

Penerapan Arsitektur EfficientNet-B0 Pada Klasifikasi Leukimia Tipe Acute Lymphoblastik Leukimia

Application of the EfficientNet-B0 Architecture in the Classification of Acute Lymphoblastic Leukemia

Alfataniah Nur Fajrina¹, Zein Hanni Pradana², Sevia Indah Purnama³, Shinta Romadhona⁴

^{1,2,4} Program Studi S1 Teknik Telekomunikasi, ³ Program Studi S1 Teknik Biomedis,

Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto

^{1,2,3,4} Jl. D.I Pandjaitan No. 128, Purwokerto, Indonesia

Informasi Makalah

Dikirim, 31 Mei 2024
Diterima, 25 Juni 2024
Diterbitkan, 30 Juni 2024

Kata Kunci:

Acute Lymphoblastik
Leukemia
Convolutional Neural Network
EfficientNetB0
Hyperparameter

Keyword:

Acute Lymphoblastik
Leukemia
Convolutional Neural Network
EfficientNetB0
Hyperparameter

INTISARI

Leukimia merupakan jenis kanker darah yang keganasannya dapat berkembang dengan cepat. Penundaan penanganan akan berakibat fatal dalam waktu beberapa bulan. Proses diagnosa dengan cepat dilakukan dengan cara memanfaatkan pemrosesan citra. Sebuah sistem dirancang agar mampu mengklasifikasikan penyakit leukimia tipe *Acute Lymphoblastic Leukemia* (ALL) ke dalam 4 kelas, yaitu: *Benign*, *Early*, (Pre) *Precursor*, dan *Pro* (Progenitor) dengan memanfaatkan salah satu arsitektur dari Convolutional Neural Network (CNN) yaitu *EfficientNet-B0*. Skenario pengujian dilakukan terhadap hyper parameter pada arsitektur *EfficientNet-B0* melalui epoch (20, 30 dan 50) dengan learning rate (0.0001, 0.0003, 0.001, 0.003) dan optimizer jenis Adam dan RMSProp. Hasilnya adalah nilai performa akurasi pada data train mencapai 97,84% dan pada data test sebesar 98,48%.

ABSTRACT

Leukemia is a type of blood cancer that can develop rapidly. Delaying treatment can be fatal within a few months. Rapid diagnosis is conducted through image processing. A system has been designed to classify Acute Lymphoblastic Leukemia (ALL) into four categories: *Benign*, *Early*, (Pre) *Precursor*, and *Pro* (Progenitor) by utilizing one of the Convolutional Neural Network (CNN) architectures, namely *EfficientNet-B0*. The testing scenario involves adjusting hyperparameters on the *EfficientNet-B0* architecture through epochs (20, 30, and 50) with learning rates (0.0001, 0.0003, 0.001, 0.003) and using Adam and RMSProp optimizers. The results indicate that the accuracy performance on the training data reaches 97.84% and on the test data 98.48%.

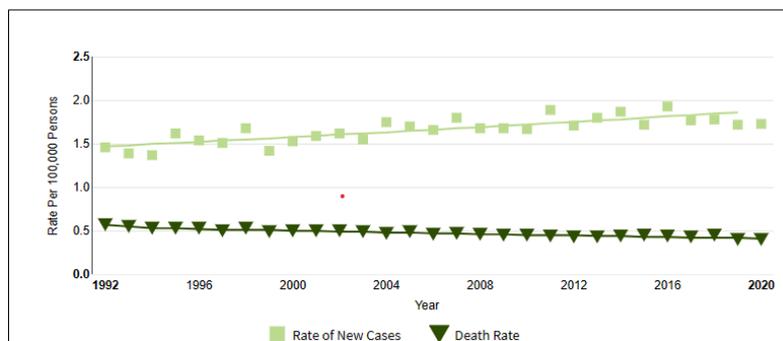
Korespondensi Penulis:

Zein Hanni Pradana
Program Studi S1 Teknik Telekomunikasi, Fakultas Teknik Telekomunikasi dan Elektro,
Institut Teknologi Telkom Purwokerto
Jl. D.I Pandjaitan No. 128, Purwokerto, Indonesia
Email : zeindana@ittelkom-pwt.ac.id

1. PENDAHULUAN

Leukemia Limfositik Akut adalah jenis kanker yang disebabkan oleh limfosit imatur di sumsum tulang [1]. Leukemia adalah kanker yang terjadi pada sel darah manusia, di mana tubuh memproduksi sel darah abnormal dalam jumlah besar [2]. Penyakit ini menyebar dengan cepat dan memerlukan perawatan segera. Akibatnya, darah tidak dapat menjalankan fungsi normalnya, membuat sistem kekebalan tubuh rentan [2][3].

Leukemia Limfositik Akut, juga dikenal sebagai leukemia limfoblastik akut, menunjukkan bahwa leukemia dapat berkembang dengan cepat dan menjadi fatal bagi kondisi pasien dalam beberapa bulan [4]. Diagnosis leukemia biasanya bergantung pada pemeriksaan sumsum tulang yang dilakukan oleh spesialis di bidangnya. Oleh karena itu, klasifikasi sel darah putih memainkan peran penting dalam membantu diagnosis leukemia [5].



Gambar 1 Diagram kasus Leukemia di US [5]

Dilihat pada Gambar 1 data diambil dari SEER (*Surveillance, Epidemiology, and End Results*). Berdasarkan data statistik di atas, kasus leukemia di AS mencapai hampir 150.000 kasus per tahun pada tahun 1992. Jumlah ini menurun secara bertahap hingga tahun 1994, tetapi terjadi peningkatan kembali pada tahun 1995, melampaui angka 150.000 kasus per tahun. Secara keseluruhan, kasus leukemia di AS dari tahun 1992 hingga 2020 mengalami fluktuasi acak setiap tahun. Namun, dari data statistik tersebut, dapat disimpulkan bahwa jumlah kasus penyakit ini cenderung meningkat, meskipun tidak signifikan.

Dalam literatur, banyak peneliti telah membahas klasifikasi leukemia menggunakan metode CNN. Beberapa peneliti telah menggunakan *Convolutional Neural Networks* untuk mengklasifikasikan sel darah normal dan abnormal serta mengusulkan berbagai metode augmentasi data sederhana karena kurangnya dataset [5]. Studi lain menerapkan teknik augmentasi data sederhana dalam mengklasifikasikan berbagai jenis kanker sel darah putih, termasuk Leukemia Limfoblastik Akut dan Leukemia Mieloid Akut, menggunakan metode CNN [6]. Selain itu, para peneliti telah menyarankan penggunaan arsitektur CNN seperti ResNet, VGG16, dan MobileNet untuk klasifikasi leukosit otomatis, mencapai akurasi sebesar 88,5% [7]. Upaya untuk meningkatkan akurasi juga telah dilakukan menggunakan metode CNN dan memanfaatkan arsitektur kuat seperti AlexNet, GoogleNet, dan ResNet, yang telah menunjukkan hasil akurasi tinggi [8] [9].

Penelitian tentang klasifikasi penyakit sel darah putih menggunakan metode CNN memiliki dampak dan manfaat yang signifikan di bidang kedokteran dan perawatan kesehatan. Ini dapat membantu profesional medis dalam mendiagnosis leukemia. CNN adalah salah satu metode pembelajaran mendalam yang telah banyak diterapkan pada data gambar. CNN beroperasi serupa dengan otak manusia, di mana komputer diberikan data gambar untuk belajar, dilatih untuk mengenali setiap elemen visual dalam gambar, dan memahami setiap pola gambar, sehingga komputer dapat mengklasifikasikan gambar tersebut. Meskipun CNN umumnya menghasilkan tingkat akurasi yang lebih tinggi, penting untuk dicatat bahwa tingkat akurasi aktual dapat bervariasi tergantung pada faktor-faktor seperti ukuran dataset, kualitas data, dan pengaturan arsitektur [1][10].

Berdasarkan deskripsi di atas, penulis merancang sistem yang dapat membantu dalam mendiagnosis Leukemia Limfoblastik Akut dengan cepat menggunakan data gambar dengan CNN. Sistem yang diusulkan memiliki kemampuan untuk mengklasifikasikan gambar ke dalam empat kelas: jinak, awal, prekursor, dan progenitor, dengan memanfaatkan dua arsitektur CNN, yaitu MobileNetV3-Large dan EfficientNet-B0, serta dua jenis optimizer, Adam dan RMSProp. Dengan menggabungkan teknologi-teknologi ini, hasil yang diharapkan dapat membantu profesional medis dalam mendiagnosis leukemia.

Dalam penyelidikan ini, pendekatan yang diadopsi mampu mengklasifikasikan Leukemia Limfoblastik Akut (ALL) menjadi tipe L1, L2, dan L3, serta tipe normal. Dataset mencakup gambar dari semua subtype, termasuk masing-masing 100 gambar L1 dan L2, 30 gambar L3 karena kelangkaannya, dan 100 gambar sumsum tulang normal. Metode yang diusulkan menunjukkan kinerja yang memuaskan dengan mengambil gambar sumsum tulang sebagai input, melakukan segmentasi, dan mengkategorikannya sebagai normal atau masuk ke dalam subtype L1, L2, dan L3. Perbandingan kinerja dengan akurasi Naive Bayesian sebesar 78,34% dan akurasi KNN sebesar 80,42% dilakukan. Namun, batasan yang diidentifikasi dalam pendekatan ini terletak pada kebutuhan untuk meningkatkan akurasi dalam segmentasi sel yang tumpang tindih [1] [11].

Dalam studi berjudul "*Automatic Classification of Leukocytes Using Deep Learning Neural Network*," para peneliti mengusulkan sistem untuk klasifikasi leukosit otomatis. Sistem ini menggunakan arsitektur CNN

untuk memprediksi kategori sel dari gambar mikroskopis. Sekitar 2000 gambar mikroskopis dari 7 jenis sel darah putih dari Rumah Sakit Menara yang berafiliasi dengan Sekolah Kedokteran Universitas Nanjing digunakan. Arsitektur CNN seperti ResNet50, VGG19, dan MobileNet diusulkan, mencapai akurasi sebesar 88,5% [12] [10].

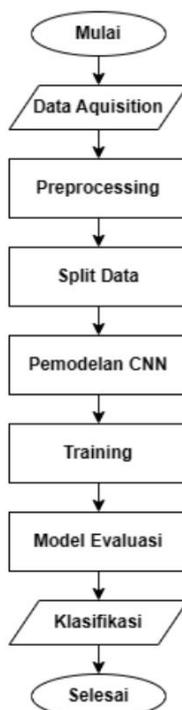
Dalam studi berjudul "*Automatic Classification of Leukocytes Using Deep Learning Neural Network*," penelitian ini menyajikan sistem untuk klasifikasi leukosit otomatis. Sistem ini mengambil gambar mikroskopis sebagai input dan menggunakan arsitektur CNN untuk memprediksi kategorinya. Kemudian, sistem menggabungkan hasil dari berbagai model dan mengklasifikasikan sel ke dalam kelas yang sesuai. Dalam studi ini, sekitar 2000 gambar mikroskopis dari 7 jenis sel darah putih digunakan, yang disediakan oleh Rumah Sakit Menara yang berafiliasi dengan Sekolah Kedokteran Universitas Nanjing. Studi ini mengusulkan penggunaan arsitektur CNN seperti ResNet50, VGG19, dan MobileNet, yang dapat mencapai akurasi sebesar 88,5% [7].

Dalam studi berjudul "*Classification of White Blood Cell Cancer (Acute Lymphoblastic Leukemia) using Convolutional Neural Network (CNN)*," para peneliti menggunakan arsitektur CNN, khususnya ResNet dan VGG16. Hasil pelatihan arsitektur VGG16 dengan optimizer Adam dan 100 epoch menunjukkan akurasi sebesar 93,80%, dengan akurasi pengujian mencapai 87,00%. Sementara itu, arsitektur ResNet mencapai akurasi pelatihan sebesar 81,81%, dengan akurasi pengujian menggunakan optimizer RMSProp dan 80 epoch. Studi ini menyarankan eksplorasi lebih lanjut untuk meningkatkan akurasi dengan memodifikasi model arsitektur, menyesuaikan hiperparameter, menggunakan jenis arsitektur yang berbeda, dan menerapkan teknik augmentasi. Selain itu, penelitian ini mengklasifikasikan data ke dalam 4 kelas, berbeda dengan studi sebelumnya yang hanya membedakan antara sel normal dan sel kanker [2].

2. METODE PENELITIAN

2.1. Alur Penelitian

Dalam perancangan sistem ini, disediakan gambaran umum tentang sistem yang akan dikembangkan.



Gambar 2 Alur Penelitian

Klasifikasi penyakit *Acute Lymphoblastic Leukemia (ALL)* menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur EfficientNetB0 dan MobileNetV3Large untuk mengklasifikasikan 4 kelas, yaitu Benign, Early, (Pre) Precursor, dan (Pro) Progenitor. Perancangan penelitian dapat dilihat pada Gambar 2 dimulai dengan data akusisi dengan 80% data digunakan sebagai training set, 10% sebagai validation set, dan 10% sebagai testing set. Selanjutnya data akan diolah melalui tahap *preprocessing* teknik yang digunakan mencakup perubahan ukuran (*resizing*), yang mengubah ukuran gambar, dan pengubahan skala (*rescaling*) ke ukuran 224 x 224. Tahap berikutnya adalah mengoptimalkan *hyperparameter* dan melatih model. Setelah model dilatih, tahap berikutnya adalah menyimpan model yang telah dilatih melalui tahap penyimpanan model.

2.1.1. Akuisisi Data

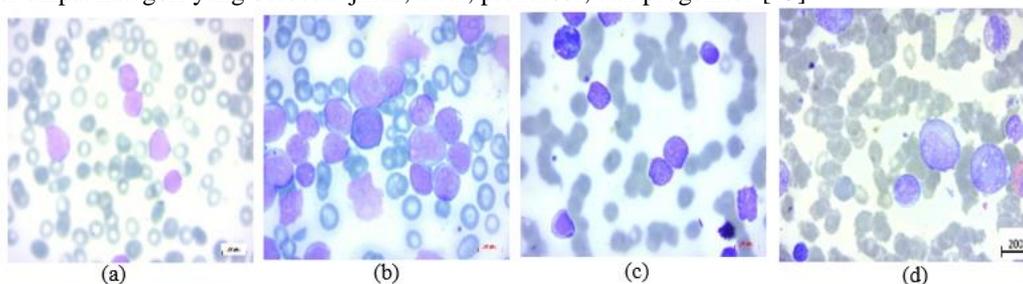
Pada tahap akuisisi data, gambar atau dataset yang akan digunakan untuk melatih model dikumpulkan. Dalam penelitian ini, dataset yang digunakan diperoleh dari sumber yang tersedia, yaitu "*Acute Lymphoblastic Leukemia (ALL) image dataset*" yang disediakan oleh Mehrad Aria dan Mustafa Ghaderzadeh di Kaggle pada tahun 2021. Total dataset terdiri dari 3256 data, dengan rincian 504 data termasuk dalam kategori benign, 985 data dalam kategori *early*, 963 data dalam kategori *pre*, dan 804 data dalam kategori *pro*. Setelah akuisisi data, langkah berikutnya adalah membagi dataset menjadi tiga subset yang berbeda, yaitu training set, validation set, dan *testing set*. Pendekatan yang diterapkan dalam membagi dataset adalah dengan rasio 80-10-10, di mana 80% data digunakan sebagai training set, 10% sebagai validation set, dan 10% sebagai *testing set*. Pembagian dataset yang proporsional ini dilakukan untuk meningkatkan efisiensi proses pelatihan dan evaluasi model, serta memastikan bahwa model yang dihasilkan memiliki kemampuan yang baik dalam menggeneralisasi data baru yang belum pernah dilihat sebelumnya. Dalam konteks penelitian ini, training set terdiri dari 328 data, validation set terdiri dari 2,604 data, dan testing set terdiri dari 324 data.

2.1.2. Pre-Processing

Pada tahap pra-pemrosesan gambar, dilakukan modifikasi pada gambar untuk memastikan bahwa model pelatihan dapat berfungsi lebih baik. Teknik pra-pemrosesan yang digunakan mencakup perubahan ukuran (*resizing*), yang mengubah ukuran gambar, dan pengubahan skala (*rescaling*), untuk mengurangi nilai piksel agar lebih mudah diproses oleh model. Langkah-langkah pra-pemrosesan ini bertujuan untuk memastikan bahwa gambar memiliki kualitas optimal sehingga model pelatihan mencapai hasil terbaik. Salah satu langkah yang diambil adalah menyesuaikan ukuran gambar menjadi 224×224 piksel. Selain itu, augmentasi gambar dilakukan menggunakan kerangka kerja keras untuk mengurangi risiko *overfitting*. *Overfitting* terjadi ketika model hanya mengenali pola dalam data pelatihan, sehingga sulit mengklasifikasikan gambar baru. Augmentasi data dilakukan dengan berbagai cara seperti *flipping*, *cropping*, rotasi, penyesuaian kecerahan, penyesuaian kontras, penyesuaian warna, penyesuaian saturasi, translasi, injeksi noise, penajaman, shear, dan lainnya. Dalam penelitian ini, teknik augmentasi yang diterapkan meliputi *cropping*, rotasi, *flipping*, dan *shifting* [15].

2.1.3. Split Data

Leukemia adalah bentuk kanker yang mempengaruhi sel darah putih, yang memainkan peran penting dalam melawan infeksi di dalam tubuh. Penyakit ini muncul sebagai keganasan dalam sel darah manusia yang berasal dari sumsum tulang [13]. Kondisi ini merupakan salah satu kelainan darah yang mempengaruhi sel darah putih manusia. Pada dasarnya, ketika leukemia berkembang, tubuh menghasilkan sel darah abnormal dalam jumlah berlebihan [14]. Sel darah abnormal ini berkembang biak dengan cepat dalam leukemia dan memerlukan intervensi medis segera. Tujuan dari studi ini adalah untuk mengklasifikasikan leukemia ALL menjadi empat kategori yang berbeda: jinak, awal, prekursor, dan progenitor [15].



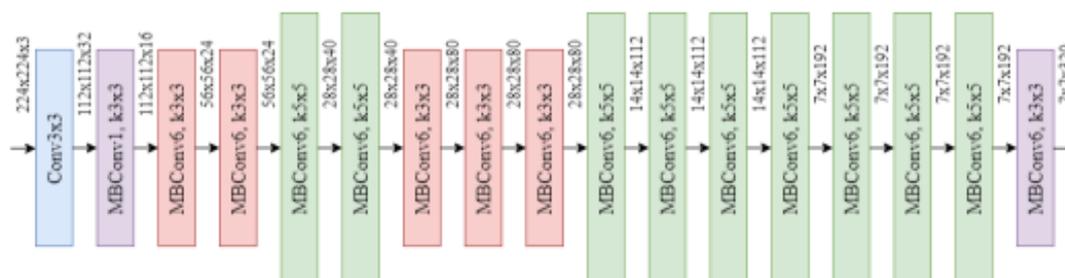
Gambar 2. Contoh gambar penyakit Leukemia Limfoblastik Akut yang terdiri dari 4 kelas: (a) Jinak, (b) Awal, (c) Pre, (d) Pro.

2.1.4. Model Deep Learning

Setelah proses pelatihan selesai, hasilnya adalah model *deep learning* yang mampu mengidentifikasi fitur dalam gambar leukemia limfoblastik akut. Model ini digunakan untuk klasifikasi gambar berdasarkan kategori masing-masing. Selanjutnya, kinerja model *deep learning* dievaluasi melalui pengujian untuk menilai efektivitasnya dalam mengklasifikasikan gambar. Penilaian dilakukan menggunakan berbagai metrik kinerja, termasuk *confusion matrix*, akurasi, *loss*, *precision*, *recall*, dan *F1-Score*. *Confusion matrix* digunakan untuk menilai akurasi prediksi model dengan menghitung prediksi yang benar. Akurasi menilai seberapa sering model membuat prediksi yang benar, sedangkan *loss* menilai seberapa efektif model mengurangi ketidaksesuaian antara prediksi. *Precision* mengukur akurasi model dalam memprediksi kelas positif, sedangkan *recall* mengevaluasi kemampuan model mendeteksi kelas positif. *F1-Score* memberikan evaluasi yang seimbang dari keseluruhan kinerja model. Setelah *hyperparameter* optimal dari model ditentukan, evaluasi dilakukan menggunakan metrik ini untuk menilai kinerja keseluruhan model.

2.2. Arsitektur EfficientNetB0

EfficientNet-B0 adalah salah satu varian dari arsitektur EfficientNet yang dikembangkan oleh tim riset Google AI dengan tujuan menciptakan model jaringan saraf yang lebih efisien secara komputasional namun tetap mempertahankan kinerja tinggi. Arsitektur EfficientNet didasarkan pada konsep *smart scaling*, yang melibatkan peningkatan kedalaman, lebar, dan resolusi model secara proporsional untuk mengoptimalkan kinerja dan efisiensi.



Gambar 3. Arsitektur EfficientNet-B0

Model EfficientNet-B0 bertindak sebagai model dasar dalam seri EfficientNet, dengan jumlah parameter paling sedikit. Meskipun sederhana, EfficientNet-B0 tetap sangat efektif dalam upaya klasifikasi gambar. Arsitektur ini terdiri dari beberapa blok berulang yang terdiri dari *Depthwise Separable convolutions*, disertai dengan Swish non linearity yang berfungsi sebagai fungsi aktivasi [16].

Teknik Depthwise Separable convolution memperkenalkan pendekatan baru dengan membagi proses konvolusi standar menjadi dua tahap: *depthwise convolution* dan *pointwise convolution*. Pada tahap *depthwise*, saluran input individual diproses secara terpisah menggunakan filter yang hanya berisi satu saluran. Hal ini menghasilkan peta fitur dengan banyak saluran, meskipun dengan intensitas komputasi yang berkurang. Selanjutnya, pada tahap *pointwise*, konvolusi 1x1 digunakan untuk menggabungkan saluran-saluran ini, menghasilkan output akhir. Teknik inovatif ini secara efektif mengurangi jumlah parameter yang dibutuhkan untuk model sambil mempertahankan standar kinerja. Selain itu, dalam konteks EfficientNet-B0, operasi penskalaan dilakukan pada kedalaman, lebar, dan resolusi model, dipandu oleh parameter penskalaan yang telah ditentukan sebelumnya. Strategi penskalaan ini memastikan adaptabilitas model di berbagai tugas dan ukuran dataset, mengurangi risiko *overfitting* atau *underfitting* [17].

2.3. Hyperparameter

Hyperparameter adalah parameter yang nilainya ditentukan sebelum proses pelatihan dimulai. Berikut adalah *hyperparameter* yang terdiri dari *optimizer*, *learning rate*, *epoch*, dan *batch size*.

2.3.1. Optimizer

Optimizer adalah algoritma yang digunakan dalam proses pelatihan model *machine learning* dan *deep learning* untuk menemukan dan mengoptimalkan nilai bobot dan bias dalam jaringan saraf. Tujuannya adalah untuk meminimalkan fungsi sehingga model dapat belajar dari data pelatihan secara efektif dan menghasilkan prediksi yang akurat pada data baru [18]. *Optimizer* yang digunakan dalam penelitian ini adalah:

- Adam:** Adam adalah algoritma optimasi yang digunakan dalam pelatihan jaringan saraf tiruan. Tujuan utama Adam adalah memperbarui bobot jaringan secara efisien dan cepat sehingga jaringan dapat belajar secara efektif dari data pelatihan [19].
- RMSProp:** *Root Mean Square Propagation* atau RMSProp adalah varian khusus dari Adagrad yang dikembangkan oleh Profesor Geoffrey Hinton untuk jaringan sarafnya. RMSProp menggunakan besaran gradien terbaru untuk menormalkan gradien, mempertahankan rata-rata bergerak dari *root mean square gradien*, sehingga dinamakan RMS [20].

2.3.2. Learning Rate

Learning rate adalah *hyperparameter* penting dalam pelatihan jaringan saraf tiruan. Ini mengontrol seberapa besar bobot jaringan disesuaikan selama setiap iterasi pelatihan. Jika learning rate terlalu kecil, pelatihan mungkin berjalan lambat karena perubahan pada bobot sangat kecil. Sebaliknya, jika learning rate terlalu besar, pelatihan bisa menjadi tidak stabil karena perubahan yang terlalu besar yang dapat melewati minimum global dari fungsi *loss*. Oleh karena itu, memilih learning rate yang tepat sangat penting untuk memastikan pelatihan jaringan berjalan lancar dan model dapat mencapai hasil yang baik [14].

2.3.3. Epoch

Jumlah *epoch* menentukan berapa kali seluruh dataset pelatihan akan ditampilkan ke model selama proses pelatihan. Jumlah *epoch* yang digunakan dalam pelatihan model adalah salah satu *hyperparameter* yang harus ditentukan. Jumlah *epoch* yang sesuai dapat bervariasi tergantung pada kompleksitas tugas dan ukuran dataset. Terlalu sedikit *epoch* mungkin tidak memberi model cukup kesempatan untuk belajar dengan baik, sementara terlalu banyak *epoch* dapat menyebabkan *overfitting* [14].

2.3.4. Batch Size

Dalam pelatihan model, data pelatihan terdiri dari kumpulan besar sampel. *Batch size* digunakan untuk membagi dataset menjadi bagian-bagian yang lebih kecil yang disebut "*batch*" atau kelompok, sehingga tidak seluruh dataset digunakan sekaligus. Dengan menggunakan *batch size*, proses pelatihan menjadi lebih efisien. *Batch size* menentukan jumlah sampel data yang digunakan dalam setiap iterasi (*epoch*) saat melatih model.

2.4. Skenario Pengujian

Skenario *hyperparameter* adalah proses mencari nilai terbaik untuk komponen penting dalam model *deep learning*. Proses ini melibatkan tiga skenario utama: pengujian *optimizer*, pengujian *learning rate*, dan pengujian jumlah *epoch*.

- Pengujian *Optimizer*:** Membandingkan dua jenis optimizer, Adam dan RMSProp, untuk menentukan mana yang menghasilkan hasil terbaik.
- Pengujian *Learning Rate*:** Menguji empat *learning rate* berbeda (0.0001, 0.0003, 0.001, dan 0.003) untuk menemukan nilai optimal.
- Pengujian *Epoch*:** Melakukan pengujian dengan tiga nilai *epoch* berbeda (20, 30, dan 50) untuk mengidentifikasi mana yang menghasilkan model dengan akurasi tinggi dan *loss* rendah.

Tujuan dari pengujian ini adalah menemukan kombinasi *optimizer*, *learning rate*, dan jumlah *epoch* yang dapat menghasilkan model *deep learning* dengan akurasi tinggi dan *loss* minimal.

2.5. Confusion Matrix

Confusion matrix adalah teknik yang digunakan untuk mengevaluasi kinerja metode klasifikasi. Ini memberikan informasi tentang seberapa baik hasil klasifikasi dari sistem sesuai dengan hasil yang sebenarnya. Tabel *confusion matrix*, seperti yang terlihat pada Tabel 1 menggambarkan perbandingan antara nilai aktual dan prediksi dari klasifikasi. Tabel 1 menggambarkan perbandingan antara nilai aktual dan nilai prediksi dari klasifikasi [18].

		Nilai Prediksi	
		Normal	Kanker
Nilai Aktual	Positif	<i>True Positive</i>	<i>False Negative</i>
	Negatif	<i>False Positive</i>	<i>True Negative</i>

Beberapa parameter digunakan untuk mengukur kinerja sistem yang akan dievaluasi dalam studi ini, termasuk:

2.5.1. Accuracy

Accuracy adalah ukuran seberapa akurat sistem mengklasifikasikan *input* untuk menghasilkan *output* yang sesuai dengan kelasnya. Rumus untuk menghitung *accuracy* dapat ditemukan dalam persamaan (1) sebagai berikut:

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

2.5.2. Precision

Precision adalah ukuran akurasi prediksi positif benar relatif terhadap total prediksi positif yang dibuat oleh sistem. Perhitungan *precision* dapat dilakukan menggunakan persamaan berikut (2) sebagai berikut:

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

2.5.3. Recall

Recall atau sensitivitas adalah ukuran akurasi sistem dalam memprediksi nilai positif benar relatif terhadap total nilai positif sebenarnya. Persamaan matematis untuk *recall* dapat dilihat pada persamaan (3) sebagai berikut:

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

2.5.4. F1-Score

F1-Score adalah metrik yang menggabungkan nilai *precision* dan *recall* dengan mempertimbangkan *false positives* dan *false negatives*. Persamaan untuk menghitung F1-Score dapat ditemukan pada persamaan (4) sebagai berikut:

$$F1 - Score = 2x \frac{precision \times recall}{precision + recall} \quad (4)$$

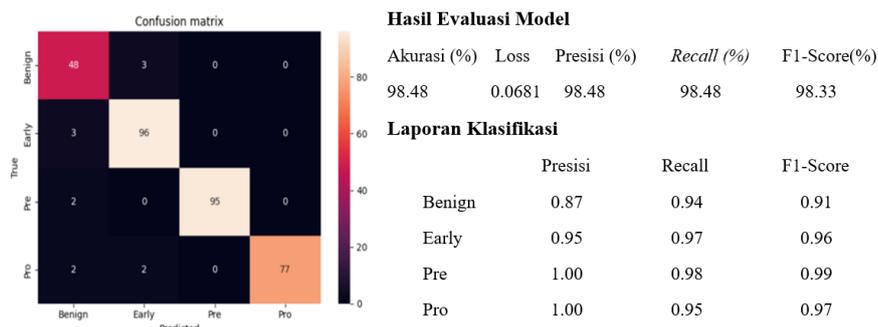
2.5.5. Loss

Parameter *loss* menginterpretasikan ketidakakuratan sistem dalam membuat prediksi untuk mengklasifikasikan sebuah input.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Hasil Pengujian *Hyperparameter* pada *Model Deep Learning*

Hasil yang diperoleh dari pengujian *hyperparameter* mengungkapkan model optimal untuk setiap arsitektur EfficientNet-B0. Selanjutnya, evaluasi model menggunakan data uji dilakukan untuk menilai kinerjanya dalam mengklasifikasikan gambar yang belum pernah dilihat sebelumnya. Proses evaluasi ini diterapkan dan diwakili menggunakan *confusion matrix*, yang memberikan wawasan tentang kinerja model pada berbagai kelas serta efektivitas keseluruhannya. Temuan dari penilaian *hyperparameter* pada model *deep learning* untuk arsitektur EfficientNet-B0 menunjukkan bahwa penggunaan *optimizer* Adam, dengan *learning rate* 0.0001, dan 30 *epoch* menghasilkan akurasi model sebesar 98.48%. *Confusion matrix* yang dihasilkan, yang ditunjukkan pada Gambar 4 menyoroti bahwa sebagian besar kesalahan prediksi terjadi pada kelas pro, di mana 2 gambar salah diklasifikasikan sebagai kelas *early* dan 2 lainnya sebagai kelas benign. Menariknya, kelas pre menunjukkan nilai recall tertinggi, mencapai 0.98. Secara keseluruhan, metrik evaluasi model untuk arsitektur EfficientNet-B0, termasuk *precision*, *recall*, dan F1-Score, masing-masing berada pada 98.48%, 98.48%, dan 98.33%.



Gambar 4. Hasil evaluasi model pada arsitektur EfficientNet-B0

3.2. Analisis Hasil Pengujian *Hyperparameter* pada *Model Deep Learning*

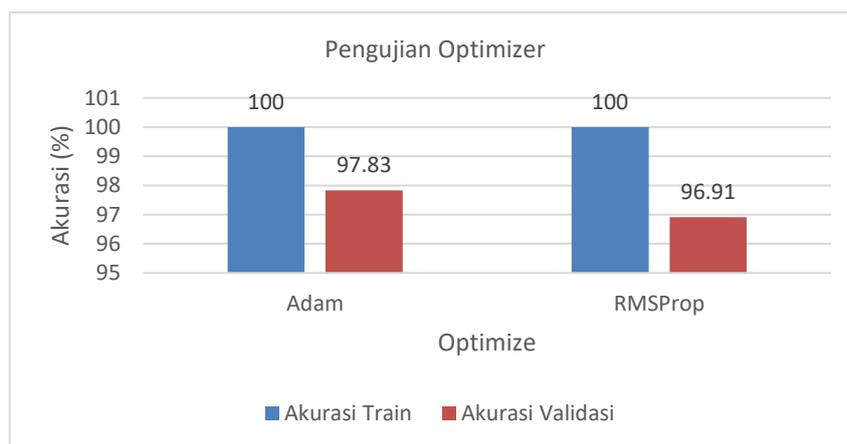
Pada tahap ini, pengujian skenario dilakukan pada *hyperparameter* arsitektur EfficientNet-B0. *Hyperparameter* adalah pengaturan yang disesuaikan sebelum proses pelatihan model dimulai [21]. Pengujian *hyperparameter* meliputi *epoch* (20, 30, 50), *learning rate* (0.0001, 0.0003, 0.001, 0.003), dan *optimizer* Adam dan RMSProp. Pengujian dilakukan dengan tujuan menemukan model terbaik berdasarkan tingkat akurasi tertinggi ketika dievaluasi menggunakan dataset validasi.

3.2.1. Kinerja Model dengan Arsitektur EfficientNet-B0

Pengujian *hyperparameter* yang dilakukan melibatkan tahap model dengan arsitektur EfficientNet-B0. Skenario pengujian *hyperparameter* dilakukan melalui tiga tahap, menghasilkan hasil berikut:

a. *Optimizer*

Pengujian *hyperparameter* pertama yang dilakukan pada arsitektur EfficientNet-B0 melibatkan perbandingan jenis *optimizer* selama proses pelatihan. Jenis *optimizer* yang akan dibandingkan adalah Adam dan RMSprop. Pengujian dilakukan menggunakan parameter lain seperti *learning rate* 0.0001 dan 30 *epoch*, dengan ukuran gambar 224 x 224 dan batch size 32. Berdasarkan parameter ini, hasil pengujian disajikan pada Gambar 5.

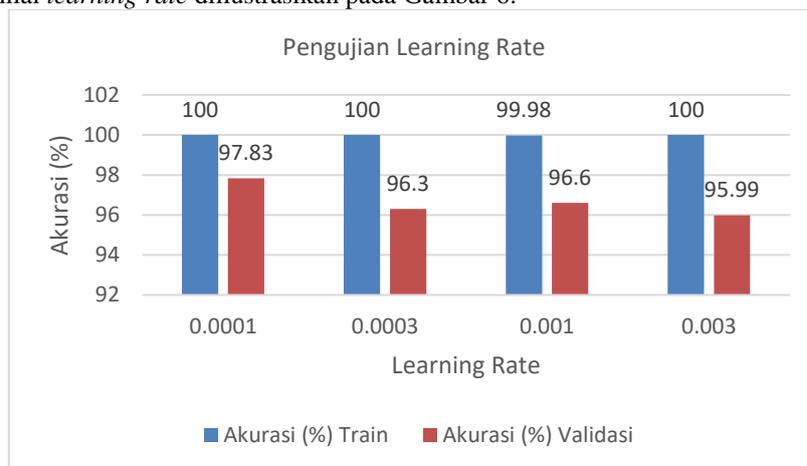


Gambar 5. Hasil pengujian *optimizer* dengan arsitektur EfficientNet-B0

Berdasarkan Gambar 5 dapat disimpulkan bahwa perbedaan jenis *optimizer* yang digunakan berdampak pada kinerja model yang dihasilkan. Hal ini terlihat dari hasil akurasi kedua jenis *optimizer* tersebut. Dari data, dapat diamati bahwa *optimizer* Adam mencapai akurasi tertinggi dibandingkan dengan *optimizer* RMSprop, dengan nilai masing-masing sebesar 100% dan 97.83%. Oleh karena itu, dalam pengujian *hyperparameter* pertama ini, model dengan kinerja terbaik dihasilkan oleh *optimizer* Adam, dan jenis *optimizer* ini akan digunakan untuk pengujian skenario berikutnya.

b. Learning Rate

Skenario pengujian *hyperparameter* kedua melibatkan perbandingan kinerja model dengan empat nilai *learning rate* yang berbeda untuk menentukan model optimal menggunakan hasil dari skenario awal. Parameter yang digunakan dalam skenario pengujian ini meliputi jenis *optimizer* Adam dan 30 *epoch*. Hasil pengujian untuk berbagai nilai *learning rate* diilustrasikan pada Gambar 6.

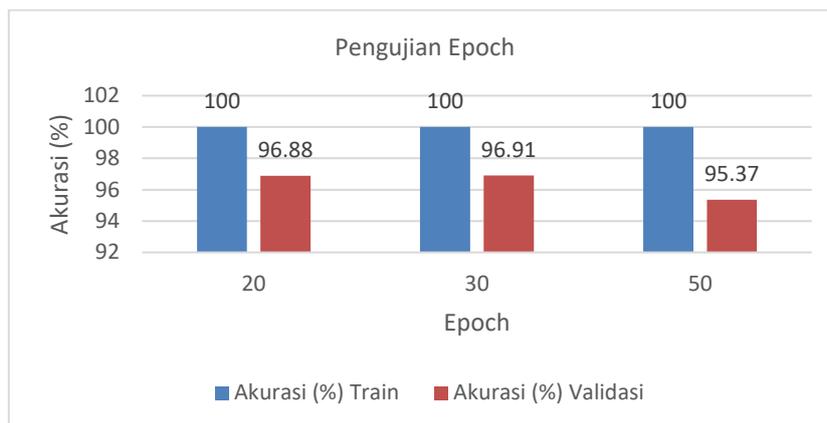


Gambar 6. Hasil pengujian *Learning Rate* dengan arsitektur EfficientNet-B0

Menurut data dalam Gambar 6 terlihat bahwa *learning rate* yang lebih rendah menghasilkan akurasi lebih tinggi dibandingkan *learning rate* yang lebih tinggi, yang cenderung menghasilkan akurasi lebih rendah. *Learning rate* yang paling efektif terlihat pada nilai 0.0001, menghasilkan akurasi pelatihan sebesar 100% dan akurasi validasi sebesar 97.83%. Oleh karena itu, untuk skenario berikutnya, nilai *learning rate* yang akan digunakan adalah 0.0001.

c. Epoch

Skenario pengujian ketiga dilakukan untuk memastikan kinerja model optimal dengan membandingkan nilai akurasi pada epoch 20, 30, dan 50. Parameter yang digunakan dalam skenario ini sesuai dengan pengujian *hyperparameter* sebelumnya, yaitu menggunakan *optimizer* Adam dan *learning rate* 0.0001. Hasil pengujian dapat diamati pada bagian berikutnya. Hasil pengujian dapat dilihat pada Gambar 7.



Gambar 7. Hasil penguujian *epoch* dengan arsitektur EfficientNet-B0

Menurut data dalam Gambar 7 terlihat bahwa nilai akurasi tertinggi pada dataset validasi dicapai pada *epoch* 30, mencapai 96.91%. Oleh karena itu, *epoch* 30 muncul sebagai *epoch* yang paling menguntungkan dibandingkan dengan nilai *epoch* lainnya dalam skenario penguujian *hyperparameter* ini.

4. KESIMPULAN

Berdasarkan pembahasan pada bab sebelumnya, kesimpulan berikut dapat diambil penguujian skenario dilakukan pada *hyperparameter* dalam arsitektur EfficientNet-B0. *Hyperparameter* yang diuji meliputi *epoch* (20, 30, 50), *learning rate* (0.0001, 0.0003, 0.001, 0.003), dan penggunaan *optimizer* Adam dan RMSProp. Penguujian *hyperparameter* model terbaik diperoleh dari arsitektur EfficientNet-B0. Arsitektur EfficientNet-B0 mencapai parameter terbaik dengan *learning rate* 0.0001, *optimizer* Adam, dan 30 *epoch*. Sedangkan untuk nilai akurasi kinerja diperoleh menggunakan data pelatihan, validasi, dan uji untuk arsitektur EfficientNet-B0. Arsitektur EfficientNet-B0 mencapai akurasi validasi sebesar 97.84%, dan ketika diuji dengan data uji, akurasinya mencapai 98.48%. Diharapkan pada penelitian selanjutnya dapat melakukan pada arsitektur seperti Alexnet, atau VGG-18 untuk memperluas cakupan evaluasi model dan performa.

UCAPAN TERIMAKASIH

Saya ingin mengungkapkan rasa terima kasih yang tulus kepada LPPM IT Telkom Purwokerto, orang tua saya, dan teman-teman terdekat saya, Fatimah Azzahra, Maryam Burhanuddin, dan Silfanny, atas bimbingan, dukungan, dan dorongan yang luar biasa selama perjalanan saya dalam menyelesaikan penelitian saya.

DAFTAR PUSTAKA

- [1] A. Rehman, N. Abbas, T. Saba, S. I. ur Rahman, Z. Mehmood, and H. Kolivand, "Classification of acute lymphoblastic leukemia using deep learning," *Microsc Res Tech*, vol. 81, no. 11, pp. 1310–1317, 2018, doi: 10.1002/jemt.23139.
- [2] M. Z. Ullah *et al.*, "An attention-based convolutional neural network for acute lymphoblastic leukemia classification," *Applied Sciences (Switzerland)*, vol. 11, no. 22, 2021, doi: 10.3390/app112210662.
- [3] A. R. Praidia, "Pengenalan Penyakit Darah menggunakan Teknik Pengolahan Citra dan Jaringan Syaraf Tiruan," *Thesis*, pp. 1–9, 2008.
- [4] J. E. Cortes, H. Kantarjian, and E. J. Freireich, "Acute lymphocytic leukemia: a comprehensive review with emphasis on biology and therapy.," *Cancer Treat Res*, vol. 84, pp. 291–323, 1996, doi: 10.1007/978-1-4613-1261-1_13.
- [5] T. T. P. Thanh, C. Vununu, S. Atoev, S.-H. Lee, and K.-R. Kwon, "Leukemia Blood Cell Image Classification Using Convolutional Neural Network," *International Journal of Computer Theory and Engineering*, vol. 10, no. 2, pp. 54–58, 2018, doi: 10.7763/ijcte.2018.v10.1198.
- [6] D. Kumar *et al.*, "Automatic Detection of White Blood Cancer from Bone Marrow Microscopic Images Using Convolutional Neural Networks," *IEEE Access*, vol. 8, no. Mm, pp. 142521–142531, 2020, doi: 10.1109/ACCESS.2020.3012292.
- [7] W. Yu *et al.*, "Automatic classification of leukocytes using deep neural network," *Proceedings of International Conference on ASIC*, vol. 2017-October, pp. 1041–1044, 2017, doi: 10.1109/ASICON.2017.8252657.
- [8] M. J. Macawile, V. V. Quiñones, A. B. Jr, J. Dela Cruz, and M. V. Caya, "2018 3rd International Conference on Control and Robotics Engineering : White Blood Cell Classification and Counting Using Convolutional Neural Network," *Nagoya Kōgyō Daigaku Institute of Electrical and Electronics Engineers*, pp. 259–263, 2018.

- [9] N. Sampathila *et al.*, “Customized Deep Learning Classifier for Detection of Acute Lymphoblastic Leukemia Using Blood Smear Images,” *Healthcare (Switzerland)*, vol. 10, no. 10, 2022, doi: 10.3390/healthcare10101812.
- [10] A. R. Revanda, C. Fatichah, and N. Suciati, “Classification of Acute Lymphoblastic Leukemia on White Blood Cell Microscopy Images Based on Instance Segmentation Using Mask R-CNN,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 5, pp. 625–637, 2022, doi: 10.22266/ijies2022.1031.54.
- [11] S. Shafique and S. Tehsin, “Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks,” *Technol Cancer Res Treat*, vol. 17, pp. 1–7, 2018, doi: 10.1177/1533033818802789.
- [12] J. Prellberg and O. Kramer, “Acute lymphoblastic leukemia classification from microscopic images using convolutional neural networks,” *Lecture Notes in Bioengineering*, pp. 53–61, 2019, doi: 10.1007/978-981-15-0798-4_6.
- [13] M. Claro *et al.*, “Convolution Neural Network Models for Acute Leukemia Diagnosis,” *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July, pp. 63–68, 2020, doi: 10.1109/IWSSIP48289.2020.9145406.
- [14] S. Perveen, A. Alourani, M. Shahbaz, U. Ashraf, and I. Hamid, “A framework for Early Detection of Acute Lymphoblastic Leukemia and its Subtypes from Peripheral Blood Smear Images Using Deep Ensemble Learning Technique,” *IEEE Access*, vol. 12, no. February, pp. 29252–29268, 2024, doi: 10.1109/ACCESS.2024.3368031.
- [15] T. Terwilliger and M. Abdul-Hay, “Acute lymphoblastic leukemia: a comprehensive review and 2017 update,” *Blood Cancer J*, vol. 7, no. 6, 2017, doi: 10.1038/BCJ.2017.53.
- [16] A. Howard *et al.*, “Searching for mobileNetV3,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.
- [17] N. Putri, “Implementasi Metode CNN Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita),” pp. 1–80, 2020.
- [18] M. S. Wibawa, “Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan,” *Jurnal Sistem dan Informatika (JSI)*, vol. 11, no. December, pp. 167–174, 2017, doi: 10.13140/RG.2.2.21139.94241.
- [19] I. R. W. Putra, “Deteksi Jenis Buah-Buahan Menggunakan Deep,” *Fakultas Teknologi dan Informatika Universitas Dinamika*, p. 60, 2020.
- [20] A. Rohim, Y. A. Sari, and Tibyani, “Convolution neural network (cnn) untuk pengklasifikasian citra makanan tradisional,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 7, pp. 7038–7042, 2019.
- [21] H. Mubarok, “Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN),” *Universitas Islam Negeri Maulana Malik Ibrahim Malang*, vol. 3, no. 1, pp. 10–12, 2019.