasi dari skenario1 yakni memperoleh akurasi sebesar 73%, presisi 67%, *recall* 73%, dan *f1-score* 69%. Sedangkan skenario 2 memperoleh akurasi sebesar 76%, presisi 66%, *recall* 76%, dan *f1-score* 69%.

b. Classification Report

Selanjutnya diperoleh *classification report* pada dari tiap skenario sebagai berikut.

Classification Report:				
	precision	recall	f1-score	support
No_DR	0.00	0.00	0.00	75
Mild	0.56	0.94	0.70	205
Moderate	0.94	0.97	0.96	375
Severe	0.33	0.02	0.04	45
Proliferate_DR	0.00	0.00	0.00	33
accuracy			0.76	733
macro avg	0.37	0.39	0.34	733
weighted avg	0.66	0.76	0.69	733

Gambar 10. Classification report skenario pertama

Dari gambar di atas, pada skenario pertama model memiliki *precision* 0.36, *recall* 0.47, dan *F1-score* 0.41 untuk kelas "Mild". Artinya, dari semua prediksi "Mild", 36% benar-benar "Mild" (*precision*). Model berhasil mengidentifikasi 47% dari semua data "Mild" yang sebenarnya (*recall*). *F1-score* 0.41 menunjukkan keseimbangan antara *precision* dan *recall*. Untuk kelas "Moderate" model memiliki *precision* 0.62, *recall* 0.72, dan *F1-score* 0.67 untuk kelas. Ini menunjukkan kinerja yang lebih baik daripada kelas "Mild". Sedangkan untuk kelas "No_DR" memiliki *precision* 0.89, *recall* 0.99, dan *F1-score* 0.93. Ini menunjukkan kinerja yang sangat baik untuk kelas ini. Pada kelas "Prolifereate_DR" memiliki *precision*, *recall*, dan *F1-score* 0.00. Ini menunjukkan bahwa model tidak dapat mengidentifikasi kelas ini sama sekali. Adapun untuk kelas "Severe" memiliki *precision* 0.40, *recall* 0.04, dan *F1-score* 0.07. Ini menunjukkan kinerja yang buruk untuk kelas ini, terutama *recall* yang sangat rendah.

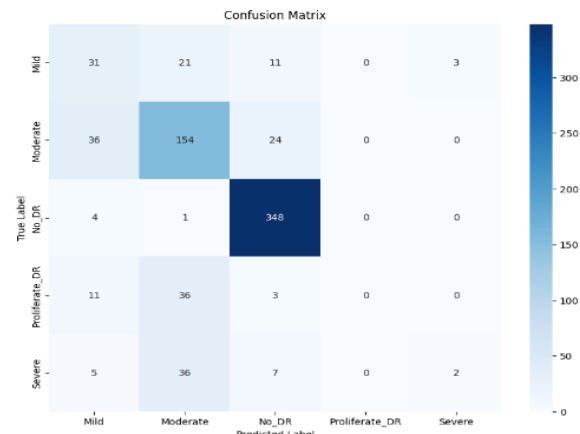
Classification Report:				
	precision	recall	f1-score	support
Mild	0.36	0.47	0.41	66
Moderate	0.62	0.72	0.67	214
No_DR	0.89	0.99	0.93	353
Prolifereate_DR	0.00	0.00	0.00	50
Severe	0.40	0.04	0.07	50
accuracy			0.73	733
macro avg	0.45	0.44	0.42	733
weighted avg	0.67	0.73	0.69	733

Gambar 11. Classification report skenario kedua

Sedangkan pada skenario kedua dapat diketahui model memiliki performa yang baik untuk kelas "Moderate" dengan *precision*, *recall*, dan *f1-score* yang tinggi. Selanjutnya model memiliki performa yang cukup baik untuk kelas "Mild", tetapi *precision*-nya lebih rendah daripada *recall*-nya. Akan tetapi model memiliki performa yang buruk untuk kelas 'No_DR', 'Severe', dan 'Prolifereate_DR' dengan *precision*, *recall*, dan *f1-score* yang sangat rendah atau bahkan 0. Hal ini menunjukkan bahwa model kesulitan dalam mengidentifikasi kelas-kelas tersebut. Sedangkan untuk akurasi (*accuracy*) model sebesar 0.76 menunjukkan bahwa model memprediksi dengan benar 76% dari total data.

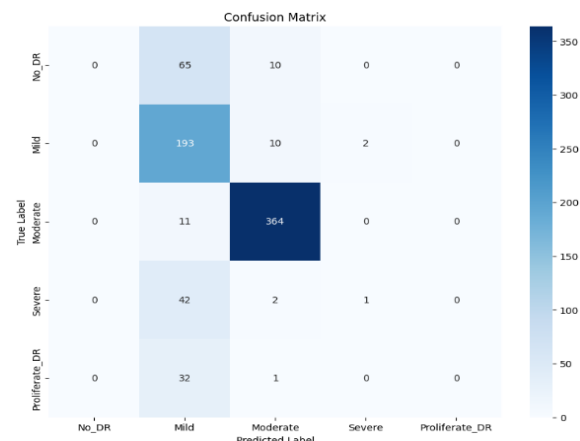
c. Confussion Matrix

Dari gambar 13, dapat diketahui bahwa pada model ini untuk kelas "Mild" 31 (Mild, Mild): True Positive (TP). Ini berarti model memprediksi 31 data sebagai "Mild" dan prediksi tersebut benar. Selanjutnya 10 (Mild, Moderate): False Positive (FP) untuk kelas "Moderate" dan False Negative (FN) untuk kelas "Mild". Ini berarti model memprediksi 10 data sebagai "Moderate" padahal sebenarnya "Mild". Adapun kelas yang mendapatkan True Positive (TP) cukup tinggi yakni kelas (No_DR, No_DR) sebanyak 348 data dan (Moderate, Moderate) sebanyak 154 data.



Gambar 12. Confusion matrix skenario pertama

Selanjutnya ditampilkan *confusion matrix* sebagai berikut.



Gambar 13. Confusion matrix skenario kedua

Dari confusion matrix tersebut dapat diketahui bahwa pada model ini kelas yang mendapatkan True Positive (TP) cukup tinggi yakni kelas "Moderate" (Moderate, Moderate) yang memprediksi 364 data sebagai "Moderate" dengan benar dan kelas "Mild" (Mild, Mild) yang memprediksi sebanyak 193 sebagai "Mild" dengan benar.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil training, skenario kedua dengan penambahan proses augmentasi menghasilkan nilai akurasi yang lebih tinggi dan stabil antara nilai *training* dan validasi, dibandingkan dengan skenario pertama yang memiliki nilai akurasi dan *loss training* dengan akurasi dan *loss validasi* kurang stabil, serta selisih nilai *training* dan validasi yang cukup jauh. Hal ini bisa saja menjadi indikasi *over-fitting*. Selanjutnya berdasarkan metrik evaluasi, menunjukkan bahwa skenario kedua memiliki performa yang cukup baik dengan akurasi yang lebih yakni sekitar 76% dibanding dengan skenario pertama sekitar 73%. Namun, dari kedua skenario memiliki *precision* dan *recall* yang tidak seimbang, menunjukkan bahwa model mungkin masih memiliki kecenderungan untuk

membuat kesalahan dalam memprediksi kelas tertentu. Sedangkan berdasarkan *classification report* dan *confusion matrix*, menunjukkan bahwa keduanya memiliki kinerja yang bervariasi untuk setiap kelas. Pada skenario pertama model berkinerja sangat baik untuk kelas "No_DR" tetapi sangat buruk untuk kelas "Prolifera_DR" dan "Severe". Sedangkan pada skenario kedua model berkinerja sangat baik pada kelas "Moderate" dan "Mild", namun perlu ditingkatkan untuk kelas-kelas yang lain. Terutama untuk kelas "No_DR", "Severe", dan "Prolifera_DR". Berdasarkan penelitian ini, dapat ditarik Kesimpulan yakni penambahan proses augmentasi memberikan peningkatan pada hasil klasifikasi, khususnya data dengan multi-kelas dan variasi yang banyak.

Berdasarkan penelitian ini, masih perlu diselidiki dan ditingkatkan kembali untuk kinerja yang buruk pada kelas-kelas tertentu, misalnya dengan menambah data training atau menyesuaikan arsitektur model. Selain itu data yang digunakan memiliki selisih yang cukup jauh antar kelasnya atau bisa disebut ketidakseimbangan kelas (*class imbalance*), hal ini bisa menjadi penyebab model memiliki performa yang buruk pada kelas-kelas minoritas, yang pada akhirnya mempengaruhi akurasi keseluruhan. Untuk itu perlu adanya perbaikan data, seperti dengan melakukan *oversampling* yakni meningkatkan jumlah data pada kelas minoritas dengan menduplikasi atau melakukan *undersampling* yakni mengurangi jumlah data pada kelas mayoritas.

DAFTAR PUSTAKA

- [1] S. N. Shivapriya, M. Alagumeenaakshi, dan S. Sasikala, "Deep Learning-Based Diabetic Retinopathy Severity Classification and Progression Time Estimation," dalam *IFAC-PapersOnLine*, Elsevier B.V., Jun 2024, hlm. 78–83. doi: 10.1016/j.ifacol.2024.07.129.
- [2] G. A. Mersha, Y. A. Alimaw, dan A. T. Woredikal, "Prevalence of diabetic retinopathy among diabetic patients in Northwest Ethiopia-A cross sectional hospital based study," *PLoS One*, vol. 17, no. 1 January, Jan 2022, doi: 10.1371/journal.pone.0262664.
- [3] Z. L. Teo dkk., "Global Prevalence of Diabetic Retinopathy and Projection of Burden through 2045: Systematic Review and Meta-analysis," *Ophthalmology*, vol. 128, no. 11, hlm. 1580–1591, Nov 2021, doi: 10.1016/j.ophtha.2021.04.027.
- [4] S. H. Abdullah, R. Magdalena, dan R. Y. N. Fu'adah, "KLASIFIKASI DIABETIC RETINOPATHY BERBASIS PENGOLAHAN CITRA FUNDUS DAN DEEP LEARNING," *JOURNAL OF ELECTRICAL AND SYSTEM CONTROL ENGINEERING*, vol. 5, no. 2, hlm. 84–90, Feb 2022, doi: 10.31289/jesce.v5i2.5659.
- [5] Y. Harjoseputro, "CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK PENGKLASIFIKASIAN AKSARA JAWA," UNIVERSITAS ATMA JAYA YOGYAKARTA.
- [6] M. H. V. Sinaga, M. Albirra, dan M. F. Sidiq, "Klasifikasi Gambar Pemandangan dengan Kecerdasan Buatan Berbasis CNN," *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 8, no. 2, 2024.
- [7] J. V. N. Rakamawati, "KLASIFIKASI DIABETIC RETINOPATHY BERDASARKAN FOTO FUNDUS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) JENIS DENSENET," UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA, Surabaya, 2021.
- [8] A. Faizin dan M. Lutfi, "PERBANDINGAN ARSITEKTUR LENET DAN GOOGLNET DALAM KLASIFIKASI DIABETIC RETINOPATHY PADA CITRA RETINA FUNDUS," *Jurnal Mahasiswa Teknik Informatika*, vol. 6, no. 1, 2022, [Daring]. Tersedia pada: www.kaggle.com
- [9] K. A. H. Wibowo dan Alamsyah, "Klasifikasi Diabetic Retinopathy Menggunakan CNN dengan Arsitektur yang Dimodifikasi," *Indones. J. Math. Nat. Sci*, vol. 46, no. 1, hlm. 45–52, 2023, [Daring]. Tersedia pada: <http://journal.unnes.ac.id/nju/index.php/JM>
- [10] T. Berliani, E. Rahardja, dan L. Septiana, "Perbandingan Kemampuan Klasifikasi Citra X-ray Paru-paru menggunakan Transfer Learning ResNet-50 dan VGG-16," *Journal of Medicine and Health*, vol. 5, no. 2, hlm. 123–135, Agu 2023, doi: 10.28932/jmh.v5i2.6116.
- [11] W. Wijaya Kusuma, R. Rizal Isnanto, A. Fauzi, dan P. Korespondensi, "Analisis Perbandingan Model CNN VGG16 dan DenseNet121 Menggunakan Kerangka Kerja TensorFlow untuk Deteksi Jenis Hewan," *Jurnal Teknik Komputer*, vol. 1, no. 4, hlm. 141–147, 2023, doi: 10.14710/jtk.v1i4.37009.
- [12] L. Alzubaidi dkk., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Des 2021, doi: 10.1186/s40537-021-00444-8.
- [13] J. E. Widyaya dan S. Budi, "Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan Pendekatan Transfer Learning Convolutional Neural Network," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr 2021, doi: 10.28932/jutisi.v7i1.3327.
- [14] Y. Zidane, N. Betha, I. Sari, A. Maulana, dan G. Primaya, "PENERAPAN DATA MINING DALAM KLASIFIKASI DATA TRANSAKSI PRODUK KOPERASI DI SMK PGRI 2 KARAWANG," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 1, hlm. 263, 2025