

# Perbandingan Penggunaan *Optimizer* dalam Klasifikasi Sel Darah Putih Menggunakan *Convolutional Neural Network*

Dede Kurniadi<sup>1</sup>, Rifky Muhammad Shidiq<sup>1</sup>, Asri Mulyani<sup>1</sup>

<sup>1</sup> Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Institut Teknologi Garut, Garut, Jawa Barat 4415, Indonesia

[Diserahkan: 19 November 2024, Direvisi: 21 Januari 2025, Diterima: 24 Februari 2025]  
Penulis Korespondensi: Dede Kurniadi (email: dede.kurniadi@itg.ac.id)

**INTISARI** — Sel darah putih adalah komponen penting dari sistem kekebalan tubuh yang bertugas melawan infeksi dan penyakit. Klasifikasi dan penghitungan sel darah putih biasanya dilakukan secara manual oleh operator berpengalaman atau menggunakan analisis sel darah otomatis. Metode manual tidak efisien, memakan waktu, dan membutuhkan banyak tenaga kerja, sedangkan mesin analisis otomatis sering kali mahal dan memiliki persyaratan tinggi untuk sampel uji. Penelitian ini bertujuan untuk membandingkan kinerja tiga jenis *optimizer*, yaitu *root mean square propagation* (RMSProp), *stochastic gradient descent* (SGD), dan *adaptive moment estimation* (Adam), dalam model klasifikasi sel darah putih menggunakan algoritma *convolutional neural network* (CNN). *Dataset* yang digunakan terdiri atas 12.392 citra yang mencakup empat kelas sel darah putih, yaitu eosinofil, neutrofil, limfosit, dan monosit. Hasil penelitian menunjukkan bahwa *optimizer* Adam memberikan kinerja terbaik dengan akurasi pelatihan mencapai 98,65% dan akurasi evaluasi sebesar 97,73%. Adam juga unggul dalam metrik lainnya, seperti *recall* (97,43%), *presisi* (97,42%), *F1-score* (97,42%), dan spesifisitas (99,11%). Nilai AUC untuk setiap kelas melebihi 90%, menunjukkan kemampuan model yang sangat baik dalam membedakan data antarkelas. *Optimizer* RMSProp menghasilkan akurasi pelatihan sebesar 98,63%, sedangkan SGD menunjukkan akurasi pelatihan lebih rendah, yaitu 83,46%. Penelitian ini menunjukkan bahwa pemilihan *optimizer* yang tepat berpengaruh pada kinerja CNN dalam klasifikasi citra sel darah putih, sehingga dapat menjadi langkah awal untuk pengembangan sistem klasifikasi medis yang lebih akurat.

**KATA KUNCI** — *Optimizer* Adam, *Convolutional Neural Network*, Klasifikasi Sel Darah Putih, *Optimizer* RMSProp, *Optimizer* SGD.

## I. PENDAHULUAN

Sel darah putih, atau leukosit, adalah komponen penting dari sistem kekebalan tubuh yang bertugas melawan infeksi dan penyakit. Terdapat beberapa jenis sel darah putih, termasuk neutrofil, limfosit, monosit, eosinofil, dan basofil, yang masing-masing memiliki peran spesifik dalam respons imun. Leukosit diproduksi di sumsum tulang dan beredar di seluruh tubuh melalui aliran darah. Kadar dan proporsi jenis sel darah putih yang berbeda dalam darah dapat memberikan indikasi penting tentang kesehatan seseorang dan membantu dalam diagnosis berbagai kondisi medis [1]. Setiap sel darah putih memiliki karakteristik unik dalam hal warna dan bentuknya. Neutrofil memiliki warna merah kebiruan dan cenderung memiliki tiga inti sel dengan berbagai bentuknya. Basofil memiliki warna yang tampak seperti bintik-bintik kebiruan. Eosinofil menampilkan warna kemerahan dalam bintik-bintiknya, monosit memiliki warna biru dengan inti sel yang berbentuk bulat panjang, sedangkan limfosit memiliki warna biru pucat dan tidak mampu bergerak bebas [2].

Klasifikasi dan penghitungan sel darah putih biasanya dilakukan secara tradisional oleh operator berpengalaman secara manual atau menggunakan analisis sel darah otomatis sepenuhnya. Namun, metode penghitungan manual tidak efisien, memakan waktu, dan membutuhkan banyak tenaga kerja. Selain itu, meskipun analisis sel darah otomatis telah digunakan untuk menganalisis sel darah putih, sering kali alat tersebut memiliki persyaratan yang tinggi untuk sampel uji dan harganya mahal, sehingga penggunaannya tidak dapat dilakukan secara luas di fasilitas layanan kesehatan atau rumah

sakit di kota [3]. Akan tetapi, perkembangan teknologi saat ini dapat digunakan untuk mempermudah klasifikasi citra sel darah putih, salah satunya adalah melalui *deep learning*. *Deep learning* merupakan bagian dari pembelajaran mesin yang melibatkan penggunaan jaringan saraf tiruan yang terdiri atas berbagai lapisan. Jaringan saraf dalam *deep learning* dapat memiliki banyak lapisan, sehingga memungkinkan dilakukannya ekstraksi fitur kompleks dari data yang diberikan. Hal ini membuat *deep learning* dapat diterapkan dalam berbagai aplikasi, termasuk dalam analisis citra medis seperti klasifikasi sel darah putih [4].

Salah satu algoritma dalam *deep learning* adalah *convolutional neural network* (CNN). CNN bekerja dengan menggunakan lapisan konvolusi untuk mengekstraksi fitur-fitur penting dari citra, seperti tepi, tekstur, dan pola. Lapisan-lapisan ini kemudian menggabungkan fitur-fitur tersebut untuk mengurangi dimensi data, sehingga memungkinkan jaringan untuk mengidentifikasi pola-pola yang lebih kompleks dan mengklasifikasikan citra ke dalam kategori yang berbeda. Selain lapisan konvolusi, CNN juga terdiri atas lapisan *pooling* dan *fully connected*. Lapisan *pooling* berfungsi untuk mengurangi ukuran representasi dan mengontrol *overfitting* dengan melakukan operasi seperti *max-pooling* atau *average-pooling*. Sementara itu, lapisan *fully connected* berperan dalam membuat keputusan akhir dengan menggabungkan semua fitur yang diekstraksi untuk memberikan hasil klasifikasi [5].

Dalam penerapan algoritma CNN untuk klasifikasi citra, pemilihan *optimizer* memainkan peran penting dalam proses pelatihan model. *Optimizer* adalah algoritma yang digunakan

untuk mengatur pembaruan bobot dan *bias* model dengan tujuan meminimalkan fungsi *loss* selama proses pelatihan. Beberapa *optimizer* yang umum digunakan dalam konteks *deep learning*, yaitu *stochastic gradient descent* (SGD), *adaptive moment estimation* (Adam), dan *root mean square propagation* (RMSProp), memiliki pendekatan berbeda dalam menangani gradien selama proses pelatihan [6]. SGD bersifat sederhana dan ringan secara komputasi, tetapi sering kali membutuhkan iterasi lebih banyak untuk mencapai konvergensi. Adam dan RMSProp, di sisi lain, menawarkan keunggulan dalam kecepatan konvergensi melalui pendekatan adaptif. Penelitian ini bertujuan untuk membandingkan kinerja ketiga *optimizer* dalam tugas klasifikasi citra sel darah putih, yang melibatkan empat kelas sel darah putih, yaitu eosinofil, neutrofil, limfosit, dan monosit. Evaluasi dilakukan menggunakan metrik seperti *confusion matrix* dan *receiver operating characteristic area under the curve* (ROC-AUC), yang mencakup nilai akurasi, presisi, *recall*, *f1-score*, dan kurva ROC-AUC. *Dataset* yang digunakan dalam penelitian ini diperoleh dari *website* penyedia *dataset online*, yaitu Kaggle, dengan nama *dataset* Blood Cells Images, sejumlah 12.392 citra dan terdiri atas empat kelas sel darah putih, yaitu eosinofil, neutrofil, limfosit, dan monosit [7]. Dengan memahami pengaruh setiap *optimizer* terhadap akurasi, diharapkan dapat diperoleh pemahaman yang lebih baik mengenai *optimizer* yang paling cocok untuk klasifikasi sel darah putih.

## II. PENELITIAN TERKAIT

Beberapa penelitian sebelumnya telah mengembangkan model CNN *fully connected* yang terdiri atas enam lapisan konvolusi, enam lapisan *pooling*, dan dua lapisan *fully connected*. Model tersebut diuji menggunakan *dataset* Blood Cell Count and Detection (BCCD) yang diperoleh dari Kaggle dengan total 285 citra, menghasilkan akurasi 96,84% [8].

Penelitian lain mengusulkan pendekatan inovatif untuk klasifikasi citra sel darah putih dengan menerapkan metode *meta learning* dan *color constancy*. Metode *meta learning* berbasis model Xception mencapai akurasi tertinggi sebesar 96,47% dengan menggunakan *dataset* Raabin yang mencakup 10.175 citra [9]. Selain itu, terdapat penelitian yang memanfaatkan algoritma *decision tree* untuk melakukan klasifikasi pada 167 citra sel darah putih yang diperoleh melalui komunikasi dengan rumah sakit, menghasilkan akurasi sebesar 92,2% [10].

Penelitian lain menerapkan algoritma *convolutional features - support vector machines* (Con-SVM) untuk mengklasifikasikan citra sel darah putih menggunakan *dataset* sebanyak 12.442 citra yang diunduh dari Kaggle. Akurasi yang dihasilkan mencapai 85,96% [11]. Selanjutnya, ada penelitian yang menggunakan algoritma *k-nearest neighbors* (KNN) untuk mendeteksi kelainan pada sel darah putih sebagai upaya deteksi dini sindrom *neoplasma mieloproliferatif*. Dari total 159 data uji, 150 berhasil diklasifikasikan sesuai target, sehingga menghasilkan akurasi 94,3% [12].

Terakhir, sebuah penelitian mengembangkan metode klasifikasi sel darah putih dengan memanfaatkan *deep dilated residual convolutional neural network* (DDRNet), yang menghasilkan akurasi sebesar 91,98% serta *F1-score* sebesar 0,96, dengan menggunakan *dataset* sebanyak 16.249 citra yang diambil dari Kaggle [13].

Berdasarkan Tabel I, terlihat bahwa berbagai metode yang digunakan untuk klasifikasi citra sel darah putih, seperti CNN *fully connected*, *meta learning*, *decision tree*, Con-SVM, dan

TABEL I  
ANALISIS KESENJANGAN PENELITIAN KLASIFIKASI CITRA SEL DARAH PUTIH

Penelitian	Metode	Kelebihan	Kekurangan
[8]	<i>Fully connected</i> CNN	Akurasi tinggi (96,84%)	Arsitektur rumit meningkatkan waktu komputasi dan biaya.
[9]	<i>Meta learning</i> dan <i>color constancy</i>	Akurasi 96,47% dengan <i>meta learning</i>	Rentan terhadap perubahan kontras dan variasi warna.
[10]	<i>Decision tree</i>	Mudah diimplementasikan, akurasi 92,2%	Kurang efektif pada data <i>noisy</i> atau fitur kompleks.
[11]	Con-SVM	Meningkatkan generalisasi	Akurasi rendah (85,96%) dibandingkan metode lain. Sensitif terhadap <i>outlier</i> dan perlu pemilihan parameter K yang tepat.
[12]	KNN	Sederhana, akurasi 94,3%	Membutuhkan komputasi tinggi karena kompleksitas arsitektur.
[13]	DDRNet	Akurasi tinggi (91,98%) dan <i>f1-score</i> (0,96), baik untuk fitur kompleks	

DDRNet, menunjukkan variasi dalam akurasi dan efisiensi. Meskipun beberapa metode berhasil menghasilkan akurasi tinggi, seperti CNN *fully connected* (96,84%) dan *meta learning* berbasis Xception (96,47%), ada tantangan yang mencakup kompleksitas model, sensitivitas terhadap perubahan kontras, serta ketidakstabilan terhadap derau dalam citra. Beberapa metode seperti *decision tree* dan Con-SVM menunjukkan keterbatasan dalam menangani fitur yang lebih kompleks dan menghasilkan akurasi yang lebih rendah.

Kesenjangan ini menunjukkan bahwa meskipun berbagai metode telah dikembangkan, ada kebutuhan untuk eksplorasi lebih lanjut dalam hal efisiensi komputasi dan ketegaran model, terutama dalam konteks klasifikasi citra medis yang memerlukan hasil akurasi dan stabilitas yang baik. Dalam konteks penelitian ini, penggunaan *optimizer* seperti Adam, SGD, dan RMSProp untuk klasifikasi citra sel darah putih pada CNN dipilih karena setiap *optimizer* memiliki keunggulan tersendiri dalam menangani berbagai tantangan pada proses pelatihan model. Adam dikenal dengan kemampuannya untuk mempercepat proses konvergensi [14], RMSProp menawarkan stabilitas pada gradien yang fluktuatif [15], sedangkan SGD masih sering digunakan sebagai *baseline* karena sederhana [16]. Dengan membandingkan ketiga *optimizer* tersebut, penelitian ini bertujuan untuk menemukan kombinasi terbaik yang dapat meningkatkan kinerja klasifikasi, baik dari segi akurasi maupun efisiensi komputasi.

Pemilihan *optimizer* yang tepat sangat penting karena dapat memperbaiki beberapa kelemahan yang teridentifikasi dalam metode sebelumnya, seperti stabilitas model saat menghadapi variasi data, kecepatan konvergensi, dan kemampuan generalisasi model terhadap data yang berbeda-beda. Dengan demikian, penelitian ini berfokus pada optimalisasi CNN menggunakan tiga *optimizer* tersebut untuk mencapai keseimbangan antara akurasi dan efisiensi proses komputasi dalam klasifikasi citra sel darah putih.

### III. METODOLOGI

Penelitian ini mengadopsi strategi studi eksperimental untuk menguji model klasifikasi citra sel darah putih menggunakan CNN dan mengevaluasi kinerjanya. Proses ini mencakup perancangan langkah-langkah yang diterapkan selama penelitian.

Alur penelitian diawali dengan mencari *dataset* citra sel darah putih dari sumber *online*, yaitu Kaggle. *Dataset* tersebut kemudian dibagi menjadi tiga bagian, yaitu data latih, data uji, dan data validasi. Pembagian ini dilakukan untuk memastikan model dapat memiliki kemampuan generalisasi yang baik terhadap data baru yang belum pernah dilihat sebelumnya serta untuk mengukur kinerja model secara akurat.

Tahap prapemrosesan dilakukan untuk mempersiapkan *dataset* agar sesuai dengan kebutuhan model CNN. Proses ini meliputi penyeimbangan (*balancing*) *dataset* menggunakan teknik *random over sampling* (ROS), pengubahan ukuran (*resizing*) citra untuk memastikan konsistensi ukuran citra, segmentasi citra untuk memisahkan sel darah putih dari latar belakang, serta normalisasi nilai piksel. Normalisasi dilakukan untuk membuat distribusi nilai piksel menjadi lebih seragam dan mengurangi variasi yang tidak diinginkan dalam data.

Pada tahap berikutnya, arsitektur model CNN dibangun dan dilatih menggunakan data latih serta data validasi. Model ini kemudian dioptimalkan dengan menggunakan tiga jenis *optimizer*, yaitu Adam, RMSProp, dan SGD. Setelah model dilatih dan parameter dioptimalkan, data uji digunakan untuk mengevaluasi kinerja akhir model. Evaluasi ini bertujuan untuk memberikan gambaran objektif mengenai kemampuan model dalam menggeneralisasi serta mengklasifikasikan data baru.

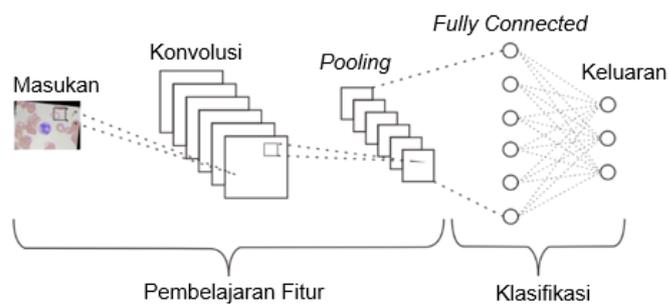
Tahapan evaluasi dilakukan dengan menggunakan *confusion matrix*. Pada evaluasi ini, beberapa metrik digunakan untuk menilai kinerja model, seperti akurasi, presisi, *recall*, *f1-score*, spesifisitas, dan ROC-AUC. Metrik ini digunakan untuk mengukur kemampuan model dalam memisahkan kelas positif dan negatif secara akurat.

Bagian hasil dan analisis memaparkan alur hasil penelitian dari implementasi CNN dengan membandingkan kinerja tiga *optimizer* dalam klasifikasi citra sel darah putih. Hasil penelitian dari setiap tahapan proses dijelaskan untuk memberikan gambaran lengkap mengenai efektivitas model dalam tugas klasifikasi.

#### A. CONVOLUTIONAL NEURAL NETWORK

CNN merupakan algoritma yang terkenal dan sering digunakan dalam *deep learning* karena kemampuannya untuk secara otomatis mengidentifikasi fitur-fitur yang relevan tanpa pengawasan manusia. CNN telah banyak digunakan dalam berbagai bidang, seperti visi komputer, pemrosesan suara, dan pengenalan wajah. Struktur CNN terinspirasi oleh neuron dalam otak manusia dan hewan, menggunakan bobot bersama serta koneksi lokal untuk memanfaatkan struktur data masukan 2D, seperti sinyal citra [17]. Dalam visi komputer, CNN terus menjadi pilihan utama untuk tugas-tugas klasifikasi citra. Penelitian terbaru memperkenalkan arsitektur *vision transformer* (ViT), yang menggabungkan CNN untuk tugas visi komputer. Studi ini menunjukkan bahwa kombinasi CNN dan *transformer* dapat meningkatkan kinerja dalam klasifikasi citra [18].

Pengembangan arsitektur CNN telah meningkatkan akurasi sistem pengenalan wajah. Salah satu studi menunjukkan bahwa CNN efektif dalam mengatasi tantangan seperti pencahayaan, sudut pandang, dan ekspresi wajah, sehingga mendukung



Gambar 1. Convolutional neural network.

penerapan sistem pengenalan wajah untuk keamanan dan autentikasi biometrik [19]. Gambar 1 memperlihatkan contoh arsitektur CNN.

Berdasarkan Gambar 1 tahap pembelajaran fitur, terdapat struktur yang menerima citra sebagai masukan awal dan melalui lapisan konvolusi serta *pooling*, lalu mengolahnya menjadi *feature map* numerik yang merepresentasikan citra. Setiap langkah dalam lapisan ini menghasilkan *feature map* yang memperkuat representasi citra sebelum diteruskan ke tahap klasifikasi. Pada tahap klasifikasi, terdapat sejumlah lapisan yang terdiri atas neuron yang terhubung sepenuhnya (*fully connected*) dengan lapisan sebelumnya. Lapisan-lapisan ini menerima masukan dari lapisan keluaran pada tahap pembelajaran fitur. Kemudian, *feature map* ini diolah dan disesuaikan melalui serangkaian lapisan tersembunyi dalam jaringan saraf, menghasilkan tingkat akurasi klasifikasi untuk setiap kelas.

Lapisan konvolusi adalah komponen penting dalam CNN yang bertanggung jawab untuk melakukan operasi konvolusi pada hasil keluaran dari lapisan sebelumnya. Di dalamnya, terdapat serangkaian filter yang dipelajari secara acak untuk mengekstrak fitur dari data masukan. Tujuannya adalah untuk memperoleh representasi fitur yang penting dari data masukan, khususnya pada citra.

Lapisan *pooling* atau *subsampling* umumnya mengikuti lapisan konvolusi dalam CNN. Fungsinya adalah untuk mereduksi dimensi keluaran dari lapisan konvolusi sepanjang dimensi spasial tinggi dan lebar. Tujuan utamanya adalah untuk mengurangi jumlah parameter yang perlu dipelajari oleh jaringan, yang juga membantu mengatasi *overfitting* dan meningkatkan kinerja serta akurasi keseluruhan dari jaringan.

Fungsi *rectified linear unit* (ReLU) adalah fungsi nonlinear yang paling umum digunakan dalam jaringan saraf saat ini. ReLU menghasilkan nilai 0 untuk semua nilai negatif  $x$ , sedangkan untuk nilai positif  $x$ , nilainya tetap sama. Fungsi ini ditunjukkan pada (1).

$$f_{relu}(x) = \max(0, x) \quad (1)$$

dengan  $x$  adalah masukan fungsi, sedangkan  $\max(0, x)$  adalah fungsi maksimum untuk memilih nilai yang lebih besar antara 0 dan  $x$ .

Lapisan *fully connected* merupakan komponen umum dalam jaringan saraf, berbeda dengan lapisan konvolusi karena tidak melibatkan operasi konvolusi dalam prosesnya untuk menghasilkan keluaran. Lapisan *fully connected* setiap neuron terhubung langsung ke setiap neuron dalam lapisan sebelumnya, menciptakan koneksi yang lebih kompleks dan memungkinkan jaringan untuk mempelajari hubungan yang lebih abstrak antara fitur-fitur yang diwakilkan oleh aktivasi. Lapisan *fully connected* ditempatkan di dua lapisan terakhir

dari jaringan dan menggunakan fungsi aktivasi *softmax* untuk menentukan probabilitas keluaran berdasarkan masukan yang diberikan [20].

### B. ADAPTIVE MOMENT ESTIMATION (ADAM)

*Optimizer* Adam adalah salah satu algoritma optimasi yang digunakan secara luas dalam pelatihan model *deep learning*. Algoritma ini menggabungkan keuntungan dari *adaptive gradient* (AdaGrad) dan RMSProp dengan mengatur secara dinamis laju eksponensial untuk estimasi momen pertama (rerata) dan kedua (varians) dari gradien untuk memperbarui parameter. *Optimizer* Adam sangat cocok untuk data yang besar dan optimasi objektif yang tidak stasioner dengan gradien yang berderau dan jarang. Dibandingkan dengan algoritma optimasi lain, Adam menunjukkan kinerja yang baik dalam konvergensi cepat dan stabilitas selama pelatihan, terutama ketika menangani masalah dengan banyak parameter dan fungsi *loss* yang mungkin nonkonveks [21]. Penggunaan Adam juga digunakan dalam optimasi model *deep learning*, khususnya pada model medis. Sebuah penelitian menunjukkan bahwa Adam mempercepat konvergensi dan menjaga akurasi, bahkan pada *dataset* medis besar dan kompleks [14]. Rumus untuk menghitung *optimizer* Adam dinyatakan dalam (2) sampai (6).

Estimasi momen pertama (rata-rata):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2)$$

estimasi momen kedua (varians tidak terpusat):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3)$$

estimasi momen pertama yang telah dikoreksi *bias*:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (4)$$

estimasi momen kedua yang telah dikoreksi *bias*:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (5)$$

pembaruan parameter:

$$\hat{\theta}_t = \theta_{t-1} - a \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (6)$$

dengan  $a$  adalah laju pembelajaran,  $\beta_1$  dan  $\beta_2$  adalah laju peluruhan untuk estimasi momen, dan  $\epsilon$  adalah konstanta kecil untuk stabilitas numerik.

### C. ROOT MEAN SQUARE PROPAGATION (RMSProp)

RMSProp adalah algoritma optimisasi yang dirancang untuk mempercepat proses pelatihan model dalam *deep learning*. Algoritma ini merupakan pengembangan dari metode SGD yang bertujuan untuk mengatasi masalah penurunan laju pembelajaran yang terlalu lambat dan fluktuasi besar dalam pembaruan parameter. RMSProp bekerja dengan cara menyimpan rata-rata kuadrat dari gradien yang telah dihitung pada setiap iterasi, sehingga dapat menyesuaikan laju pembelajaran untuk setiap parameter secara individual [15]. Dalam beberapa tahun terakhir, RMSProp diakui unggul dalam berbagai aplikasi *deep learning*. Penelitian menunjukkan bahwa algoritma ini lebih baik daripada Adam, khususnya pada tugas optimasi nonkonveks, sehingga cocok untuk arsitektur kompleks dan *dataset* sulit [22]. Rumus untuk menghitung *optimizer* RMSProp dinyatakan dalam (7) dan (8).

Rata-rata bergerak dari kuadrat gradien:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (7)$$

Pembaruan parameter:

$$\theta_t = \theta_{t-1} - a \frac{g_t}{\sqrt{v_t + \epsilon}} \quad (8)$$

dengan  $\alpha$  adalah laju pembelajaran,  $\beta$  adalah laju peluruhan untuk rata-rata bergerak dari gradien kuadrat, dan  $\epsilon$  adalah konstanta kecil untuk stabilitas numerik.

### D. STOCHASTIC GRADIENT DESCENT (SGD)

SGD adalah *optimizer* yang sederhana, tetapi sangat efektif dalam konteks pembelajaran mesin. SGD melakukan pembaruan parameter dengan menggunakan satu contoh acak dari *dataset* pada setiap iterasi, yang mengurangi waktu komputasi dan memori yang diperlukan dibandingkan dengan metode *gradient descent batch* yang menghitung gradien untuk seluruh *dataset* sebelum melakukan pembaruan. Meskipun SGD dapat mempercepat proses pelatihan, penggunaan satu contoh acak juga memperkenalkan derau ke dalam estimasi gradien, yang dapat menyebabkan fluktuasi dalam jalur konvergensi dan membuat proses pelatihan menjadi lebih tidak stabil. Namun, fluktuasi ini juga dapat membantu algoritma untuk keluar dari *saddle points* dan menemukan solusi yang lebih baik [16]. Penelitian terbaru telah menunjukkan bahwa variasi dari SGD, seperti SGD dengan momentum atau dengan adaptasi laju pembelajaran, makin memperkuat keunggulan metode ini dalam berbagai aplikasi *deep learning*, termasuk pengenalan citra dan deteksi objek [23]. Rumus untuk menghitung *optimizer* SGD dinyatakan dalam (9).

Pembaruan parameter:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (9)$$

dengan  $\alpha$  adalah pembelajaran mesin, sedangkan  $g_t$  gradien dari fungsi objektif terhadap parameter pada langkah waktu  $t$ .

### E. SEGMENTASI CITRA

Segmentasi citra merupakan proses penting dalam teknologi visi komputer, yang bertujuan untuk membagi citra menjadi beberapa wilayah dengan karakteristik tertentu. Hal ini memungkinkan identifikasi dan ekstraksi target yang diinginkan dari citra, yang menjadi langkah penting dalam pemrosesan citra hingga analisis citra. Segmentasi memungkinkan dilakukannya pemisahan antara objek yang diminati (sasaran) dengan latar belakangnya, yang memiliki beragam aplikasi di berbagai bidang, seperti pemrosesan citra medis, pengenalan pola, dan kecerdasan buatan [24]. Dalam pemrosesan citra medis, misalnya, metode segmentasi berbasis *deep learning* telah meningkatkan akurasi deteksi dalam diagnosis kanker paru-paru dari citra *computed tomography* (CT) [25]. Selain itu, penggunaan segmentasi untuk pemetaan jalan dalam citra satelit juga memberikan hasil yang lebih baik dalam sistem navigasi otomatis [26].

### F. CONFUSION MATRIX

*Confusion matrix* adalah tabel berukuran  $N \times N$  yang digunakan dalam konteks klasifikasi untuk mengevaluasi kinerja model. Nilai  $N$  pada tabel ini mewakili jumlah kelas yang diprediksi. Dengan membandingkan hasil prediksi model terhadap nilai sebenarnya, *confusion matrix* menyediakan ringkasan komprehensif mengenai akurasi model serta mengidentifikasi area-area yang perlu ditingkatkan dalam proses klasifikasi [27]. Matriks ini terdiri atas empat komponen utama: *true positive* (TP), *false negative* (FN), *false positive* (FP), dan *true negative* (TN). Dari kombinasi nilai-nilai ini, beberapa metrik evaluasi, seperti akurasi, presisi, *recall*, dan *f1-*

score dapat dihitung untuk mengevaluasi kinerja model secara menyeluruh.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Akurasi adalah rasio antara jumlah prediksi yang benar (baik positif maupun negatif) terhadap total jumlah prediksi.

$$Presisi = \frac{TP}{TP+FP} \quad (11)$$

Presisi mengukur ketepatan model dalam memprediksi kelas positif. Makin tinggi presisi, makin sedikit prediksi positif yang salah (*false positive*).

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

Recall atau sensitivitas adalah kemampuan model untuk menangkap semua sampel positif yang ada.

$$F1 - Score = \frac{2 \times precision \times recall}{precision+recall} \quad (13)$$

F1-score adalah rata-rata harmonis dari presisi dan recall. Metrik ini digunakan ketika menyeimbangkan presisi dan recall.

$$Spesifisitas = \frac{TN}{TN+FP} \quad (14)$$

Spesifisitas mengukur kemampuan model untuk memprediksi kelas negatif dengan benar.

### G. ROC-AUC

Kurva ROC-AUC digunakan untuk mengukur kemampuan sebuah model klasifikasi dalam membedakan antara kelas-kelas yang berbeda dalam berbagai skenario *threshold* [28]. Evaluasi untuk menghitung nilai AUC ini menghasilkan (15).

$$AUC = \frac{1}{2} (Sensitivitas/Recall + Spesifitas) \quad (15)$$

AUC adalah ukuran kemampuan model untuk membedakan antara kelas positif dan negatif pada berbagai ambang batas.

## IV. HASIL DAN PEMBAHASAN

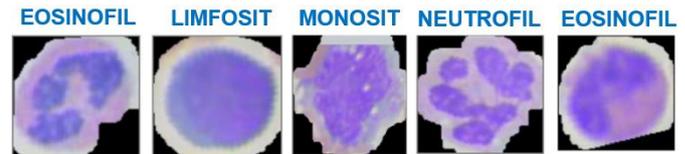
Pada bagian ini, disajikan hasil penelitian dari implementasi algoritma CNN dengan membandingkan tiga *optimizer* untuk klasifikasi citra sel darah putih. Alur proses setiap tahapan dijelaskan sebagai berikut.

### A. DATA SOURCE

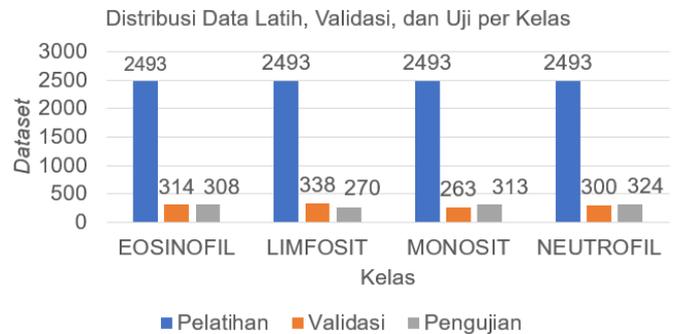
Langkah pertama yang dilakukan pada tahapan *data source* adalah pengumpulan *dataset* dari penyedia *dataset online*, yaitu Kaggle. Gambar 2 memperlihatkan empat sampel data citra sel darah putih. Berdasarkan Gambar 2, *dataset* ini terdiri atas 12.392 citra, yaitu 3.120 citra untuk kelas eosinofil, 3.103 citra untuk kelas limfosit, 3.098 citra untuk kelas monosit, dan 3.123 citra untuk kelas neutrofil.

Setelah tahapan pengumpulan data, selanjutnya dilakukan pembagian *dataset* menjadi tiga bagian, yaitu 80% untuk data latih, 10% untuk data validasi, dan 10% untuk data uji. Pemilihan proporsi ini didasarkan pada hasil penelitian yang menunjukkan bahwa pembagian tersebut dapat mencapai akurasi hingga 95,69% [29]. Gambar 3 menunjukkan distribusi pembagian *dataset*.

Gambar 3 dihasilkan melalui pengolahan program yang membagi *dataset* citra sel darah putih ke dalam tiga bagian utama, yaitu data latih (biru), data validasi (oranye), dan data uji (hijau). Sumbu x merepresentasikan kelas pada *dataset*, sedangkan sumbu y merepresentasikan total jumlah data. Pada



Gambar 2. Sampel dataset.



Gambar 3. Distribusi pembagian dataset.

kelas eosinofil terdapat 2.443 data untuk pelatihan, 303 data untuk validasi, dan 295 data untuk pengujian. Kelas limfosit memiliki 2.467 data untuk pelatihan, 313 data untuk validasi, dan 321 data untuk pengujian. Kelas monosit terdiri atas 2.324 data untuk pelatihan, 312 data untuk validasi, dan 304 data untuk pengujian. Terakhir, kelas neutrofil memiliki 2.482 data untuk pelatihan, 287 data untuk validasi, dan 295 data untuk pengujian.

### B. PRAPEMROSESAN

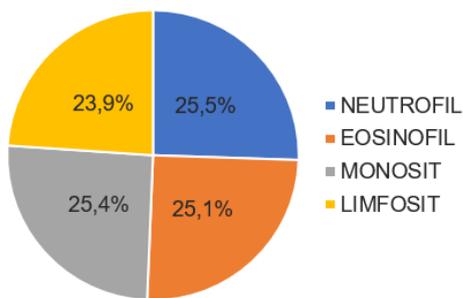
Pada tahapan ini dilakukan prapemrosesan untuk mempersiapkan *dataset* sebelum masuk ke tahapan pemodelan. Langkah pertama adalah melakukan penyeimbangan *dataset* pada jumlah setiap kelas yang tidak seimbang. Gambar 4(a) menunjukkan visualisasi *dataset* citra sel darah putih yang memiliki ukuran setiap kelas tidak seimbang. Untuk mengatasi hal ini, dilakukan proses penyeimbangan data menggunakan teknik *random over sampling*. Teknik ini mengatasi ketidakseimbangan kelas dalam tugas klasifikasi dengan meningkatkan jumlah sampel pada kelas minoritas melalui penggantian sampel yang ada secara acak dengan penggantian [30].

Gambar 4 menggambarkan kondisi sebelum dan setelah proses penyeimbangan data. Setelah proses penyeimbangan dilakukan, hasilnya ditampilkan pada Gambar 4(b), yang menunjukkan bahwa jumlah sampel pada semua kelas telah seimbang.

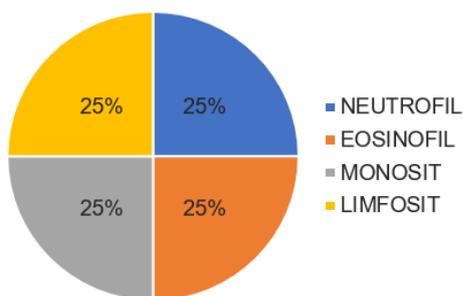
Namun, perlu diperhatikan bahwa penggunaan teknik *oversampling* seperti *random over sampling* dapat memperkenalkan potensi bias ke dalam model. Bias ini muncul karena penggandaan sampel secara acak dengan penggantian dapat meningkatkan kemungkinan *overfitting* terhadap kelas minoritas, sehingga model mungkin belajar pola-pola yang kurang umum [31].

Setelah langkah penyeimbangan *dataset*, dilakukan perubahan ukuran citra untuk memperkecil ukuran sebelum memasuki model CNN. Gambar 5 memvisualisasikan proses perubahan ukuran citra dari ukuran  $320 \times 240$  ke  $120 \times 120$ . Proses ini bertujuan untuk memastikan bahwa masukan citra memiliki dimensi yang konsisten, yang memungkinkan jaringan untuk mengurangi kompleksitas komputasi [32].

Langkah berikutnya adalah segmentasi citra, yang dilakukan untuk memisahkan sel darah putih dari latar

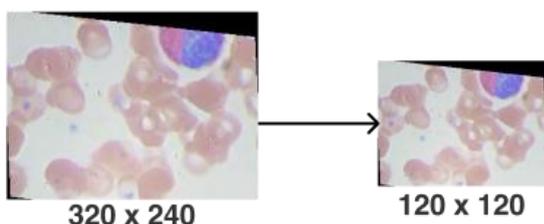


(a)



(b)

Gambar 4. Perbandingan distribusi data, (a) sebelum penyeimbangan, (b) sesudah penyeimbangan.



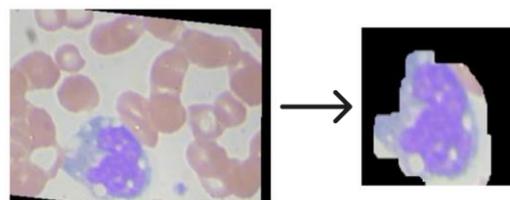
Gambar 5. Proses pengubahan ukuran citra.

belakang menggunakan OpenCV. Proses dimulai dengan *thresholding* untuk menonjolkan objek berdasarkan intensitas piksel, diikuti dengan dilasi untuk memperbesar area objek dan erosi untuk menghilangkan derau. Dilasi ulang diterapkan untuk memperjelas bentuk objek. Selanjutnya, algoritma *Canny edge detection* mendeteksi tepi objek, diikuti oleh deteksi kontur untuk mengidentifikasi batas objek. Kotak pembatas (*bounding box*) dibuat di sekitar kontur untuk menyoroti area yang terdeteksi. Hasil akhirnya adalah citra sel darah putih yang berhasil diekstraksi, seperti terlihat pada Gambar 6.

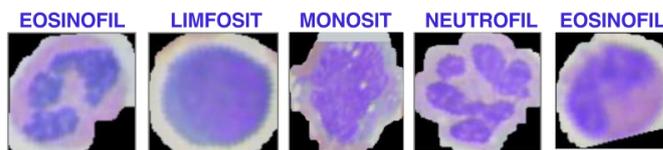
Pada Gambar 6 terlihat area yang telah disegmentasi dari proses sebelumnya, sehingga citra tersebut lebih fokus pada objek yang disegmentasi, memperlihatkan detail yang lebih jelas setelah objek tersebut dipisahkan dari latar belakang. Setelah semua proses segmentasi dilakukan, hasil akhir dari proses segmentasi tersebut akan diterapkan pada keseluruhan *dataset* citra sel darah putih.

Gambar 7 memperlihatkan lima contoh hasil segmentasi citra sel darah putih yang berhasil diidentifikasi dan diklasifikasikan. Setiap citra menunjukkan berbagai jenis sel darah putih, yaitu limfosit, neutrofil, eosinofil, dan monosit, yang telah melalui proses segmentasi. Segmentasi yang dilakukan berhasil memisahkan sel-sel darah putih dari latar belakangnya, sehingga bentuk dan struktur setiap sel terlihat lebih baik. Ini merupakan langkah prapemrosesan sebelum melanjutkan ke proses klasifikasi menggunakan model CNN.

Setelah tahapan penyeimbangan data, perubahan ukuran citra dan segmentasi dilakukan, tahapan berikutnya adalah



Gambar 6. Proses segmentasi citra.



Gambar 7. Sampel lima citra hasil proses segmentasi.

normalisasi nilai piksel. Proses normalisasi merupakan langkah penting untuk mempersiapkan data, memastikan masukan memiliki nilai yang seragam, terutama ukuran citra dengan piksel konsisten. Proses normalisasi nilai piksel dilakukan terhadap *dataset* citra sel darah putih yang akan digunakan dalam proses pelatihan, validasi, dan pengujian model. Normalisasi dilakukan dengan membagi setiap nilai piksel dalam citra dengan 255,0, sehingga mengubah rentang nilai piksel dari [0, 255] menjadi [0, 1].

C. MODEL CNN

Pada tahapan ini, dilakukan proses pembangunan model CNN untuk klasifikasi citra sel darah putih. Arsitektur CNN yang digunakan untuk klasifikasi citra sel darah putih ditampilkan pada Tabel II. Terlihat pada Tabel II bahwa arsitektur CNN yang dirancang pada penelitian ini disusun secara khusus untuk klasifikasi citra sel darah putih dengan mempertimbangkan efisiensi dan efektivitas. Lapisan awal berupa *Conv2D* dengan 16 filter berukuran  $3 \times 3$ , yang bertujuan untuk mengekstraksi fitur dasar seperti tepi dan pola sederhana dari citra berukuran  $120 \times 120$  piksel. Setelah itu, setiap lapisan konvolusi diikuti oleh *max pooling* untuk mengurangi dimensi data, sehingga menurunkan jumlah parameter dan risiko *overfitting*, sambil meningkatkan efisiensi komputasi. Lapisan *separable convolution* digunakan untuk menangkap pola yang lebih kompleks dengan jumlah parameter yang lebih sedikit, sehingga diperoleh keseimbangan akurasi. Untuk menstabilkan pelatihan, lapisan *batch normalization* diterapkan setelah beberapa konvolusi, membantu mempercepat proses pelatihan dan menjaga stabilitas gradien. Selain itu, *dropout* digunakan sebelum lapisan *fully connected* untuk mencegah *overfitting* dengan secara acak menonaktifkan neuron selama pelatihan. Pemilihan elemen-elemen ini mencerminkan keseimbangan antara kompleksitas dan kemampuan generalisasi untuk tugas klasifikasi citra sel darah putih.

Setelah merancang arsitektur, langkah berikutnya adalah melatih model menggunakan data latih dan data validasi dengan model yang telah dibangun sebelumnya. Pada tahap ini, dilakukan penyesuaian terhadap penggunaan *optimizer* pada parameter yang digunakan. Pengembangan metode penelitian dilakukan menggunakan Kaggle Notebooks dengan memanfaatkan GPU T4 yang disediakan oleh platform tersebut untuk mempercepat proses pelatihan model. Perangkat lunak yang digunakan meliputi Python versi 3.8, TensorFlow dan Keras versi 2.8 untuk pengembangan model CNN, serta beberapa *library* pendukung lainnya, seperti NumPy versi 1.21, Pandas versi 1.3, Matplotlib, dan Seaborn untuk analisis dan

TABEL II  
ARSITEKTUR CNN

Lapisan (Tipe)	Bentuk Keluaran	Parameter#
<i>conv2d</i>	(None, 120, 120, 16)	448
<i>max_pooling2d</i>	(None, 60, 60, 16)	0
<i>separable_conv2d</i>	(None, 60, 60, 32)	688
<i>batch_normalization</i>	(None, 60, 60, 32)	128
<i>max_pooling2d_1</i>	(None, 30, 30, 32)	0
<i>separable_conv2d_2</i>	(None, 30, 30, 64)	2400
<i>batch_normalization_1</i>	(None, 30, 30, 64)	256
<i>max_pooling2d_2</i>	(None, 15, 15, 64)	0
<i>separable_conv2d_4</i>	(None, 15, 15, 128)	8896
<i>batch_normalization_2</i>	(None, 15, 15, 128)	512
<i>max_pooling2d_3</i>	(None, 7, 7, 128)	0
<i>dropout</i>	(None, 7, 7, 128)	0
<i>separable_conv2d_6</i>	(None, 7, 7, 256)	34176
<i>batch_normalization_3</i>	(None, 7, 7, 256)	1024
<i>max_pooling2d_4</i>	(None, 3, 3, 256)	0
<i>Flatten</i>	(None, 2304)	0
<i>dense_ (Dense)</i>	(None, 512)	1180160
<i>dense_1 (Dense)</i>	(None, 128)	65664
<i>dense_2 (Dense)</i>	(None, 64)	8256
<i>dense_3 (Dense)</i>	(None, 4)	260
<i>Total params:</i> 1.397.028		
<i>Trainable params:</i> 1.396.068		
<i>Non-trainable params:</i> 960		

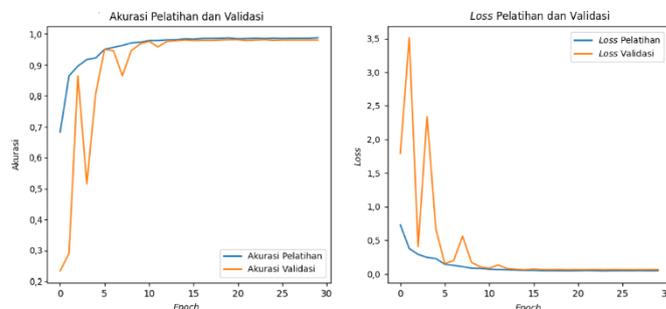
visualisasi data. *Optimizer* yang digunakan dalam eksperimen ini mencakup RMSProp, SGD, dan Adam. Parameter yang diterapkan meliputi *batch size* sebesar 32, yang dipilih untuk menyeimbangkan efisiensi komputasi, stabilitas pembelajaran, dan kemampuan generalisasi model [33]. Laju pembelajaran sebesar 0,0001 digunakan untuk memastikan pembaruan parameter berjalan stabil dan bertahap. Penelitian ini menunjukkan bahwa pemilihan laju pembelajaran tersebut mampu mencegah model gagal konvergen atau mengalami pembelajaran yang terlalu lambat [34]. Jumlah *epoch* yang diterapkan sebanyak 30, yang dianggap cukup untuk mencapai konvergensi tanpa risiko *overfitting*, terutama karena didukung oleh penerapan teknik regularisasi seperti *dropout* dan *batch normalization*.

Berdasarkan Tabel III, dapat disimpulkan bahwa *optimizer* Adam memberikan hasil terbaik dengan akurasi pelatihan sebesar 98,65% dan akurasi validasi sebesar 97,94%, serta nilai *loss* yang paling rendah, baik pada data latih maupun data validasi, yaitu 4,35% dan 6,35%. RMSProp juga memberikan kinerja yang baik, tetapi sedikit di bawah Adam, dengan akurasi pelatihan 98,63% dan akurasi validasi 96,87%, serta *loss* yang lebih tinggi pada validasi dibandingkan Adam. *Optimizer* SGD menunjukkan kinerja yang paling rendah di antara ketiga *optimizer*, dengan akurasi pelatihan 83,46% dan akurasi validasi 87,57%, serta nilai *loss* yang jauh lebih tinggi pada data latih maupun data validasi, yaitu masing-masing 40,39% dan 30,73%. Dari sisi efisiensi komputasi, waktu yang dibutuhkan untuk pelatihan setiap *optimizer* adalah sebagai berikut. SGD membutuhkan waktu 3,82 menit, Adam memerlukan waktu 3,59 menit, dan RMSProp membutuhkan waktu 3,82 menit. Berdasarkan hasil ini, *optimizer* Adam memberikan kinerja terbaik dengan efisiensi komputasi yang unggul dibandingkan *optimizer* lainnya.

Setelah melakukan percobaan penggunaan *optimizer*, diperoleh model CNN optimal pada *optimizer* Adam dengan parameter *batch size* 32, laju pembelajaran 0,0001, dan jumlah *epoch* sebanyak 30. Grafik dari model optimal CNN yang

TABEL III  
HASIL PELATIHAN, VALIDASI, DAN *LOSS* PADA TIAP *OPTIMIZER*

<i>Optimizer</i>	Akurasi (%)	<i>Loss</i> (%)	Akurasi Validasi (%)	<i>Loss</i> Validasi (%)
RMSProp	98,63	5,08	96,87	13,02
SGD	83,46	40,39	87,57	30,73
Adam	98,65	4,35	97,94	6,35



Gambar 8. Hasil grafik akurasi dan *loss*.

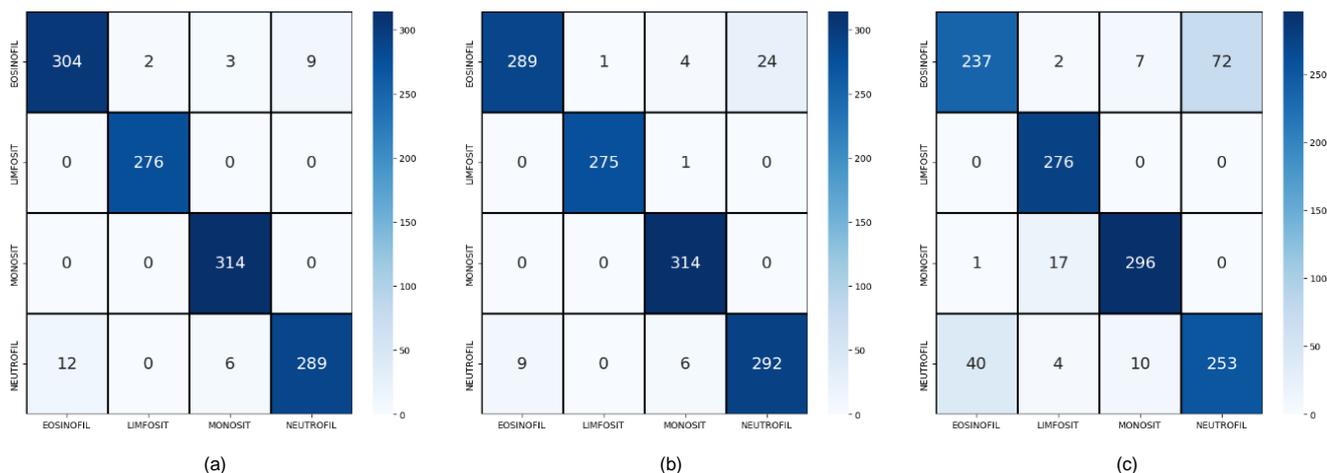
membandingkan metrik akurasi dan *loss* antara data latih dan data validasi ditunjukkan pada Gambar 8.

Gambar 8 diperoleh dari pengolahan program yang menghasilkan grafik akurasi dan *loss* berdasarkan hasil pelatihan model. Berdasarkan Gambar 8, grafik akurasi menunjukkan bahwa akurasi pelatihan (garis biru) meningkat stabil dari 0,3 hingga mendekati 1,0 pada *epoch* ke-20, sementara akurasi validasi (garis oranye) meningkat cepat di awal pelatihan (0-5 *epoch*) sebelum stabil mendekati akurasi pelatihan. Pola ini mengindikasikan generalisasi model yang baik tanpa tanda-tanda *overfitting*. Pada grafik *loss*, *training loss* (garis biru) menurun tajam di awal pelatihan (0-5 *epoch*), mencerminkan efektivitas *optimizer* dalam meminimalkan galat. *Validation loss* (garis oranye) memiliki tren serupa dengan fluktuasi kecil di awal pelatihan (*epoch* 3-5) sebelum stabil, menunjukkan proses penyesuaian bobot terhadap data validasi. Penurunan *loss* yang konsisten hingga mendekati 0 di akhir pelatihan mencerminkan konvergensi model yang baik. Secara keseluruhan, tren ini menunjukkan bahwa model mencapai akurasi tinggi dan *loss* rendah dengan kinerja optimal, didukung oleh efektivitas *optimizer* dalam mengelola pelatihan.

#### D. EVALUASI

Tahap ini dilakukan untuk menilai kinerja model dalam memprediksi data uji yang belum pernah dilihat sebelumnya. Model optimal dari percobaan parameter *tuning* diuji untuk mengukur kemampuan generalisasi. Untuk mendapatkan nilai akurasi, presisi, *recall*, *f1-score*, dan spesifisitas, dilakukan evaluasi *confusion matrix* pada *optimizer* RMSProp, Adam, dan SGD, seperti ditampilkan pada Gambar 9.

Gambar 9 dihasilkan melalui pengolahan program yang memvisualisasikan hasil klasifikasi model dalam bentuk *confusion matrix*. Berdasarkan Gambar 9, terlihat bahwa *optimizer* Adam menunjukkan kinerja terbaik dalam klasifikasi sel darah putih. Adam memiliki akurasi tinggi dengan sedikit kesalahan klasifikasi, terutama pada kelas neutrofil yang kadang salah diklasifikasikan sebagai eosinofil. Hal ini disebabkan oleh penggunaan kombinasi momentum dan pembobotan adaptif yang membuat Adam lebih stabil dan cepat dalam konvergensi, meskipun fitur antarkelas tidak sepenuhnya jelas. RMSProp juga menunjukkan kinerja baik, tetapi sedikit lebih banyak kesalahan pada kelas eosinofil karena berfokus pada adaptasi laju pembelajaran, yang membuatnya stabil, tetapi kurang efektif dalam membedakan



Gambar 9. Confusion matrix, (a) optimizer Adam, (b) optimizer RMSProp, (c) optimizer SGD.

kelas dengan fitur mirip. SGD, meskipun efektif, memiliki kesalahan lebih tinggi, terutama dalam membedakan eosinofil dan neutrofil, karena lebih sederhana dalam memperbarui bobot dan lebih sensitif terhadap pengaturan laju pembelajaran serta kesulitan membedakan kelas dengan fitur visual yang mirip. Setelah mendapatkan nilai TP, TN, FP, dan FN dari *confusion matrix*, nilai akurasi, presisi, *recall*, *f1-score*, dan spesifisitas dihitung menggunakan (10) sampai (14).

Berdasarkan hasil evaluasi model dalam Tabel IV, diketahui bahwa *optimizer* Adam memberikan kinerja terbaik di antara ketiga *optimizer* yang diuji. *Optimizer* Adam menghasilkan akurasi sebesar 97,37%; model dapat mengklasifikasikan sebagian besar citra dengan benar, meskipun akurasi saja tidak cukup untuk menilai kinerja secara menyeluruh. *Recall* 97,43% menunjukkan kemampuan model dalam mendeteksi semua sel darah putih dengan benar, mengurangi risiko *false negative* yang dapat berbahaya dalam dunia medis, seperti kelalaian dalam mendiagnosis kondisi kritis. Presisi 97,42% menandakan bahwa model sangat tepat dalam mengklasifikasikan sel darah putih pada kelas yang benar, mengurangi kemungkinan *false positive* yang dapat menyebabkan *overdiagnosis* dan prosedur medis yang tidak perlu. Dengan *f1-score* sebesar 97,42%, model menunjukkan keseimbangan yang baik antara *recall* dan presisi, yang penting untuk menghindari kesalahan diagnosis seperti *underdiagnosis* atau *overdiagnosis*. Terakhir, nilai spesifisitas sebesar 99,11% menunjukkan kemampuan model dalam menghindari kesalahan klasifikasi negatif, mengurangi kemungkinan *false positive*, yang membantu memperkuat kepercayaan dokter pada hasil diagnosis otomatis.

Selanjutnya, evaluasi menggunakan kurva ROC AUC dilakukan untuk memahami kemampuan model dalam membedakan kelas positif dan negatif melalui area di bawah kurva ROC. Kurva ROC memperlihatkan kinerja model untuk setiap kelas dengan membandingkan *true positive rate* (TPR) dan *false positive rate* (FPR). Hasil kurva ROC dari ketiga *optimizer* ditampilkan pada Gambar 10.

Gambar 10 dihasilkan melalui pengolahan program yang memvisualisasikan metrik evaluasi model menggunakan kurva ROC dan nilai AUC. Berdasarkan Gambar 10, sumbu y menunjukkan nilai TPR, yaitu proporsi positif yang diidentifikasi dengan benar dari semua kelas positif yang sebenarnya. Sumbu x menunjukkan FPR, yaitu proporsi negatif yang salah diidentifikasi dari semua kelas negatif yang sebenarnya. Garis biru, hijau, merah, dan ungu

TABEL IV  
EVALUASI MODEL TIAP OPTIMIZER

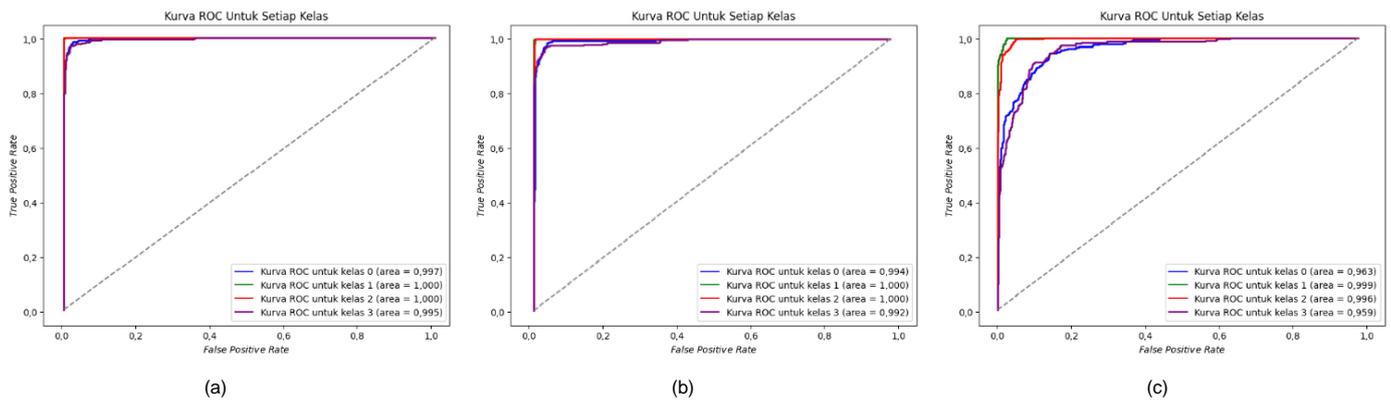
Optimizer	Akurasi	Recall	Presisi	F1-Score	Spesifisitas
RMSProp	0,9630	0,9641	0,9641	0,9637	0,9875
SGD	0,8741	0,8780	0,8749	0,8750	0,9579
Adam	0,9737	0,9743	0,9742	0,9742	0,9911

menggambarkan kinerja model pada berbagai ambang batas, sedangkan garis hitam putus-putus menunjukkan kinerja model acak. Hasil analisis ROC pada empat kelas sel darah putih menunjukkan bahwa *optimizer* Adam menunjukkan kinerja terbaik, dengan rata-rata AUC sebesar 1,00, yang menunjukkan kemampuan model untuk membedakan kelas dengan sangat baik. *Optimizer* RMSProp berada di posisi kedua dengan rata-rata AUC 0,992, menunjukkan kinerja yang sangat baik, tetapi sedikit di bawah Adam. *Optimizer* SGD memiliki rata-rata AUC terendah, yaitu 0,959. Hal ini menunjukkan kinerja yang kurang optimal dalam membedakan kelas-kelas tersebut, dibandingkan dengan Adam dan RMSProp. Oleh karena itu, *optimizer* Adam adalah yang terbaik dalam hal kemampuan membedakan kelas sel darah putih.

Dengan mempertimbangkan aspek praktis, *optimizer* Adam merupakan pilihan terbaik untuk sistem diagnostik otomatis di fasilitas medis dengan infrastruktur yang memadai. Namun, dalam kondisi dengan keterbatasan sumber daya, RMSProp atau bahkan SGD dapat dipertimbangkan sebagai alternatif yang lebih efisien. Untuk aplikasi dunia nyata, pelatihan model lebih lanjut dengan data dari berbagai institusi medis dapat meningkatkan generalisasi dan kepercayaan terhadap sistem ini, sehingga memastikan manfaat maksimal dalam mendukung diagnosis medis yang akurat dan terpercaya.

Berdasarkan hasil penelitian yang diperoleh, klasifikasi citra sel darah putih dengan penggunaan parameter terbaik pada *optimizer* Adam menggunakan algoritma CNN memiliki kinerja yang paling baik jika dibandingkan dengan penelitian-penelitian sebelumnya yang menggunakan metode yang berbeda. Tabel V merupakan perbandingan antara penelitian ini dengan penelitian sebelumnya.

Tabel V menunjukkan bahwa kombinasi CNN dengan *optimizer* Adam memberikan kinerja paling unggul dalam klasifikasi citra sel darah putih dibandingkan dengan metode sebelumnya. Dengan nilai akurasi, *recall*, presisi, *f1-score*, dan AUC, berturut-turut, 97,37%, 97,43%, 97,42%, 97,42%, dan 99%, model ini mengungguli metode lain, termasuk CNN *fully*



Gambar 10. Kurva ROC dan Nilai AUC, (a) Optimizer Adam, (b) Optimizer RMSProp, (c) Optimizer SGD.

TABEL V  
PERBANDINGAN HASIL PENELITIAN

Penelitian	Metode	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)	Spesifisitas (%)	AUC (%)
Penelitian ini	CNN dan <i>optimizer Adam</i>	97,37	97,42	97,43	97,42	99,11	99,00
[8]	CNN <i>fully connected</i>	96,84	96,39	96,26	-	97,35	-
[9]	<i>Meta learning</i> dan <i>color constancy</i>	96,47	95,61	95,61	95,61	-	-
[10]	<i>Decision tree</i>	92,20	-	-	-	-	-
[11]	Con-SVM	85,96	-	-	-	-	-
[12]	KNN	94,30	-	-	-	-	-
[13]	DDRNet	91,98	-	-	96,00	-	-

*connected*, yang dilaporkan dalam penelitian sebelumnya dengan akurasi 96,84% [8], serta metode *meta learning* dan *color constancy* yang mencapai akurasi 96,47% [9]. Kinerja lebih rendah juga terlihat pada metode klasik, seperti *decision tree* dengan akurasi 92,2% [10], Con-SVM dengan akurasi 85,96% [11], dan KNN dengan akurasi 94,3% [12]. Penelitian lain yang menggunakan DDRNet menunjukkan akurasi sebesar 91,98% [13]. Meskipun DDRNet menunjukkan kemampuan yang baik dalam menangani fitur kompleks, kinerja keseluruhannya masih berada di bawah CNN dengan *optimizer Adam* dalam penelitian ini. Hasil ini menunjukkan bahwa pendekatan CNN dengan *optimizer Adam* dapat menghadirkan solusi yang lebih adaptif dalam menganalisis citra medis. Penelitian ini menunjukkan bahwa kombinasi tersebut merupakan pilihan terbaik untuk klasifikasi citra sel darah putih.

## V. KESIMPULAN

Penelitian ini menemukan bahwa di antara ketiga *optimizer* yang diuji—Adam, RMSProp, dan SGD—*optimizer Adam* menunjukkan kinerja terbaik dalam melatih model CNN untuk klasifikasi sel darah putih. Berdasarkan evaluasi menggunakan *confusion matrix*, Adam mencapai akurasi 97,37%, *recall* 97,43%, *presisi* 97,42%, *f1-score* 97,42%, dan *spesifisitas* 99,11%. Nilai-nilai ini menandakan bahwa Adam unggul dalam semua metrik evaluasi. Dibandingkan dengan RMSProp, Adam menunjukkan keunggulan dengan selisih akurasi sebesar 1,07%, *recall* 1,02%, *presisi* 1,01%, *f1-score* 0,95%, dan *spesifisitas* 0,36%. Sementara itu, selisih kinerja Adam dibandingkan dengan SGD jauh lebih signifikan, dengan keunggulan akurasi sebesar 10,96%, *recall* 9,63%, *presisi* 9,93%, *f1-score* 9,92%, dan *spesifisitas* 3,32%. Selain itu, pada kurva ROC-AUC juga terlihat bahwa nilai setiap kelas berada di atas 90%, yang menunjukkan bahwa model dapat membedakan data antarkelas dan termasuk ke dalam kategori *excellent classification*. Berdasarkan hasil ini, disarankan untuk mengimplementasikan model ke dalam sistem berbasis *web*, yang memungkinkan pengunggahan citra dan klasifikasi secara

*real-time* untuk mendukung kebutuhan praktis di bidang medis. Penelitian ini memiliki keterbatasan dalam evaluasi, yaitu perlunya validasi lebih lanjut dengan data eksternal dan pengujian di lingkungan klinis *real-world* untuk memastikan efektivitas model dalam aplikasi klinis yang sesungguhnya.

## KONFLIK KEPENTINGAN

Penulis menyatakan bahwa selama melaksanakan penelitian dan penulisan artikel ilmiah dengan judul “Perbandingan Penggunaan *Optimizer* dalam Klasifikasi Sel Darah Putih Menggunakan *Convolutional Neural Network*” ini, tim penulis tidak memiliki konflik kepentingan dengan pihak mana pun.

## KONTRIBUSI PENULIS

Konseptualisasi, Dede Kurniadi dan Rifky Muhammad Shidiq; metodologi, Dede Kurniadi; perangkat lunak, Rifky Muhammad Shidiq; validasi, Dede Kurniadi dan Asri Mulyani; penulisan—penyusunan draf asli, Dede Kurniadi dan Rifky Muhammad Shidiq; penulisan—peninjauan dan penyuntingan, Dede Kurniadi, Rifky Muhammad Shidiq, dan Asri Mulyani; visualisasi, Rifky Muhammad Shidiq; pengawasan, Dede Kurniadi; pendanaan, Dede Kurniadi.

## UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Institut Teknologi Garut atas dukungan dan pendanaan penelitian ini. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah memberikan kontribusi dan dukungan, baik secara langsung maupun tidak langsung, selama proses penelitian.

## REFERENSI

- [1] B.J. Bain, “Performing a blood count,” dalam *Blood Cells: A Practical Guide*, 6th ed. West Sussex, Inggris: John Wiley & Sons Ltd, 2022, ch. 2, hal. 17–63.
- [2] W. King, K. Toler, dan J. Woodell-May, “Role of white blood cells in blood- and bone marrow-based autologous therapies,” *Biomed Res. Int.*, vol. 2018, no. 1, hal. 1–8, Jul. 2018, doi: 10.1155/2018/6510842.

- [3] N. Dong, M. Zhai, J. Chang, dan C. Wu, "A self-adaptive approach for white blood cell classification towards point-of-care testing," *Appl. Soft Comput.*, vol. 111, hal. 1–13, Nov. 2021, doi: 10.1016/j.asoc.2021.107709.
- [4] M.A. Ali, F. Dornaika, dan I. Arganda-Carreras, "White blood cell classification: Convolutional neural network (CNN) and vision transformer (ViT) under medical microscope," *Algorithms*, vol. 16, no. 11, hal. 1–17, Nov. 2023, doi: 10.3390/a16110525.
- [5] T.A. Sadoon dan M.H. Ali, "An overview of medical images classification based on CNN," *Int. J. Curr. Eng. Technol.*, vol. 10, no. 6, hal. 900–905, Nov./Des. 2020, doi: 10.14741/ijcet/v.10.6.1.
- [6] E.M. Dogo, O.J. Afolabi, dan B. Twala, "On the relative impact of optimizers on convolutional neural networks with varying depth and width for image classification," *Appl. Sci.*, vol. 12, no. 23, hal. 1–36, Des. 2022, doi: 10.3390/app122311976.
- [7] P. Mooney, "Blood cell images." Kaggle. Tanggal akses: 13-Mar-2024. [Online]. Tersedia: <https://www.kaggle.com/datasets/paultimothymooney/blood-cells>
- [8] K.G. Kannan dkk. "Classification of WBC cell classification using fully connected convolution neural network," *J. Phys., Conf. Ser.*, vol. 2466, No. 1, Apr. 2023, Art. No. 012033, doi: 10.1088/1742-6596/2466/1/012033.
- [9] E. Rivas-Posada, M.I. Chacon-Murguia, J.A. Ramirez-Quintana, dan C. Arzate-Quintana, "Classification of leukocytes using meta-learning and color constancy methods," *J. Ilmiah Tek. Elekt. Komput. Inform.*, vol. 8, no. 4, hal. 486–499, Des. 2022, doi: 10.26555/jiteki.v8i4.25192.
- [10] C. Panjaitan, Y. Panjaitan, D. Sitanggang, dan S.W. Tarigan, "Image processing for detection of dengue virus," *J. Sis. Inf. Ilmu Komput. Prima*, vol. 7, no. 2, hal. 26–34, Feb. 2024, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v7i2.4799.
- [11] A. Ekiz, K. Kaplan, dan H.M. Ertuñç, "Classification of white blood cells using CNN and Con-SVM," dalam *2021 29th Signal Process. Commun. Appl. Conf. (SIU)*, 2021, hal. 1–4, doi: 10.1109/SIU53274.2021.9477962.
- [12] Z.E. Fitri, L.N.Y. Syahputri, dan A.M.N. Imron, "Classification of white blood cell abnormalities for early detection of myeloproliferative neoplasms syndrome based on k-nearest neighbor," *Sci. J. Inform.*, vol. 7, no. 1, hal. 136–142, Mei 2020, doi: 10.15294/sji.v7i1.24372.
- [13] M. Jawahar, L.J. Anbarasi, S. Narayanan, dan A.H. Gandomi, "An attention-based deep learning for acute lymphoblastic leukemia classification," *Sci. Rep.*, vol. 14, hal. 1–20, Jul. 2024, doi: 10.1038/s41598-024-67826-9.
- [14] Y. Shao dkk., "An improved BGE-Adam optimization algorithm based on entropy weighting and adaptive gradient strategy," *Symmetry*, vol. 16, no. 5, hal. 1–16, Mei 2024, doi: 10.3390/sym16050623.
- [15] R. Elshamy, O. Abu-Elnasr, M. Elhoseny, dan S. Elmougy, "Improving the efficiency of RMSProp optimizer by utilizing Nesterov in deep learning," *Sci. Rep.*, vol. 13, hal. 1–16, Mei 2023, doi: 10.1038/s41598-023-35663-x.
- [16] Y. Tian, Y. Zhang, dan H. Zhang, "Recent advances in stochastic gradient descent in deep learning," *Mathematics*, vol. 11, no. 3, hal. 1–23, Feb. 2023, doi: 10.3390/math11030682.
- [17] L. Alzubaidi dkk., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, hal. 1–74, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
- [18] A. Dosovitskiy dkk., "An image is worth 16x16 words: Transformers for image recognition at scale," dalam *9th Int. Conf. Learn. Represent.*, 2021, hal. 1–22.
- [19] I. Adjabi, A. Ouahabi, A. Benzaoui, dan A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, hal. 1–52, Agu. 2020, doi: 10.3390/electronics9081188.
- [20] W. Di, A. Bhardwaj, dan J. Wei, *Deep Learning Essentials*. Birmingham, Inggris: Packt, 2018.
- [21] Y. Wang, Z. Xiao, dan G. Cao, "A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis," *J. Vibroengineering*, vol. 24, no. 4, hal. 666–678, Jun. 2022, doi: 10.21595/jve.2022.22271.
- [22] Q. Tong, G. Liang, dan J. Bi, "Calibrating the adaptive learning rate to improve convergence of ADAM," *Neurocomputing*, vol. 481, hal. 333–356, Apr. 2022, doi: 10.1016/j.neucom.2022.01.014.
- [23] S. Nagendram dkk., "Stochastic gradient descent optimisation for convolutional neural network for medical image segmentation," *Open Life Sci.*, vol. 18, no. 1, hal. 1–15, Agu. 2023, doi: 10.1515/biol-2022-0665.
- [24] Z. Wang, E. Wang, dan Y. Zhu, "Image segmentation evaluation: A survey of methods," *Artif. Intell. Rev.*, vol. 53, hal. 5637–5674, Des. 2020, doi: 10.1007/s10462-020-09830-9.
- [25] Y. Said, A.A. Alsheikhy, T. Shawly, dan H. Lahza, "Medical images segmentation for lung cancer diagnosis based on deep learning architectures," *Diagnostics*, vol. 13, no. 3, hal. 1–15, Feb. 2023, doi: 10.3390/diagnostics13030546.
- [26] V. Chaudhary, P.K. Buttar, dan M.K. Sachan, "Satellite imagery analysis for road segmentation using U-Net architecture," *J. Supercomput.*, vol. 78, hal. 12710–12725, Jul. 2022, doi: 10.1007/s11227-022-04379-6.
- [27] I. Markoulidakis dkk., "Multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, hal. 1–22, Des. 2021, doi: 10.3390/technologies9040081.
- [28] Ş.K. Çorbacıoğlu dan G. Akseil, "Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value," *Turkish J. Emerg. Med.*, vol. 23, no. 4, hal. 195–198, Okt. –Des. 2023, doi: 10.4103/tjem.tjem\_182\_23.
- [29] V.R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min., ASA Data Sci. J.*, vol. 15, no. 4, hal. 531–538, Agu. 2022, doi: 10.1002/sam.11583.
- [30] T. Wongvorachan, S. He, dan O. Bulut, "A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining," *Information*, vol. 14, no. 1, hal. 1–15, Jan. 2023, doi: 10.3390/info14010054.
- [31] C. Yang dkk., "Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data," *J. Big Data*, vol. 11, hal. 1–18, Jan. 2024, doi: 10.1186/s40537-023-00857-7.
- [32] A. Semma dkk., "Writer identification: The effect of image resizing on CNN performance," dalam *6th Int. Conf. Smart City Appl.*, 2021, hal. 501–507, doi: 10.5194/isprs-archives-XLVI-4-W5-2021-501-2021.
- [33] T.A.A.H. Kusuma, K. Usman, dan S. Saidah, "People counting for public transportations using you only look once method," *J. Tek. Inform.*, vol. 2, no. 1, hal. 57–66, Jun. 2021, doi: 10.20884/1.jutif.2021.2.2.77.
- [34] T.S. Nabila dan A. Salam, "Classification of brain tumors by using a hybrid CNN-SVM model," *J. Appl. Inform. Comput.*, vol. 8, no. 2, hal. 241–247, Des. 2024, doi: 10.30871/jaic.v8i2.8277.