



Implementasi *Convolutional Neural Network* berbasis *Transfer Learning* untuk Klasifikasi *Acute Lymphoblastic Leukemia*

Rana Adinda Manalus Fata¹, Jenny Putri Hapsari²

^{1,2}Electrical Engineering, Faculty of Industrial Technology, Universitas Islam Sultan Agung Semarang

Email author: adindarana24@gmail.com¹, jenny@unissula.ac.id²

Article Info

Article history:

Received May 20, 2026

Revised May 21, 2026

Accepted June 28, 2026

Available: July 7, 2026

Published: July 30, 2026

Keywords:

Leukemia

Convolutional Neural Network

Deep Learning

Transfer Learning

EfficientNet-B3

ABSTRACT

Leukemia is a cancer that originates in human blood cells. The most common type of leukemia (97%), with an incidence of 4–4.5 cases per 100,000 children per year, is Acute Lymphoblastic Leukemia (ALL)[1]. This indicates that leukemia can progress rapidly and become fatal for the patient within a few months. Therefore, a supporting method is needed that can classify blood cells automatically, quickly, and accurately. This method is a Convolutional Neural Network (CNN) using the EfficientNet-B3 architecture as a pre-trained model or for transfer learning. This dataset consists of 3,527 blood cell images that have been preprocessed to a size of 224x224x3 and image enhancement has been applied. The images were trained on the pre-trained model and then combined with Global Average Pooling (GAP), Batch Normalization, Dense, and Softmax layers until the model could classify the images into the ALL or HEM classes. The results of the study show that the EfficientNet-B3 architecture is capable of classifying white blood cell images into the ALL and HEM classes through the transfer learning process. The best hyperparameter configuration for optimal results includes a learning rate of 0.0001 and the RMSProp optimizer. The model achieved the best training accuracy of 100% at epoch 30 and a batch size of 16, while the best testing accuracy was 96% at epoch 50 and a batch size of 16. Additionally, the precision, recall, and F1-score were 96%, 94%, and 95%, respectively.

Corresponding Author:

Rana Adinda Manalus Fata,

Universitas Islam Sultan Agung Semarang

Jl. Kaligawe Raya No.Km.4, Terboyo Kulon, Kec. Genuk, Kota Semarang, Jawa Tengah 50112

Email: adindarana24@gmail.com



1. LATAR BELAKANG

Leukemia adalah kanker yang berasal dari sel darah manusia. Hal ini disebabkan oleh tubuh yang memproduksi sel darah abnormal dalam jumlah yang banyak. Sel darah abnormal merupakan kelompok sel darah putih yang dapat menyebabkan tidak dapat berfungsinya sel-sel darah tersebut dengan baik[2]. Jenis leukemia yang paling sering terjadi (97%) dengan insidensi sebesar 4–4,5 kasus/100.000 anak per-tahun adalah *Acute Lymphoblastic Leukemia* (ALL)[1]. Hal tersebut

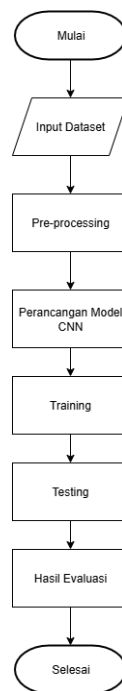
menunjukkan bahwa *leukemia* dapat berkembang dengan cepat dan menjadi fatal bagi kondisi pasien dalam beberapa bulan [3]. Diagnosis *leukemia* biasanya melibatkan analisis jumlah sel darah secara menyeluruh, pengambilan sampel sumsum tulang, dan analisis mikroskopis oleh seorang ahli hematologi. Namun, prosedur ini sering kali memerlukan waktu yang lama, biaya yang tinggi, dan bergantung pada keterampilan tenaga medis. Selain itu, hasil dari analisis manual dapat dipengaruhi oleh pandangan subjektif, yang dapat meningkatkan risiko terjadinya kesalahan diagnosa [4]. Oleh karena itu, dibutuhkan suatu metode pendukung yang mampu melakukan klasifikasi sel darah secara otomatis, cepat, dan akurat.

Seiring berkembangnya teknologi dalam deep learning, khususnya pengolahan citra digital kini semakin umum digunakan untuk mengklasifikasikan sel darah. Perkembangan teknologi ini mencakup pengambilan data citra pada sebuah sampel sel darah yang di input kedalam sebuah proses pengolahan citra. Pengolahan citra tersebut menyerupai fungsi jaringan saraf yang ada di dalam otak manusia. Metode ini adalah *Convolutional Neural Network* (CNN). CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia, sehingga memiliki kemampuan mengolah informasi citra [5]. Dalam melakukan proses pembelajaran pada CNN dapat memakan proses waktu komputasi yang lama dan memperlambat proses klasifikasi, oleh karena itu perlu ditambahkan *transfer learning* dalam proses pembelajaran tersebut. *Transfer learning* memungkinkan proses pelatihan model dilakukan secara efektif meskipun dengan keterbatasan jumlah data pelatihan, karena model tersebut telah dilatih sebelumnya menggunakan dataset berskala besar [6]. Salah satu arsitektur yang dikembangkan untuk mendukung pendekatan transfer learning adalah *EfficientNet*.

Beberapa penelitian sebelumnya menunjukkan CNN berhasil mencapai tingkat akurasi yang tinggi dalam mengklasifikasikan citra medis, termasuk citra sel darah dalam mendeteksi *leukemia*. Ni Putu Dina Agustina tentang “Klasifikasi Subtipe Leukemia Limfoblastik Akut (LLA) pada Citra Mikroskopis Sel Darah Menggunakan Arsitektur *EfficientNet-B3* dengan Dataset Seimbang” model ini mencapai akurasi pengujian 97,50% dan Skor F1 rata-rata 0,97 melalui pendekatan *transfer learning* [4]. Selain itu, Hani et al., dengan judul “Prediksi Kanker Darah Menggunakan Metode *Convolutional Neural Network*” menunjukkan bahwa implementasi CNN dengan arsitektur *MobileNetV2* dalam memprediksi sel kanker darah putih menghasilkan nilai akurasi 95.6%, presisi 94.8%, recall 96.2%, dan F1-score 95.5% [7]. Berdasarkan latar belakang tersebut, penelitian ini berfokus pada model *transfer learning EfficientNet-B3* untuk klasifikasi *Acute Lymphoblastic Leukemia*. Penelitian ini diharapkan dapat memberikan hasil klasifikasi yang dapat dibagi menjadi dua kelas, yaitu sel darah putih normal dan sel darah putih yang mengalami kanker (*Acute Lymphoblastic Leukemia/ALL*).

2. METODE PENELITIAN

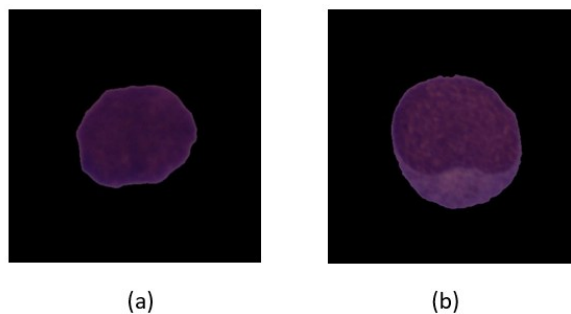
Pada perancangan ini, dilakukan dengan penerapan metode CNN untuk mengklasifikasikan penyakit kanker sel darah putih dengan dua kategori kelas yaitu sel Normal dan sel Kanker. Berikut bentuk alur perancangan yang akan dilihat pada Gambar 1 berikut:



Gambar 1 Flowchart Penelitian

2.1 Input Dataset

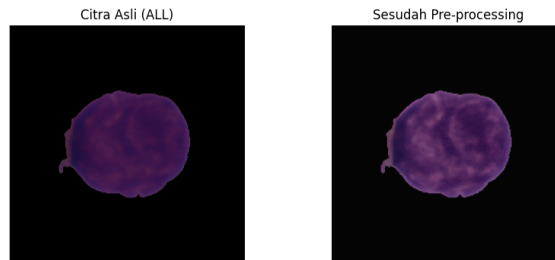
Dataset yang digunakan pada penelitian ini diambil dari *ALL Challenge dataset of ISBI 2019* (C-NMC 2019) dibuat oleh *The Cancer Imaging Archive (TCIA) Public Access* dengan format bpm atau bitmap.



Gambar 2 Citra Mikroskopis Leukemia (a) Sel Leukemia (b) Sel Normal [8]

2.2 Pre-Processing

Pada tahap ini citra gambar akan *resize* atau perubahan ukuran gambar. Tujuan dari *resize* ini untuk menentukan ukuran piksel yang digunakan dalam pemrosesan citra. Selain itu, penggunaan *image enchancement* berfungsi untuk memperbaiki kualitas citra agar memperoleh citra yang lebih sesuai digunakan untuk aplikasi lebih lanjut.

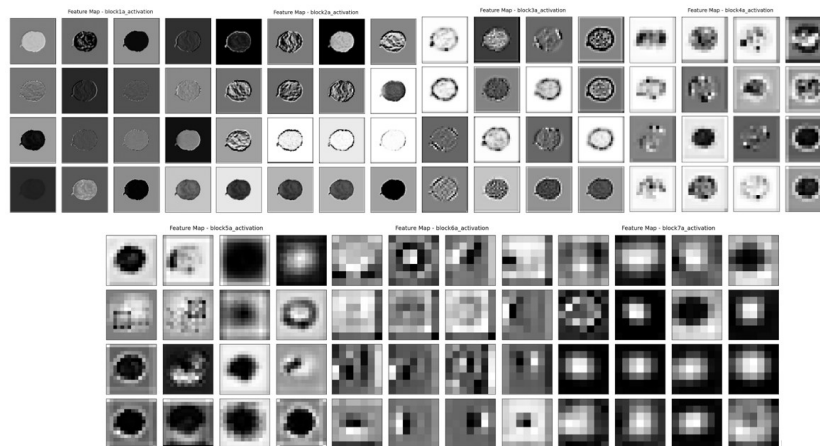


Gambar 3 Citra asli dan Citra sesudah dipre-processing

3.3 Perancangan Model CNN

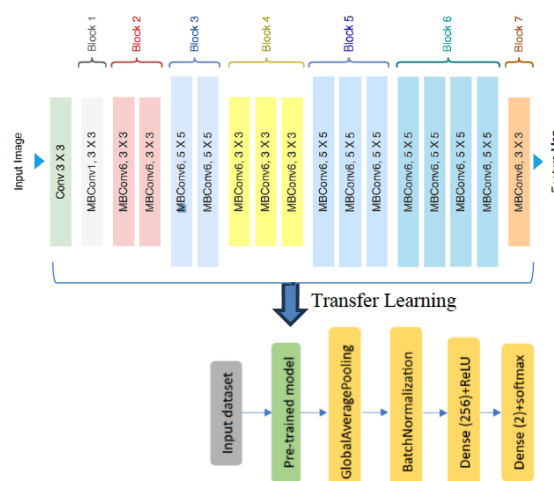
Perancangan model sistem CNN kali ini menggunakan arsitektur *EfficientNet-B3* sebagai pre-trained atau transfer learning. Metode *transfer learning* pada arsitektur *EfficientNet-B3* dilakukan untuk melatih arsitektur dengan data atau parameter baru. Input citra dalam proses transfer learning menggunakan citra dengan jumlah kelas yang sudah disediakan sebelumnya oleh *ImageNet*. Karena kelas citra yang akan di uji pada pemodelan ini adalah kelas ALL dan HEM, maka layer yang diambil untuk *training* ulang adalah bagian *layer pre-trained*. Setelah mengambil lapisan pada *layer pre-trained*, selanjutnya akan digabungkan dengan layer baru yang dimodelkan sendiri. Selama proses tersebut, model akan mempelajari fitur baru yang dapat diperoleh dari dataset baru.

Untuk tipe model arsitektur *Convolutional Neural Network* (CNN) yang dirancang yaitu menggunakan tipe *sequential* yang dibuat menggunakan *library* bawaan dari *TensorFlow* dan *Keras*. Input citra yang dipakai menggunakan citra dengan ukuran original 450x450x3 piksel. Jadi sebelum masuk ke model, citra dilakukan *pre-processing* dengan *resize* menjadi ukuran 224x224x3 piksel agar sesuai dengan ukuran *input* model *EfficientNet-B3*. Selain itu, citra akan ditambahkan CLAHE (*Contrast Limited Adaptive Histogram Equalization*) untuk meningkatkan kontras pada citra tanpa menimbulkan *noise* berlebih. Input citra hasil *pre-processing* kemudian dimasukkan ke dalam arsitektur *EfficientNet-B3* yang terdiri dari tujuh blok utama untuk mengekstraksi fitur secara bertahap, di mana setiap blok dibangun menggunakan *Mobile Inverted Bottleneck Convolution* (MBConv). Blok MBConv memiliki struktur khas, yaitu memperluas dimensi kanal melalui konvolusi 1x1, kemudian melakukan *depthwise convolution* dengan kernel berukuran 3x3 atau 5x5, diikuti mekanisme *squeeze-and-excitation* (SE) untuk menonjolkan kanal-kanal penting, serta akhirnya memproyeksikan kembali hasilnya dengan konvolusi 1x1. Arsitektur ini memungkinkan jaringan untuk tetap efisien dalam jumlah parameter namun tetap mampu menangkap informasi yang relevan. Pada blok pertama, digunakan MBConv1 dengan kernel 3x3, sementara pada blok-blok berikutnya digunakan MBConv6 dengan kernel 3x3 atau 5x5 dan jumlah pengulangan yang berbeda-beda sesuai tahapannya [9]. Proses ini dilakukan secara bertingkat dari blok 1 hingga blok 7. Output dari blok terakhir *EfficientNet-B3* berupa *feature map* yang telah merepresentasikan karakteristik fitur citra secara menyeluruh. Pada setiap blok akan menghasilkan fitur-fitur berbeda yang ditangkap dari blok sebelumnya. Semakin kedalam citra, fitur yang ditangkap semakin unik. *Feature map* inilah yang selanjutnya digunakan sebagai masukan ke tahap pemrosesan lanjutan pada bagian *classification*.



Gambar 4 Feature Maps

Selanjutnya, *feature maps* tersebut diproses menggunakan *global average pooling* (GAP). Pada *layer* ini akan mengambil rata-rata dari fitur yang dihasilkan oleh lapisan sebelumnya dalam tiap *channel* [10]. Selanjutnya, vektor ini diteruskan ke *layer batch normalization* untuk menstabilkan proses pelatihan. Setelah itu, dilanjutkan ke *layer dense* (256) dan aktivasi *relu* untuk mengekstraksi fitur yang lebih abstrak dari fitur-fitur yang telah diperoleh sebelumnya. Terakhir, untuk *output layer* menggunakan *dense* (2) dan fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* mengembalikan probabilitas dari setiap kelas, di mana kelas target akan memiliki nilai probabilitas yang lebih tinggi daripada kelas lainnya. Berikut perancangan model CNN yang akan di buat ada pada Gambar 5



Gambar 5 Perancangan Model CNN

3.4 Pelatihan

Pada tahap ini, sistem yang telah dirancang akan dipelajari oleh data citra. Data pada pelatihan ini menggunakan data uji dengan jumlah 80% dari total seluruh dataset. Pelatihan dilakukan dengan menggunakan variasi 10, 20, 30, 40, 50 dan 60 *epoch* serta *learning rate* sebesar 0,0001. *Batch size* yang digunakan sebesar 32 dan 16 dengan menambahkan optimizer *RMSprop* agar mencapai nilai yang optimum. Selain itu, menggunakan *hyperparameter* tambahan yakni dengan *decay step* 900 dan *decay rate* 0.95. Keluaran dari pelatihan ini adalah hasil *training loss*, *validation loss*, *training accuracy* dan

validation accuracy dari pemodelan yang telah dibangun dan hasil keluaran akan divisualisasikan melalui kurva *training*.

3.5 Pengujian

Setelah dilakukan pelatihan, selanjutnya data citra akan di uji. Pengujian tersebut dilakukan dengan cara mengetahui nilai dari *confusion matrix*nya. Evaluasi model dilakukan dengan memonitor jumlah *True Positif (TP)*, *True Negatif (TN)*, *False Positif (FP)*, dan *False Negatif (FN)*. Dari data-data tersebut, dapat dihitung *accuracy*, *precision*, *recall* dan *f1-score* dari model yang dihasilkan[11]. Berikut untuk persamaan dari *accuracy*, *presision*, *recall*, dan *f1- score* (1), (2), (3) dan (4) [12].

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

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

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

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

3. HASIL DAN ANALISA

3.1. Hasil Model Training

Pada pelatihan ini, telah disimulasikan pengaruh variasi *epoch* 10, 20, 30, 40, 50, dan 60 serta *batch size* sebesar 32 dan 16 menggunakan arsitektur *EfficientNet-B3* dengan sistem transfer learning. Pada Gambar 6 merupakan salah satu contoh dari hasil *training accuracy* yang dijalankan di platform *Google Collab*.

```
Epoch 1/10
89/89 ————— 217s 1s/step - accuracy: 0.6311 - loss: 0.9857 - val_accuracy: 0.7224 - val_loss: 0.5785
Epoch 2/10
89/89 ————— 23s 259ms/step - accuracy: 0.8230 - loss: 0.4991 - val_accuracy: 0.7720 - val_loss: 0.4688
Epoch 3/10
89/89 ————— 25s 277ms/step - accuracy: 0.8851 - loss: 0.3064 - val_accuracy: 0.8555 - val_loss: 0.3653
Epoch 4/10
89/89 ————— 23s 256ms/step - accuracy: 0.9171 - loss: 0.2207 - val_accuracy: 0.8555 - val_loss: 0.3666
Epoch 5/10
89/89 ————— 23s 256ms/step - accuracy: 0.9283 - loss: 0.1847 - val_accuracy: 0.8739 - val_loss: 0.3291
Epoch 6/10
89/89 ————— 23s 262ms/step - accuracy: 0.9585 - loss: 0.1179 - val_accuracy: 0.8725 - val_loss: 0.3466
Epoch 7/10
89/89 ————— 23s 260ms/step - accuracy: 0.9502 - loss: 0.1225 - val_accuracy: 0.7748 - val_loss: 0.5940
Epoch 8/10
89/89 ————— 23s 256ms/step - accuracy: 0.9594 - loss: 0.1091 - val_accuracy: 0.8980 - val_loss: 0.2758
Epoch 9/10
89/89 ————— 23s 257ms/step - accuracy: 0.9694 - loss: 0.1262 - val_accuracy: 0.9023 - val_loss: 0.3268
Epoch 10/10
89/89 ————— 23s 260ms/step - accuracy: 0.9708 - loss: 0.0805 - val_accuracy: 0.8229 - val_loss: 0.4856
Accuracy: 91.74%
```

Gambar 6 Hasil Accuracy Training Model Di Pemograman Pada 10 Epoch Dan Batch size 32

Berdasarkan hasil dari training model diatas, nilai *accuracy* semakin meningkat seiring bertambahnya *epoch*. Selanjutnya, nilai *accuracy training* pada setiap variasi *epoch* dengan *batch size* 32 terlihat dalam Tabel 1

Table 1 Hasil Model Accuracy Traningng Dengan Batch size 32

<i>Epoch</i>	<i>Accuracy</i>
10	91.74%
20	95.57%
30	99.96%
40	97.48%
50	99.40%
60	100%

Dari hasil pelatihan, model dengan menggunakan *batch size* 32 memiliki hasil *accuracy training* terbaik 100% dengan jumlah 60 *epoch* dan nilai *accuracy* yang kurang optimal adalah 91.74% dengan jumlah 10 *epoch*. Variasi *epoch* pada saat *training* cukup berpengaruh dimana semakin besar jumlah *epoch*, *accuracy* yang diperoleh juga akan semakin tinggi. Hal ini terjadi karena model memiliki lebih banyak kesempatan untuk mempelajari pola fitur pada data citra.

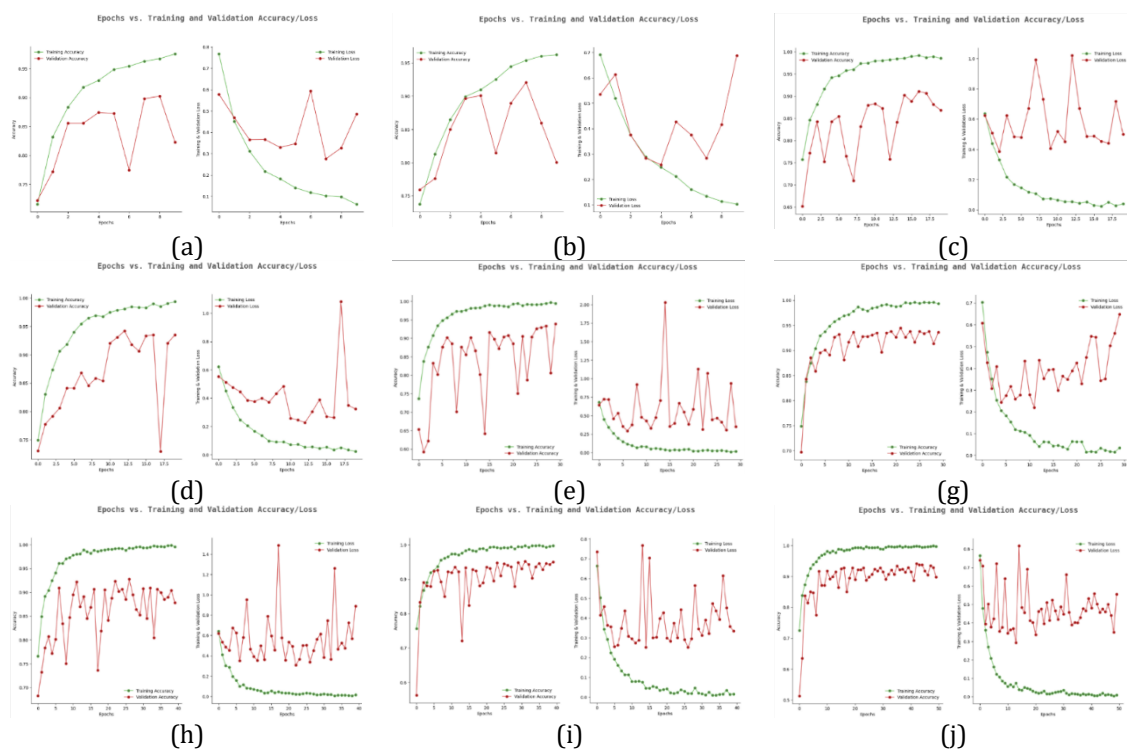
Table 2 Hasil Model Accuracy Training Dengan Batch size 16

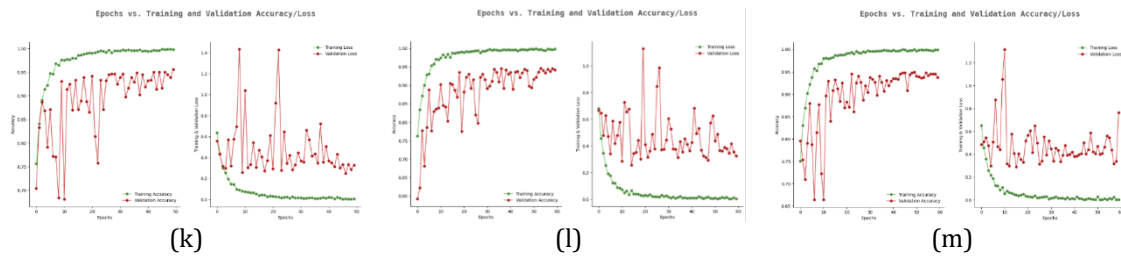
<i>Epoch</i>	<i>Accuracy</i>
10	89.58%
20	99.61%
30	100%
40	100%
50	100%
60	100%

Selanjutnya, untuk hasil pengujian *accuracy training* dengan menggunakan *batch size* 16 ditunjukkan pada Tabel 2, dapat dilihat bahwa *accuracy training* minimum adalah sebesar 89,58% pada jumlah 10 *epoch*, Dimana nilai ini lebih rendah dibandingkan *batch size* 32. Namun, pada *epoch* 20 *accuracy* meningkat menjadi 99,61% dan mencapai nilai 100% pada *epoch* 30, 40, 50, dan 60. Dan untuk *accuracy training* terbaik di dapatkan pada jumlah 30 *epoch*, dimana model mendapatkan nilai *accuracy* sebesar 100% dalam jumlah *epoch* yang sedikit. Oleh karena itu, dapat disimpulkan bahwa *batch size* 16 memberikan performa *training* yang lebih unggul dibandingkan *batch size* 32.

3.2. Kurva Model Training

Dari gambar kurva hasil pelatihan diatas, model dengan *batch size* 16 lebih cepat dalam melakukan pembelajaran daripada model dengan *batch size* 32. Terlihat pada kurva model dengan *batch size* 32, terdapat gap antara kurva *accuracy training* dan validasi yang cukup jauh yang ditunjukkan pada Gambar c dan Gambar h. Gap ini mengindikasikan bahwa model cenderung mengalami *overfitting*. Selain itu, kurva *loss* yang dihasilkan oleh model dengan *batch size* 32 belum sampai di nilai yang paling minimum. Sebaliknya, untuk model dengan *batch size* 16, kenaikan kurva training cukup stabil dan gap antar *training* dan validasi cukup dekat yang ditunjukkan pada Gambar i, Gambar k, dan Gambar m.

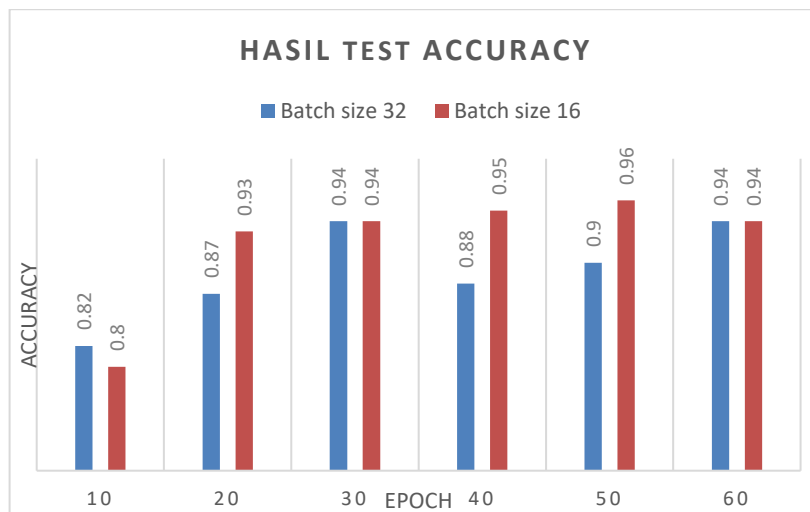




Gambar 7 Kurva Model Training Epoch 10, 20, 30, 40, 50 dan 60 dengan Batch size 32 dan 16

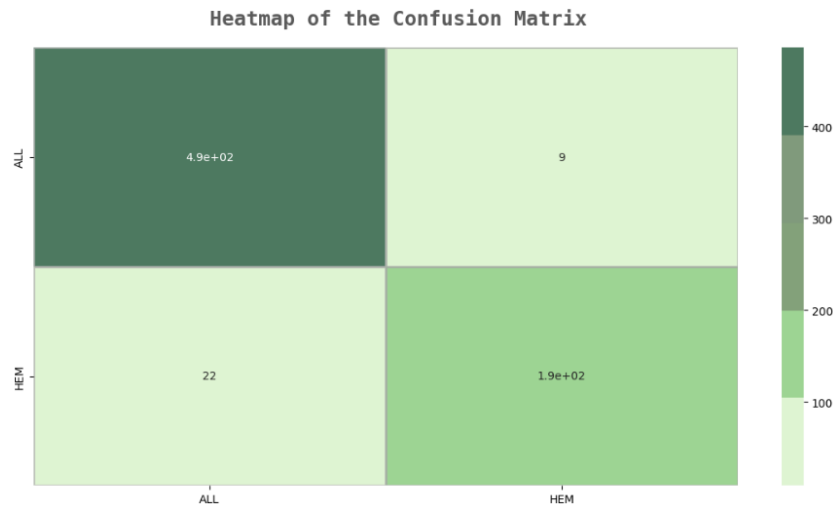
3.3. Hasil Model Pengujian

Dari hasil pengujian yang terdapat pada Gambar 8, diketahui bahwa semua model pengujian dengan menggunakan *batch size* 16 menunjukkan performa pengujian yang lebih unggul dibandingkan dengan *batch size* 32 pada hampir seluruh variasi *epoch*. Namun, pada jumlah 10 *epoch accuracy* dari model dengan *batch size* 32 lebih unggul dengan nilai 0.82 atau 82% dan *batch size* 16 hanya memperoleh 0.80 atau 80%. Terlihat pada *epoch* 20, 30, 40, 50 dan 60, model dengan *batch size* 16 secara konsisten menghasilkan nilai test *accuracy* yang lebih tinggi, yaitu mencapai 0,93 atau 93% pada 20 *epoch* 0.93 atau 93% pada 30 *epoch*, 0.95 atau 95% pada 40 *epoch*, dan nilai tertinggi sebesar 0,96 atau 96% pada 50 *epoch*. Sementara itu, model dengan *batch size* 32 pada *epoch* yang sama menunjukkan nilai *accuracy* yang lebih rendah. Secara keseluruhan *batch size* 16 tetap menunjukkan keunggulan karena mampu mencapai nilai test *accuracy* yang lebih optimum dan lebih stabil pada sebagian besar variasi *epoch*.



Gambar 8 Hasil Model Pengujian

Nilai test *accuracy* yang diperoleh pada setiap pengujian dihitung berdasarkan hasil *confusion matrix* pada data uji. Sebagai contoh, pada pengujian model dengan *epoch* 50 dan *batch size* 16, diperoleh hasil evaluasi sebagai berikut.

Gambar 9 Confusion Matrix pada *Epoch* 50 dan Batch size 16

Sesuai dengan persamaan (1), Nilai *accuracy* diperoleh dari perbandingan jumlah prediksi yang benar yaitu TP dan TN terhadap keseluruhan jumlah data citra yang diuji

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

$$Accuracy = \frac{490+190}{490+22+9+190}$$

$$Accuracy = \frac{680}{711}$$

$$Accuracy = 0.956 \approx 0.96.$$

Berdasarkan hasil perhitungan tersebut, model mampu mengklasifikasikan citra sel darah putih leukemia dengan nilai *accuracy* sebesar 96%.

3.4 Hasil Rata-rata nilai Precision, Recall dan F1-score

Table 3 Hasil Rata-rata nilai Precision, Recall dan F1-score

Epoch	Batch size 32			Batch size 16		
	Precision	Recall	F1-score	Precision	Recall	F1-score
10	0.8	0.85	0.81	0.78	0.84	0.79
20	0.84	0.88	0.85	0.92	0.93	0.92
30	0.93	0.92	0.93	0.92	0.93	0.92
40	0.85	0.91	0.87	0.94	0.94	0.94
50	0.87	0.9	0.88	0.96	0.94	0.95
60	0.94	0.92	0.93	0.95	0.9	0.92

Untuk mendapatkan hasil rata-rata dari nilai *precision*, *recall*, dan *f1-score* diatas didapatkan sesuai dengan persamaan yang sudah dijelaskan pada persamaan (2), (3), dan (4). Sebagai contoh, digunakan data hasil pengujian terhadap 706 citra testing yang merujuk pada *confusion matrix* model terbaik pada Gambar 9 dengan rincian sebagai berikut:

Perhitungan untuk kelas ALL;

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

$$Precision = \frac{490}{512}$$

$$Precision = 0.96$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Recall} = \frac{490}{499}$$

$$\text{Recall} = 0.98$$

$$f1 - \text{score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$f1 - \text{score} = 2 * \frac{0.96 * 0.98}{0.96 + 0.98}$$

$$f1 - \text{score} = 0.97$$

Perhitungan untuk kelas HEM;

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Precision} = \frac{190}{199}$$

$$\text{Precision} = 0.95$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Recall} = \frac{190}{190+FN}$$

$$\text{Recall} = 0.90$$

$$f1 - \text{score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$f1 - \text{score} = 2 * \frac{0.95 * 0.90}{0.95 + 0.90}$$

$$f1 - \text{score} = 0.92$$

Berdasarkan perhitungan di atas, rata-rata nilai *precision*, *recall*, dan *F1-score*, dapat dilihat bahwa model yang paling unggul diperoleh pada penggunaan *batch size* 16, khususnya pada *epoch* 50. Sesuai dengan *confusion matrix* pada Gambar 9 untuk kelas ALL diperoleh nilai *True Positive* (TP) sebesar 490, *False Positive* (FP) sebesar 22, *False Negative* (FN) sebesar 9, dan *True Negative* (TN) sebesar 190. Berdasarkan perhitungan dengan persamaan (2), persamaan (3), dan persamaan (4), model tersebut menghasilkan nilai *precision* sebesar 0.96 yang berarti bahwa 96% data yang diprediksi sebagai kelas ALL oleh model benar-benar termasuk ke dalam kelas ALL. Selanjutnya, untuk nilai *recall* sebesar 0.98 menunjukkan bahwa sebanyak 98% dari seluruh data ALL yang sebenarnya berhasil terdeteksi oleh model. Dan untuk perbandingan rata-rata antara *precision* dan *recall* atau biasa disebut dengan *f1-score* memiliki nilai sebesar 0.97 atau 97%. Dan untuk kelas HEM, diperoleh nilai *True Positive* (TP) sebesar 190, *False Positive* (FP) sebesar 9, *False Negative* (FN) sebesar 22, dan *True Negative* (TN) sebesar 490 didapatkan nilai *precision* sebesar 0.95 atau 95%, nilai *recall* sebesar 0.90 atau 90% dan nilai *f1-score* mendapatkan nilai 0.92. Berdasarkan rata-rata dari kedua kelas, diperoleh nilai *precision* sebesar 0.96 *recall* sebesar 0.94 dan *f1-score* sebesar 0.95.

Sebaliknya, nilai yang kurang optimal diperoleh pada *epoch* 10 untuk kedua *batch size*. Pada model dengan *batch size* 32, model hanya mencapai *precision* sebesar 80%, *recall* 85%, dan *F1-score* 81%, sedangkan pada *batch size* 16 nilai *precision* 78%, *recall* 84%, dan *F1-score* 79%. Hal ini menunjukkan bahwa pada jumlah *epoch* yang rendah, model belum mampu mempelajari pola data secara optimal.

3.5. Hasil Klasifikasi Model Terbaik

Table 4 Hasil Training dan Testing Accuracy Model Terbaik

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Batch size</i>
30	100	16
60	100	32
<i>Epoch</i>	<i>Testing Accuracy</i>	<i>Batch size</i>
50	96	16

<i>Epoch</i>	<i>Training Accuracy</i>	<i>Batch size</i>
30	94	32

Berdasarkan Tabel 4, model terbaik pada proses *training* yang dilakukan pada *batch size* 16 mendapatkan nilai 100% pada *epoch* 30, sedangkan pada model dengan *batch size* 32 mendapatkan nilai 100% pada *epoch* 60. Dimana pada proses training menggunakan 2.257 data training dan performa model dipantau dengan menggunakan 564 data validasi untuk memastikan konsistensi akurasi sesuai dengan pembagian dataset sebelumnya. Dan untuk model terbaik pada proses *testing* dengan jumlah data 706 data *testing*, model dengan *batch size* 16 mendapatkan hasil 96% pada *epoch* 50 dan model dengan *batch size* 32 mendapatkan hasil 94% pada *epoch* 30. Secara keseluruhan, performa model dengan *batch size* 16 lebih unggul dari segi proses training dan testing. Oleh karena itu, model terbaik untuk pengklasifikasian *Acute Lymphoblastic Leukemia* menggunakan arsitektur *EfficientNet-B3* dengan sistem *transfer learning* adalah model dengan *batch size* 16.

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisa yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Arsitektur *EfficientNet-B3* mampu melakukan klasifikasi citra sel darah putih ke dalam kelas ALL dan HEM melalui proses transfer learning. Pada proses pertama citra yang diambil dari dataset akan melalui pre-processing dengan melakukan pelabelan pada setiap gambar dalam dataset dan citra tersebut di resize menjadi ukuran 224x224 serta dilakukan image enhancement untuk meningkatkan kualitas citra. Setelah itu, citra melewati tahap model pre-trained *EfficientNet-B3* dimana pada proses tersebut model memanfaatkan bobot yang telah dilatih sebelumnya dari ImageNet, kemudian layer terakhir dari *EfficientNet-B3* diteruskan dengan layer baru yang sudah ditambahkan. Feature map yang dihasilkan dari model pre-trained akan diteruskan Global Average Pooling, Batch Normalization, dan Dense (256). Terakhir sebelum sampai pengklasifikasian, akan melewati Dense (2) dan Fungsi Aktivasi Softmax untuk mengetahui probabilitas citra tersebut masuk ke dalam kelas ALL atau HEM.
2. Pada pengujian di atas, peforma model arsitektur *EfficientNet-B3* dengan menggunakan variasi *epoch* 10, 20, 30, 40, 50 dan 60 serta *batch size* 16 dan 32, didapatkan hasil *accuracy* training terbaik adalah 100% pada *epoch* 30 dan *batch size* 16. Sedangkan untuk *accuracy* testing terbaik didapatkan nilai 96% pada *epoch* 50 dan *batch size* 16. Berdasarkan hasil *confusion matrix* diperoleh nilai evaluasi untuk *precision*, recall dan f1-score berturut-turut didapatkan 96%, 94% dan 95% yang mengindikasikan bahwa model memiliki kemampuan klasifikasi yang baik dalam membedakan citra sel kedalam kelas ALL atau HEM dengan tingkat akurasi yang optimal dan tingkat kesalahan yang minimal, serta menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan.
3. Pengoptimalan pada model dapat dilakukan dengan cara mencoba semua *hyperparameter* untuk mendapatkan hasil yang tinggi. Dari semua percobaan, diperoleh *best hyperparameter* dengan *denser layer* 256 dan fungsi aktivasi relu, dropout 0.4, *learning rate* 0.0001, *batch size* 16 dan jumlah *epoch* 50 dan *optimizer* RMSProp. Dengan parameter tersebut dapat membantu model dalam mengoptimalkan proses pelatihan model, sehingga meningkatkan kemampuan klasifikasi *Acute Lymphoblastic Leukemia* ke dalam dua kelas secara lebih akurat dan stabil.

REFERENCES

- [1] C. M. Wairo, S. Nugroho, and H. Suyuti, "PERAN EKSPRESI p53 DAN SURVIVIN TERHADAP HEMOGLOBIN, LEUKOSIT, DAN TROMBOSIT PADA LEUKEMIA LIMFOBLASTIK AKUT ANAK YANG MENDAPATKAN KEMOTERAPI," *Majalah Kesehatan*, vol. 6, no. 1, pp. 23–34, 2019, doi: 10.21776/ub.majalahkesehatan.006.01.3.
- [2] A. R. Praid, "Pengenalan Penyakit Darah menggunakan Teknik Pengolahan Citra dan Jaringan Syaraf Tiruan," *Universitas Indonesia*, pp. 1–9, 2008.
- [3] A. N. Fajrina, Z. H. Pradana, S. I. Purnama, and S. Romadhona, "Penerapan Arsitektur EfficientNet-B0 Pada Klasifikasi Leukimia Tipe Acute Lymphoblastik Leukimia," *JURNAL RISET REKAYASA ELEKTRO*, vol. 6, no. 1, pp. 59–68, Jun. 2024, [Online]. Available: <http://jurnalnasional.ump.ac.id/index.php/JRRE>
- [4] N. P. D. Agustina, "Klasifikasi Subtipe Leukemia Limfoblastik Akut (LLA) pada Citra Mikroskopis Sel Darah Menggunakan Arsitektur EfficientNet-B3 dengan Dataset Seimbang," *Jurnal Locus Penelitian dan Pengabdian*, vol. 4, no. 6, pp. 1–16, 2025, doi: 10.58344/locus.v4i6.4321.
- [5] S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, p. 76, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>
- [6] Y. Mardianto, T. Dewi, and P. Risma, "Analisis Klasifikasi Kematangan Buah Tomat dengan Pendekatan Transfer Learning Model EfficientNet," *Jurnal Techno Bahari*, vol. 11, 2024, [Online]. Available: <https://www.kaggle.com/datasets/nexuswho/tomatofruits/da>
- [7] H. Istiqomah, P. Purwono, and R. Ardianto, "Prediksi Kanker Darah Menggunakan Metode Convolutional Neural Network," *Jurnal Ilmu Komputer dan Informatika*, vol. 4, no. 1, pp. 51–60, 2024, doi: 10.54082/jiki.156.
- [8] A. , & G. R. Gupta, "ALL Challenge dataset of ISBI 2019 [Data set]," *The Cancer Imaging Archive*, 2019.
- [9] S. A. Panggabean and S. Susilawati, "Penerapan Arsitektur EfficientNet dalam Model CNN untuk Optimalisasi Klasifikasi Gambar Fashion pada Dataset Fashion MNIST," *INCODING: Journal of Informatics and Computer Science Engineering*, vol. 5, no. 2, pp. 281–292, Oct. 2025, doi: 10.34007/incoding.v5i2.980.
- [10] M. Bagas Tirta Kusuma, "Klasifikasi Oral Squamous Cell Carcinoma (OSCC) Menggunakan Convolutional Neural Network (CNN) dengan ResNet-152," *Universitas Multimedia Nusantara*, 2025.
- [11] Karsito and S. Susanti, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia," *Jurnal Teknologi Pelita Bangsa*, vol. 9, pp. 43–48, 2019.
- [12] Ž. Vujović, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.
- [13] S. T. Sabu and A. M. Thomas, "AI-Powered Product Recommendation And Purchase Probability Analysis," in *Proc. Nat. Conf. Emerg. Comput. Appl. (NCECA) 2025*, Amal Jyothi College of Engineering, vol. 1, no. 6, 2025, doi: 10.5281/zenodo.15486634.
- [14] "Personalizing E-Commerce by Optimizing LLMs for Tailored Product Recommendations," in *2025 IEEE Int. Conf. Big Data Artif. Intell. (BDAI)*, IEEE, 2025.
- [15] K.-Z. Liao and H. A. Santoso, "The Role of Generative AI in e-Commerce Recommender Systems: Methods, Trends and Insights," *J. Inform. Web Eng.*, vol. 4, no. 3, pp. 35–63, 2025, doi: 10.33093/jiwe.2025.4.3.3.