

**IMPLEMENTASI EFFICIENTNETV2M DENGAN WEIGHTED  
ENSEMBLE NETWORKS UNTUK KLASIFIKASI CITRA  
ACUTE LYMPHOBLASTIC LEUKEMIA**

**PROPOSAL SKRIPSI**



**Oleh:**

**Glendy Hernandez Putra Mahardika Gunantoro**

**220411100076**

**Dosen Pembimbing 1 : Prof. Dr. Arif Muntasa, S.Si., M.T.**

**Dosen Pembimbing 2 : Husni, S.Kom., M.T.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**BANGKALAN**

**2025**

## ABSTRAK

Diagnosis Acute Lymphoblastic Leukemia (ALL) secara konvensional melalui pemeriksaan mikroskopis oleh ahli patologi merupakan proses yang memakan waktu, sangat bergantung pada keahlian, dan rentan terhadap subjektivitas antar pengamat. Seiring perkembangan teknologi, deep learning menawarkan solusi menjanjikan untuk diagnosis berbantuan komputer, namun kebanyakan penelitian membahas performa arsitektur tunggal dan hanya sedikit yang membahas tentang *ensemble learning*. Penelitian ini mengusulkan pendekatan dua tahap: pertama, investigasi sistematis terhadap tiga skenario LR (0.001, 0.0001, dan 0.00001) untuk melatih model EfficientNetV2-M; kedua, implementasi metode Weighted Ensemble of Networks (WEN). Kebaruan utama penelitian ini terletak pada investigasi dua skenario WEN: WEN Inter-LR (menggabungkan model-model dari 5-Fold di dalam satu skenario LR) dan WEN Intra-LR (menggabungkan model-model "juara" antar skenario LR). Untuk setiap skenario WEN, penelitian ini akan menguji dan membandingkan tiga metrik berbeda sebagai dasar penentuan bobot: Akurasi, F1-Score, dan Cohen's Kappa, yang dieksplorasi karena keandalannya menangani ketidakseimbangan kelas. Data yang digunakan adalah dataset publik C-NMC 2019, dan seluruh model dilatih menggunakan 5-Fold Cross Validation untuk evaluasi yang robust. Penelitian ini berfokus pada analisis komparatif untuk mengidentifikasi LR optimal serta skenario ensemble dan skema pembobotan WEN yang diharapkan mampu menghasilkan performa klasifikasi ALL paling optimal, melampaui kinerja model tunggal terbaik.

**Kata Kunci:** *Acute Lymphoblastic Leukemia, EfficientNetV2, Weighted Ensemble of Networks, Klasifikasi Citra*

## DAFTAR ISI

ABSTRAK .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iv
DAFTAR TABLE .....	v
BAB 1 PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	5
1.2.1    Permasalahan.....	5
1.2.2    Solusi Permasalahan .....	5
1.2.3    Pertanyaan Penelitian .....	6
1.3    Tujuan dan Manfaat .....	7
1.4    Batasan Masalah .....	8
1.5    Metode Usulan.....	9
1.6    Sistematika Penulisan.....	11
BAB II KAJIAN PUSTAKA .....	13
2.1    Penelitian Terdahulu .....	13
2.2    Acute Lymphoblastic Leukemia (ALL).....	17
2.3    Klasifikasi .....	18
2.4    Convolutional Neural Network (CNN) .....	18
2.4.1    Convolution Layer.....	19
2.4.2    Non-Linear Activation Functions (Swish) .....	21
2.4.3    Pooling Layer .....	22
2.4.4    Fully Connected Layer .....	23
2.5    EfficientNet.....	26
2.6    Optimizer .....	33
2.7    Area Under Curve.....	34

2.7.1	Kurva Receiver Operating Characteristic (ROC) .....	35
2.8	Weighted Ensemble of Network .....	36
2.8.1	Pembobotan Berdasarkan F1-Score.....	38
2.8.2	Pembobotan Berdasarkan Akurasi.....	38
2.8.3	Pembobotan Berdasarkan Cohen's Kappa .....	38
2.9	Cross Validation.....	39
2.10	Evaluasi Kinerja Sistem.....	40
2.10.1	Confusion Matrix.....	41
2.10.2	Accuracy .....	41
2.10.3	Precision .....	42
2.10.4	Recall .....	42
2.10.5	F1-Score.....	42
2.10.6	Cohen's Kappa .....	43
2.11	Learning Rate .....	44
2.12	Class Weight .....	45
BAB III METODE USULAN.....		48
3.1	Dataset .....	48
3.2	Rancangan Arsitektur Penelitian .....	50
3.3	Rancangan Sistem Pelatihan Model.....	51
3.4	Rancangan Weighted Ensemble of Networks (WEN).....	54
3.5	Skenario Uji Coba .....	57
DAFTAR PUSTAKA .....		60

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Convolutional Neural Network.....	19
<b>Gambar 2. 2</b> Proses Konvolusi .....	19
<b>Gambar 2. 3</b> Grafik fungsi aktivasi swish .....	22
<b>Gambar 2. 4</b> Pooling Layer.....	23
<b>Gambar 2. 5</b> Fully Connected Layer .....	24
<b>Gambar 2. 6</b> Grafik penskalaan depth dan resolution terhadap akurasi dan kompleksitas.....	29
<b>Gambar 2. 7</b> Arsitektur EfficientNet.....	31
<b>Gambar 2. 8</b> Grafik Performa EfficientNet .....	32
<b>Gambar 2. 9</b> K-Fold Cross Validation .....	40
<b>Gambar 3. 1</b> File Gambar ALL.....	49
<b>Gambar 3. 2</b> Rancangan Arsitektur Penelitian.....	50
<b>Gambar 3. 3</b> Rancangan Sistem Pelatihan Model .....	52
<b>Gambar 3. 4</b> Rancangan Weighted Ensemble of Networks (Intra-LR) .....	56

## DAFTAR TABLE

<b>Table 2. 1</b> Penelitian Terkait .....	16
<b>Table 2. 2</b> Confusion Matrix.....	41
<b>Table 3. 1</b> Komposisi Struktur Asli Dataset .....	48
<b>Table 3. 2</b> Skenario Uji Coba Pelatihan.....	58
<b>Table 3. 3</b> Skenario Emsemble Pembobotan .....	58
<b>Table 3. 1</b> Komposisi Struktur Asli Dataset .....	42
<b>Table 3. 2</b> Skenario Uji Coba Pelatihan.....	51
<b>Table 3. 3</b> Skenario Emsemble Pembobotan .....	51

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kanker darah merupakan salah satu tantangan terbesar dalam dunia medis global, dengan *Acute Lymphoblastic Leukemia* (ALL) menjadi jenis kanker darah yang paling umum didiagnosis pada anak-anak. Patologi penyakit ini ditandai oleh proliferasi abnormal, atau pertumbuhan tak terkendali, dari sel limfoblas (sel darah putih yang belum matang) di dalam sumsum tulang[1]. Akumulasi sel-sel abnormal ini mendesak sel-sel darah sehat—termasuk sel darah merah, trombosit, dan sel darah putih matang—yang pada akhirnya dapat menyebar secara progresif ke organ vital lain seperti hati, limpa, dan sistem saraf pusat. Diagnosis dini menjadi faktor yang sangat krusial; keberhasilan terapi dan angka harapan hidup pasien berkorelasi langsung dengan seberapa cepat penyakit ini terdeteksi dan ditangani[2].

Meskipun urgensinya sangat tinggi, standar untuk diagnosis ALL secara konvensional masih mengandalkan metode yang padat karya dan memiliki keterbatasan signifikan. Metode ini adalah pemeriksaan mikroskopis apusan darah tepi atau sumsum tulang oleh ahli patologi. Proses ini secara inheren sangat bergantung pada faktor manusia: keahlian, pengalaman, dan bahkan kondisi fisik serta kelelahan (*fatigue*) pemeriksa. Ketergantungan ini melahirkan risiko subjektivitas dan variabilitas hasil antar-pengamat (*inter-observer variability*)[3]. Selain itu, proses manual ini memakan waktu yang berharga. Dalam skenario di mana laboratorium menerima ratusan sampel, keterbatasan ini dapat menyebabkan keterlambatan diagnosis. Keterbatasan inilah yang mendorong perlunya inovasi teknologi untuk mengembangkan sistem diagnosis yang lebih objektif, konsisten, cepat, dan akurat.

Perkembangan pesat dalam bidang kecerdasan buatan (Artificial Intelligence), khususnya deep learning, menawarkan solusi yang sangat menjanjikan untuk mengatasi tantangan tersebut. Sistem diagnosis berbantuan komputer (*Computer-Aided Diagnosis - CAD*) yang ditenagai oleh *deep learning* telah menunjukkan kapasitas yang luar biasa dalam menganalisis citra medis[4]. Arsitektur Convolutional

Neural Network (CNN) secara khusus telah terbukti sangat efektif dalam tugas klasifikasi citra medis, karena kemampuannya untuk secara otomatis mempelajari fitur-fitur hierarkis dari data visual, dari tepi sederhana hingga pola selular yang kompleks, tanpa perlu rekayasa fitur (*feature engineering*) manual[5].

Dalam beberapa tahun terakhir, berbagai penelitian telah berupaya menerapkan beragam arsitektur CNN untuk klasifikasi citra ALL. Banyak peneliti fokus pada penggunaan model tunggal yang dioptimalkan. Sebagai contoh, penelitian oleh Amreen Batool[6], menunjukkan hasil yang sangat impresif dengan menggunakan arsitektur EfficientNet-B3, yang dimodifikasi dengan depthwise separable convolutions untuk efisiensi, dan berhasil mencapai akurasi 99.31% pada dataset C-NMC 2019. Demikian pula, Muhammad Zakir Ullah[7], mengintegrasikan modul Efficient Channel Attention (ECA) ke dalam arsitektur VGG16 untuk membantu model fokus pada fitur relevan, dan berhasil mencapai akurasi rata-rata 91.1%.

Meskipun model tunggal ini menunjukkan akurasi yang tinggi, ketergantungan pada satu arsitektur memiliki kelemahan, yaitu rentan terhadap overfitting dan bias terhadap set fitur spesifik yang dipelajari oleh arsitektur tersebut. Untuk mengatasi ini, pendekatan lain adalah dengan melakukan optimasi hyperparameter secara ekstensif atau memodifikasi arsitektur dasarnya. Penelitian oleh Kevyn Alifian Hernanda Wibowo[5], membuktikan hal ini dengan menerapkan Optimasi Bayesian pada EfficientNetV2M, yang berhasil meningkatkan akurasi dari 87.61% menjadi 91.37%. Pendekatan yang lebih radikal dilakukan oleh Arif Muntasar[8], yang memodifikasi EfficientNetV2-S dengan Hybrid Kernel dan Enkapsulasi Jaringan, dan berhasil mencapai F1-Score 97.26%.

Pendekatan-pendekatan tersebut, meskipun sangat kuat, menunjukkan bahwa performa model tunggal masih dapat ditingkatkan, seringkali dengan menambah kompleksitas pada arsitektur atau proses tuning. Pendekatan ketiga, yang juga sangat kompleks, adalah metode hibrida yang menggabungkan *feature fusion* dari beberapa model CNN sebelum dimasukkan ke classifier machine learning tradisional. Ibrahim Abdulrab Ahmed[9], menerapkan ini dengan menggabungkan fitur dari DenseNet121,

ResNet50, dan MobileNet, yang kemudian diklasifikasikan oleh Random Forest, dan berhasil mencapai akurasi 98.8% pada dataset C-NMC 2019. Kelemahan dari metode fusi fitur semacam ini adalah kompleksitas dalam implementasi dan tingginya kebutuhan komputasi untuk mengekstraksi fitur dari berbagai model secara bersamaan.

Keterbatasan dan kompleksitas dari berbagai pendekatan ini—baik mengandalkan model tunggal yang dioptimalkan maupun fusi fitur yang rumit—menegaskan sebuah fakta: mengandalkan satu model tunggal, terlepas dari seberapa canggih arsitekturnya, seringkali belum cukup untuk menangkap seluruh kompleksitas dan variasi fitur pada citra sel leukemia yang heterogen[10]. Setiap arsitektur (seperti VGG, ResNet, atau EfficientNet) memiliki bias dan cara pandangnya sendiri dalam mengekstraksi pola. Di sinilah filosofi ensemble learning menawarkan alternatif yang elegan dan kuat.

Teknik ensemble learning digunakan untuk menggabungkan prediksi dari beberapa model, sehingga menghasilkan keputusan akhir yang lebih robust dan akurat. Daripada bertaruh pada satu ahli saja, ensemble learning bertindak seperti panel ahli di mana keputusan kolektif lebih dipercaya[11]. Salah satu pendekatan ensemble yang efektif adalah Weighted Ensemble of Networks (WEN), yang memberikan bobot kontribusi yang berbeda untuk setiap model berdasarkan kinerjanya. Dengan cara ini, model yang lebih unggul akan memiliki pengaruh lebih besar terhadap hasil klasifikasi akhir[12].

Penelitian ini mengusulkan implementasi arsitektur EfficientNetV2 sebagai base model. Arsitektur ini dipilih karena merupakan evolusi dari EfficientNet yang revolusioner, yang tidak hanya berfokus pada akurasi tetapi juga pada efisiensi komputasi [5]. EfficientNetV2 menggunakan compound scaling untuk menyeimbangkan kedalaman, lebar, dan resolusi jaringan secara cerdas, serta memanfaatkan blok MBConv yang efisien. Secara spesifik, penelitian ini akan berfokus pada satu varian, yaitu EfficientNetV2-M (*Medium*), yang dipilih karena menawarkan keseimbangan optimal antara kedalaman arsitektur dan efisiensi komputasi.

Namun, untuk mencapai performa puncak dari arsitektur tersebut, optimasi hyperparameter menjadi aspek yang krusial. Arsitektur yang kuat saja tidak cukup; performa model deep learning sangat bergantung pada konfigurasi pelatihannya. Salah satu tantangan utama dalam pelatihan adalah overfitting (di mana model terlalu menghafal data latih dan gagal pada data baru), yang pengelolaannya sangat dipengaruhi oleh pemilihan hyperparameter yang tepat.

Selanjutnya, penelitian ini akan melakukan investigasi sistematis terhadap hyperparameter yang paling berpengaruh, yaitu *Learning Rate* (LR). Model EfficientNetV2-M akan dilatih dan dievaluasi dalam tiga skenario LR yang berbeda: 0.001, 0.0001, dan 0.00001, untuk menemukan laju pembelajaran yang paling optimal.

Setelah mengeksplorasi performa model tunggal, penelitian ini akan melangkah lebih jauh dengan menerapkan *Weighted Ensemble of Networks* (WEN). Di sinilah letak kebaruan (novelty) utama dari penelitian ini. Berbeda dari penelitian sebelumnya yang menggabungkan arsitektur berbeda[12], penelitian ini mengusulkan strategi ensemble yang didasarkan pada hasil optimasi *hyperparameter*. Penelitian ini akan mengeksplorasi dua skenario ensemble utama: pertama, Ensemble Inter-LR, yang menggabungkan model-model dari 5-Fold Cross Validation yang dilatih di dalam satu skema LR yang sama (misalnya, 5 model dari LR 0.001 digabungkan); dan kedua, Ensemble Intra-LR, yang menggabungkan model-model terbaik yang mewakili setiap skema LR (model terbaik dari LR 0.001, 0.0001, dan 0.00001 digabungkan).

Pendekatan WEN ini, yang divalidasi oleh Chayan Mondal [12], akan diinvestigasi lebih lanjut. Temuan Mondal akan dijadikan titik awal untuk menguji kembali skema pembobotan mana yang paling efektif. Oleh karena itu, kebaruan penelitian ini adalah menerapkan dan memvalidasi strategi pembobotan WEN (Akurasi, F1-Score, dan Cohen's Kappa) secara spesifik pada ensemble model EfficientNetV2-M yang dihasilkan dari skenario pelatihan *hyperparameter* (LR) yang berbeda.

Hipotesis penelitian ini adalah: (1) Akan terdapat perbedaan kinerja yang signifikan antara model tunggal yang dilatih dengan LR 0.001, 0.0001, dan 0.00001.

(2) Model Weighted Ensemble (baik Inter-LR maupun Intra-LR) akan memiliki performa yang lebih tinggi daripada performa model tunggal terbaik sekalipun. (3) Strategi ensemble Intra-LR berpotensi menjadi model paling unggul karena menggabungkan "keahlian" dari model-model yang dilatih dengan cara yang fundamental berbeda. (4) Akan terdapat perbedaan kinerja yang signifikan di antara ketiga skema pembobotan (Akurasi, F1, Kappa), yang akan mengidentifikasi strategi ensemble paling optimal.

Melalui penerapan metode ini pada dataset publik C-NMC 2019, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan sistem CAD untuk diagnosis leukemia. Hasil dari penelitian ini berpotensi meningkatkan akurasi, objektivitas, dan efisiensi diagnosis. Lebih jauh lagi, penelitian ini dapat menjadi referensi penting bagi pengembangan metode ensemble learning yang dikombinasikan dengan strategi optimasi hyperparameter untuk aplikasi klasifikasi citra medis lainnya.

## **1.2 Rumusan Masalah**

### **1.2.1 Permasalahan**

Bagaimana merancang dan mengimplementasikan model deep learning yang efektif dan optimal sebagai metode diagnosis Acute Lymphoblastic Leukemia (ALL) pada citra mikroskopis sel darah. Tantangan utamanya terletak pada penentuan konfigurasi hyperparameter yang tepat, seperti Learning Rate, untuk melatih arsitektur tunggal (EfficientNetV2-M) secara optimal. Selain itu, terdapat permasalahan lebih lanjut apakah penggabungan beberapa model melalui metode ensemble dapat memberikan peningkatan performa yang lebih signifikan dibandingkan model tunggal terbaik sekalipun.

### **1.2.2 Solusi Permasalahan**

Solusi yang diusulkan dalam penelitian ini adalah mengimplementasikan sistem klasifikasi deep learning melalui pendekatan investigasi multi-tahap untuk menemukan konfigurasi paling optimal. Pendekatan ini terbagi dalam dua tahapan utama:

1. Pelatihan Model Tunggal dan Optimasi Hyperparameter: Tahap pertama adalah melatih dan mengevaluasi arsitektur Convolutional Neural Network (CNN) EfficientNetV2-M. Pada tahap ini, dilakukan investigasi sistematis terhadap hyperparameter *Learning Rate* (LR) dengan menguji tiga skenario berbeda (0.001, 0.0001, dan 0.00001) untuk menemukan performa model tunggal terbaik.
2. Implementasi Weighted Ensemble of Networks (WEN): Tahap kedua adalah menggabungkan model-model yang telah dilatih pada tahap pertama. Metode WEN akan diinvestigasi dalam dua skenario (Inter-LR dan Intra-LR) dan diuji menggunakan tiga skema pembobotan berbeda (berdasarkan Akurasi, F1-Score, dan Cohen's Kappa) untuk menentukan strategi ensemble yang paling unggul.

Dengan membandingkan performa dari berbagai skenario hyperparameter (LR) pada model tunggal dan berbagai skenario pada model ensemble (WEN), penelitian ini bertujuan mengidentifikasi arsitektur dan strategi pelatihan yang paling akurat serta andal untuk diagnosis ALL.

### **1.2.3 Pertanyaan Penelitian**

Berdasarkan permasalahan yang telah dijelaskan di atas, dapat dirumuskan pertanyaan dalam penelitian sebagai berikut:

1. Bagaimana pengaruh perbedaan Learning Rate (0.001, 0.0001, dan 0.00001) terhadap performa arsitektur tunggal EfficientNetV2-M dalam melakukan klasifikasi citra Acute Lymphoblastic Leukemia?
2. Bagaimana performa dari penerapan metode Weighted Ensemble of Networks (WEN), baik skenario Inter-LR (gabungan 5-Fold dalam satu LR) maupun skenario Intra-LR (gabungan model terbaik antar LR) dalam klasifikasi citra ALL?
3. Manakah di antara skema pembobotan berdasarkan metrik Akurasi, F1-Score, dan Cohen's Kappa yang menghasilkan performa paling optimal pada model

Weighted Ensemble of Networks (Inter-LR dan Intra-LR) untuk kasus klasifikasi citra ALL?

### **1.3 Tujuan dan Manfaat**

#### A. Tujuan Penelitian

Sejalan dengan rumusan masalah yang telah dipaparkan, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengevaluasi dan membandingkan performa arsitektur tunggal EfficientNetV2-M yang dilatih menggunakan tiga skema *Learning Rate* (0.001, 0.0001, dan 0.00001) untuk klasifikasi citra *Acute Lymphoblastic Leukemia*.
2. Menganalisis performa yang dihasilkan oleh metode *Weighted Ensemble of Networks* (WEN), baik skenario Inter-LR (gabungan 5-Fold dalam satu LR) maupun skenario Intra-LR (gabungan model terbaik antar LR), dibandingkan dengan performa arsitektur tunggal terbaik.
3. Mengidentifikasi skema pembobotan yang paling optimal untuk model *Weighted Ensemble of Networks* (Inter-LR dan Intra-LR) dengan membandingkan penggunaan metrik Akurasi, F1-Score, dan Cohen's Kappa sebagai dasar penentuan bobot.

#### B. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Bidang Medis: Memberikan kontribusi terhadap pengembangan sistem diagnosis komputer yang akurat dan objektif untuk *Acute Lymphoblastic Leukemia*, sehingga dapat membantu ahli patologi dalam mempercepat dan meningkatkan keandalan proses diagnosis.
2. Bagi Ilmu Pengetahuan: Memperkaya khazanah penelitian di bidang deep learning, khususnya terkait penerapan arsitektur EfficientNetV2-M dan

efektivitas metode *Weighted Ensemble of Networks* dengan berbagai skema pembobotan untuk klasifikasi citra medis.

3. Bagi Peneliti Selanjutnya: Hasil penelitian ini dapat menjadi referensi yang kuat dan landasan bagi pengembangan metode deteksi penyakit berbasis citra medis yang lebih lanjut, baik dalam domain hematologi maupun bidang medis lainnya.
4. Bagi Penulis: Meningkatkan kompetensi dan pengalaman praktis dalam merancang, mengimplementasikan, dan mengevaluasi sistem cerdas berbasis deep learning untuk menyelesaikan permasalahan nyata di sektor kesehatan.

#### **1.4 Batasan Masalah**

Agar penelitian ini lebih terarah dan fokus pada tujuan yang telah ditetapkan, maka diberikan batasan-batasan masalah sebagai berikut:

1. Objek penelitian ini terbatas pada klasifikasi citra untuk penyakit *Acute Lymphoblastic Leukemia* (ALL). Jenis leukemia lain atau subtipe dari ALL tidak termasuk dalam lingkup kajian penelitian ini.
2. Dataset yang digunakan dalam penelitian adalah dataset publik C-NMC 2019. Penelitian ini tidak melakukan proses pengumpulan atau pengambilan data primer dari rumah sakit maupun laboratorium.
3. Arsitektur model yang digunakan dibatasi pada satu varian dari EfficientNetV2, yaitu Medium (M).
4. Metode ensemble yang diterapkan terbatas pada *Weighted Ensemble of Networks* (WEN), dengan skema pembobotan berdasarkan hasil validasi metrik Akurasi, F1-Score, dan Cohen's Kappa.
5. Proses pelatihan model menggunakan pendekatan transfer learning dengan memanfaatkan bobot pra-terlatih (*pre-trained weights*) dari dataset ImageNet.

6. Evaluasi performa model dibatasi pada metrik kuantitatif yang mencakup akurasi, presisi, recall, F1-score, Cohen's Kappa, serta visualisasi kurva *Receiver Operating Characteristic* (ROC) dan nilai *Area Under Curve* (AUC). Aspek lain seperti kecepatan komputasi dan penggunaan memori tidak dievaluasi.
7. Penelitian ini berfokus pada domain teknis klasifikasi citra dan tidak membahas aspek klinis yang lebih dalam, seperti penentuan stadium penyakit atau prognosis pasien. Tahapan implementasi atau deployment model ke dalam sistem klinis juga tidak termasuk dalam cakupan penelitian.
8. Eksperimen hyperparameter terbatas pada investigasi tiga skema *Learning Rate* (0.001, 0.0001, dan 0.00001). *Hyperparameter* lain seperti optimizer (Adam), jumlah epoch, dan batch size diatur sebagai nilai konstan selama proses pelatihan.

## 1.5 Metode Usulan

Penelitian ini dilaksanakan secara sistematis melalui beberapa tahapan utama yang dirancang untuk mencapai tujuan yang telah ditetapkan. Adapun tahapan metodologi penelitian yang dilakukan adalah sebagai berikut:

### 1. Studi Literatur

Tahap awal adalah melakukan kajian mendalam terhadap literatur dan penelitian terdahulu yang relevan. Fokus studi mencakup teori klasifikasi citra medis, arsitektur deep learning khususnya EfficientNetV2, konsep ensemble learning, dan implementasi metode Weighted Ensemble of Networks (WEN). Tahap ini bertujuan untuk membangun landasan teoretis yang kuat dan mengidentifikasi posisi penelitian ini terhadap riset sebelumnya.

### 2. Pengumpulan dan Persiapan Data

Dataset yang digunakan adalah C-NMC 2019 yang berisi citra sel darah putih. Data tersebut kemudian melalui tahap persiapan (preprocessing) yang meliputi

proses normalisasi nilai piksel dan augmentasi data untuk meningkatkan variasi dan mencegah overfitting. Selanjutnya, dataset dibagi menjadi tiga bagian: data latih, data validasi, dan data uji.

### 3. Perancangan dan Pelatihan Model Tunggal

Merancang dan mengimplementasikan model klasifikasi menggunakan arsitektur EfficientNetV2-M. Pelatihan model akan dilakukan dalam tiga skenario hyperparameter yang berbeda berdasarkan *Learning Rate* (0.001, 0.0001, dan 0.00001). Setiap skenario model akan dilatih menggunakan pendekatan transfer learning dengan bobot pra-terlatih dari ImageNet, melalui proses fine-tuning untuk adaptasi pada tugas klasifikasi biner (*Acute Lymphoblastic Leukemia* vs. Normal)

### 4. Implementasi Weighted Ensemble of Networks (WEN)

Mengimplementasikan dua skenario ensemble menggunakan metode WEN. Skenario pertama adalah WEN Inter-LR, yang menggabungkan prediksi dari 5 model hasil cross-validation yang dilatih dalam satu skenario *Learning Rate* yang sama. Skenario kedua adalah WEN Intra-LR, yang menggabungkan prediksi dari tiga model terbaik (satu model terbaik mewakili setiap skenario *Learning Rate*). Pada tahap ini, dilakukan perhitungan bobot untuk setiap model berdasarkan performa individunya (Akurasi, F1-Score, dan Cohen's Kappa) yang diukur pada data evaluasi.

### 5. Pengujian dan Evaluasi Model

Melakukan pengujian terhadap model-model tunggal (hasil dari 3 skenario LR) dan model-model ensemble (Inter-LR dan Intra-LR) menggunakan data uji yang belum pernah dilihat sebelumnya. Kinerja model dievaluasi secara kuantitatif menggunakan metrik akurasi, presisi, recall, F1-score, Cohen's Kappa, serta visualisasi kurva Receiver Operating Characteristic (ROC) dan nilai Area Under Curve (AUC) untuk membandingkan efektivitas antara model

tunggal dan ensemble, serta untuk menentukan skema pembobotan WEN yang paling optimal

#### 6. Analisis dan Penarikan Kesimpulan

Menganalisis hasil evaluasi secara komprehensif untuk menjawab rumusan masalah. Hasil analisis kemudian disajikan, dan ditarik kesimpulan mengenai performa model serta efektivitas metode yang diusulkan.

#### 7. Penyusunan Laporan

Mendokumentasikan seluruh proses, hasil, dan analisis penelitian secara sistematis ke dalam format laporan skripsi, yang diakhiri dengan simpulan dan saran untuk penelitian di masa mendatang.

### **1.6 Sistematika Penulisan**

Proposal penelitian ini disusun dengan sistematika penulisan yang terdiri dari tiga bab utama, yaitu sebagai berikut:

#### **Bab I : Pendahuluan**

Bab ini merupakan bagian pengantar yang menguraikan konteks penelitian. Isinya mencakup latar belakang masalah, rumusan masalah, tujuan dan manfaat yang ingin dicapai, batasan masalah, gambaran umum metodologi penelitian, serta sistematika penulisan proposal.

#### **Bab II : Tinajuan Pustaka**

Bab ini menyajikan landasan teoretis yang menjadi dasar penelitian. Pembahasan di dalamnya meliputi teori mengenai *Acute Lymphoblastic Leukemia* (ALL), pengolahan citra, *deep learning*, arsitektur EfficientNetV2, dan konsep *ensemble learning* dengan metode *Weighted Ensemble of Networks* (WEN). Selain itu, bab ini juga mengulas penelitian-penelitian relevan sebelumnya sebagai referensi.

#### **Bab III : Metode Usulan**

Bab ini menjelaskan secara rinci rancangan dan langkah-langkah yang akan dilakukan untuk menjawab rumusan masalah. Uraian di dalamnya mencakup deskripsi dataset yang akan digunakan, tahapan persiapan data (*preprocessing*), perancangan arsitektur model, skenario pelatihan dan pengujian, serta metrik evaluasi yang akan digunakan untuk mengukur kinerja model

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Dalam penelitian yang dilakukan oleh Chayan Mondal, dkk[12], diangkat permasalahan mengenai diagnosis Acute Lymphoblastic Leukemia (ALL) yang secara tradisional memakan waktu, memerlukan keahlian tinggi, dan rentan terhadap subjektivitas antar pengamat. Tujuan penelitian ini adalah untuk mengotomatisasi proses diagnosis menggunakan beberapa arsitektur Deep Convolutional Neural Networks (CNNs) dan meningkatkan akurasi klasifikasi melalui metode Weighted Ensemble of Networks (WEN). Penelitian ini menggunakan dataset publik C-NMC-2019, dengan melakukan beberapa tahap pra-pemrosesan seperti center-cropping, random oversampling untuk mengatasi ketidakseimbangan kelas, dan augmentasi data. Beberapa model CNN yang diuji meliputi Xception, VGG-16, DenseNet-121, MobileNet, dan InceptionResNet-V2. Hasil penelitian menunjukkan bahwa model tunggal dengan kinerja terbaik adalah Xception yang mencapai akurasi 85.9%. Namun, metode ensemble yang diusulkan, yaitu WEN yang bobotnya dihitung berdasarkan metrik Cohen's Kappa, berhasil mencapai kinerja yang lebih unggul dengan akurasi 89.8% dan F1-score 89.7%, membuktikan bahwa penggabungan model dapat meningkatkan keandalan diagnosis secara signifikan.

Penelitian yang dilakukan oleh Diana Baby, dkk[11]. menangani permasalahan identifikasi leukemia limfositik secara manual yang memakan waktu dan rentan terhadap kesalahan. Tujuan dari penelitian ini adalah untuk mengembangkan sebuah sistem deteksi otomatis yang efisien. Dalam metodologinya, penelitian ini memanfaatkan delapan variasi arsitektur EfficientNet (mulai dari B0 hingga B7) untuk melakukan ekstraksi fitur dari citra sel darah yang berasal dari dataset ALL-IDB. Fitur-fitur yang telah diekstraksi tersebut kemudian diklasifikasikan menggunakan ensemble voting classifier yang menggabungkan tiga model machine learning: Support Vector Machine (SVM), Random Forest, dan Logistic Regression. Hasil dari sistem yang diusulkan ini sangat baik, dengan pencapaian akurasi klasifikasi keseluruhan sebesar

98.5%. Meskipun demikian, penelitian ini mencatat bahwa akurasi untuk mendeteksi sel leukemia secara spesifik (97%) masih perlu ditingkatkan lebih lanjut.

Penelitian oleh Kevyn Alifian Hernanda Wibowo, dkk[5]. mengatasi kesulitan dalam diagnosis dini leukemia yang memakan waktu dan mahal, serta kurang optimalnya model deep learning yang ada. Tujuan penelitian ini adalah untuk meningkatkan akurasi klasifikasi dengan membangun model CNN berbasis arsitektur EfficientNetV2M dan menerapkan Optimasi Bayesian untuk menyetel hyperparameter. Menggunakan dataset citra sel darah C-NMC 2019, metode yang diusulkan berhasil menunjukkan peningkatan kinerja yang signifikan. Hasilnya, model yang telah dioptimalkan dengan Optimasi Bayesian berhasil mencapai akurasi 91.37%, jauh lebih tinggi dibandingkan model dasar tanpa optimasi yang hanya mencapai 87.61%.

Penelitian yang dilakukan oleh Arif Muntas, dkk[8]. Berfokus untuk mengatasi masalah waktu dan biaya yang signifikan dalam diagnosis Leukemia Limfoblastik Akut (ALL), serta keterbatasan pada model deep learning seperti EfficientNetV2-S yang rentan terhadap overfitting. Metode yang diusulkan adalah memodifikasi arsitektur EfficientNetV2-S dengan pendekatan Hybrid Kernel yang mengintegrasikan tiga jenis konvolusi (reguler, dilatasi, dan depth-wise), serta menerapkan metode Enkapsulasi Jaringan untuk menangkap fitur spasial yang lebih kaya. Dengan menggunakan dataset C-NMC-2019 yang telah diseimbangkan, model dievaluasi melalui 5-fold cross-validation. Hasil terbaik dicapai dengan learning rate 0.0001 dan 10 epoch, menghasilkan akurasi rata-rata 97.25%, presisi 96.77%, recall 97.75%, dan F1-Score 97.26%.

Penelitian oleh Amreen Batool, dkk[6]. Membahas tantangan dalam identifikasi manual Acute Lymphoblastic Leukemia (ALL) yang memakan waktu dan rumit. Tujuannya adalah mengembangkan model deep learning yang ringan namun kuat untuk klasifikasi sel darah. Mereka mengusulkan penggunaan arsitektur EfficientNet-B3, yang dioptimalkan dengan depthwise separable convolutions untuk mengurangi jumlah parameter dan meningkatkan efisiensi komputasi. Dengan menggunakan dataset C\_NMC\_2019 dan dataset ALL multi-kelas, model ini melalui tahap pra-pemrosesan

dan augmentasi data yang ekstensif. Hasilnya sangat mengesankan, di mana model ini mencapai akurasi 99.31% dan F1-score 99.35% untuk klasifikasi biner (ALL vs. Normal), serta akurasi 96.81% untuk klasifikasi multi-kelas.

Penelitian oleh Ibrahim Abdulrab Ahmed, dkk[9]. Menangani masalah diagnosis manual Acute Lymphoblastic Leukemia (ALL) yang rawan kesalahan, terutama pada tahap awal penyakit di mana sel sehat dan terinfeksi sulit dibedakan. Tujuannya adalah untuk mengembangkan sistem diagnosis dini yang sangat akurat. Metodologi yang diusulkan adalah pendekatan hibrida yang kompleks, dimulai dari peningkatan kualitas gambar, segmentasi sel darah putih (WBC) menggunakan active contour, lalu fusi fitur yang diekstraksi dari tiga model CNN pra-terlatih (DenseNet121, ResNet50, dan MobileNet). Fitur-fitur ini kemudian direduksi dimensinya menggunakan PCA sebelum diklasifikasikan dengan Random Forest (RF) atau XGBoost. Pada dataset C-NMC 2019 dan ALL-IDB2, sistem ini menunjukkan kinerja yang luar biasa. Model hibrida yang menggabungkan fusi fitur dari ketiga CNN dengan classifier RF berhasil mencapai akurasi 98.8% pada dataset C-NMC 2019 dan akurasi sempurna 100% pada dataset ALL-IDB2.

Penelitian oleh Muhammad Zakir Ullah, dkk[7]. Mengatasi permasalahan diagnosis Acute Lymphoblastic Leukemia (ALL) yang sulit dan memakan waktu dengan metode tradisional yang rawan kesalahan. Tujuan dari penelitian ini adalah untuk menciptakan sistem klasifikasi otomatis yang akurat untuk membedakan antara sel normal dan sel ALL. Metode yang diusulkan adalah model berbasis CNN yang menggunakan VGG16 sebagai arsitektur dasar, namun diintegrasikan dengan modul Efficient Channel Attention (ECA) untuk membantu model fokus pada fitur-fitur yang paling relevan dari citra sel. Dengan menggunakan dataset C-NMC 2019, hasil penelitian menunjukkan bahwa penambahan modul ECA secara signifikan meningkatkan performa. Model yang diusulkan berhasil mencapai akurasi rata-rata 91.1%, sebuah peningkatan yang jelas dari 83.9% yang dicapai oleh VGG16 standar.

Penelitian oleh Larissa Ferreira Rodrigues[3] mengatasi masalah subjektivitas dan waktu yang lama dalam diagnosis manual Acute Lymphoblastic Leukemia (ALL),

serta kesulitan dalam menemukan hyperparameter optimal untuk model deep learning. Tujuan penelitian ini adalah mengembangkan model hibrida yang mengoptimalkan arsitektur ResNet-50V2 menggunakan Algoritma Genetik (GA) untuk menyetel hyperparameter kunci seperti dropout, learning rate, dan momentum. Dengan menggunakan dataset ALL-IDB2, pendekatan ini berhasil mencapai kinerja yang sangat tinggi, dengan akurasi rata-rata 98.46%, presisi 98.55%, dan F1-score 98.46%. Hasil ini menunjukkan peningkatan signifikan dibandingkan model tanpa optimasi dan membuktikan efektivitas GA dalam menemukan konfigurasi hyperparameter yang optimal untuk klasifikasi ALL.

**Table 2. 1** Penelitian Terkait

No	Penelitian, Tahun	Permasalahan, Dataset	Model/Solusi	Hasil
1	Chayan Mondal, dkk.[12], 2021	Acute Lymphoblastic Leukemia, C-NMC 2019	Menggunakan beberapa model CNN (Xception, VGG-16, dll.) dan metode Weighted Ensemble of Networks (WEN) berbasis kinerja.	Model ensemble (WEN <sup>kappa</sup> ) mencapai akurasi 89.8%, mengungguli model tunggal terbaik (Xception) dengan akurasi 85.9%.
2	Diana Baby, dkk.[11], 2022	Acute Lymphoblastic Leukemia, ALL-IDB	Ekstraksi fitur menggunakan EfficientNet (B0-B7) dan klasifikasi dengan ensemble voting classifier (SVM, RF, LR).	Mencapai akurasi klasifikasi sebesar 98.5%.
3	Kevyn Alifian Hernanda Wibowo, dkk[5]., 2024	Acute Lymphoblastic Leukemia, C-NMC 2019	CNN dengan arsitektur EfficientNetV2M dengan Optimasi Bayesian.	Meningkatkan akurasi dari 87.61% menjadi 91.37%.
4	Arif Muntasar, dkk[8]., 2024	Acute Lymphoblastic Leukemia, C-NMC 2019	Modifikasi EfficientNetV2-S dengan pendekatan Hybrid Kernel dan Enkapsulasi Jaringan.	Mencapai akurasi rata-rata 97.25% dan F1-Score 97.26%.

5	Amreen Batool, dkk[6].., 2023	Acute Lymphoblastic Leukemia, C-NMC 2019 dan ALL-IDB	EfficientNet-B3 dengan depthwise separable convolutions.	Akurasi 99.31% (klasifikasi biner) dan 96.81% (multi-class).
6	Ibrahim Abdulrab Ahmed, dkk.[9], 2023	Acute Lymphoblastic Leukemia, C-NMC 2019 dan ALL-IDB	Fusi fitur 3 CNN (DenseNet121, ResNet50, MobileNet), PCA, klasifikasi RF atau XGBoost.	Akurasi 98.8% (C-NMC 2019) diperoleh dari fusi fitur DenseNet121, ResNet50, MobileNet, dan classifier Random Forest. Akurasi 100% (ALL-IDB2): Diperoleh dari semua metode fusi fitur baik dengan classifier RF maupun XGBoost.
7	Muhammad Zakir Ullah, dkk[7].., 2021	Acute Lymphoblastic Leukemia, C-NMC 2019	CNN berbasis VGG16 dengan modul Efficient Channel Attention (ECA).	Mencapai akurasi 91.1%.
8	Larissa Ferreira Rodrigues[3], 2022	Acute Lymphoblastic Leukemia, ALL-IDB	Optimasi hyperparameter ResNet-50V2 menggunakan Algoritma Genetik (GA).	Mencapai akurasi 98.46%

## 2.2 Acute Lymphoblastic Leukemia (ALL)

Leukemia merupakan salah satu jenis kanker darah yang ditandai dengan produksi abnormal sel darah putih di dalam sumsum tulang. Pertumbuhan sel darah putih yang tidak terkendali ini dapat mengganggu fungsi normal tubuh dalam melawan infeksi, mengangkut oksigen, serta pembekuan darah. Terdapat beberapa tipe leukemia, salah satunya adalah ALL, yaitu leukemia yang berkembang cepat dan menyerang sel limfosit. ALL banyak ditemukan pada anak-anak, meskipun juga dapat menyerang orang dewasa[11].

Diagnosis yang cepat dan akurat sangat penting untuk menentukan penanganan lebih lanjut. Dalam praktik klinis, diagnosis leukemia biasanya dilakukan melalui pemeriksaan laboratorium seperti morfologi darah tepi dan analisis sumsum tulang, namun proses tersebut memerlukan waktu serta keahlian medis. Oleh karena itu, diperlukan pendekatan berbasis computer-aided diagnosis (CAD) yang memanfaatkan citra sel darah untuk mempercepat proses deteksi dan klasifikasi ALL[10].

### **2.3 Klasifikasi**

Klasifikasi merupakan proses pengelompokan objek atau data ke dalam kelas tertentu berdasarkan fitur atau karakteristik yang dimilikinya. Dalam konteks machine learning, klasifikasi dilakukan dengan membangun model yang dilatih dari data berlabel, sehingga mampu memprediksi kelas dari data baru. Metode klasifikasi banyak digunakan dalam berbagai bidang, seperti pengenalan wajah, deteksi objek, analisis teks, dan juga bidang medis[13].

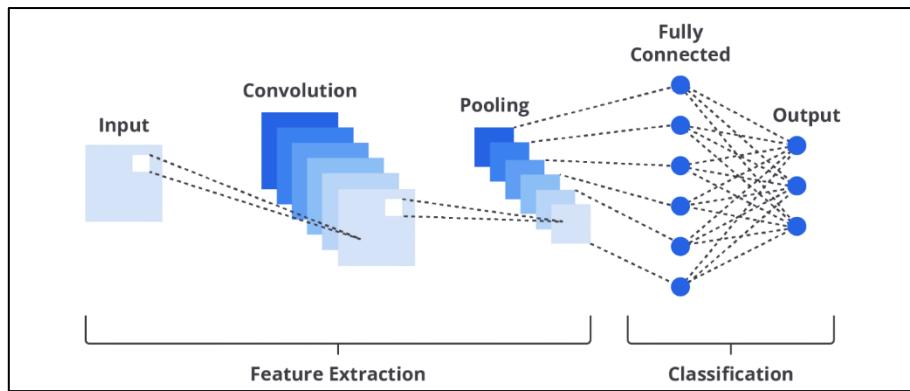
### **2.4 Convolutional Neural Network (CNN)**

Convolutional Neural Network (CNN) adalah salah satu arsitektur jaringan saraf tiruan yang banyak digunakan dalam pemrosesan citra, khususnya pada tugas deteksi dan pengenalan objek. CNN terinspirasi dari mekanisme sistem visual pada makhluk hidup, di mana proses pengenalan pola dilakukan secara bertingkat dari yang sederhana hingga kompleks[14]. CNN mampu secara otomatis mengekstraksi dan memahami hierarki fitur spasial dari citra, sehingga sangat efektif dalam pemrosesan data berbentuk grid, seperti gambar[15].

CNN termasuk ke dalam metode supervised learning, yaitu pembelajaran dengan data yang telah memiliki label, sehingga jaringan dapat dilatih untuk melakukan klasifikasi secara akurat. Dalam praktiknya, CNN bekerja dengan melakukan operasi konvolusi melalui beberapa lapisan yang berjalan paralel. Setiap neuron direpresentasikan dalam bentuk dua dimensi, sehingga CNN sangat sesuai untuk data berupa citra

Secara umum, arsitektur CNN terdiri dari tiga lapisan utama, yaitu convolutional layer, pooling layer, dan fully connected layer.

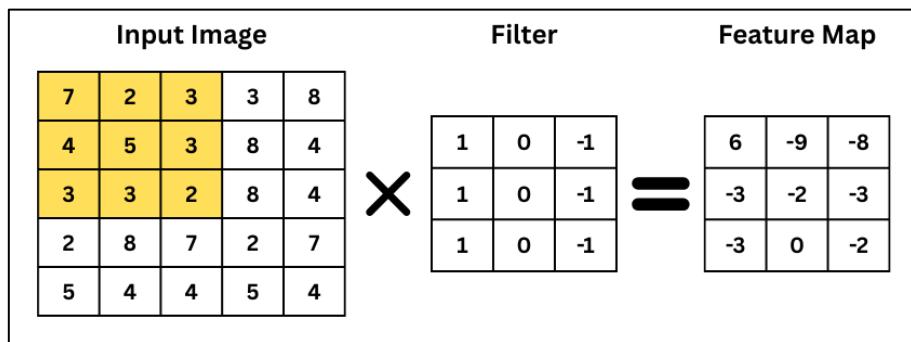
**Gambar 2. 1 Convolutional Neural Network**



#### 2.4.1 Convolution Layer

Convolution layer adalah lapisan utama dalam CNN yang berfungsi melakukan ekstraksi fitur dari citra masukan. Dalam banyak arsitektur, convolutional layer dapat ditumpuk lebih dari satu, dengan tujuan agar jaringan mampu mengekstraksi fitur yang semakin kompleks pada tiap lapisan[16].

**Gambar 2. 2 Proses Konvolusi**



Proses konvolusi dilakukan dengan cara mengalikan nilai piksel citra dengan kernel atau filter berukuran tertentu, kemudian menjumlahkannya. Sebagai contoh, jika sebuah filter berukuran 3x3 dikalikan dengan potongan citra berukuran 3x3, maka hasil perkalian elemen-elemen tersebut dijumlahkan, dan nilainya akan

ditempatkan pada posisi yang sesuai di feature map. Proses ini diulangi hingga seluruh bagian citra telah diproses.

Secara matematis, proses konvolusi antara matriks masukan dengan matriks filter/kernel hingga menghasilkan feature map dapat dinyatakan dalam persamaan berikut.

$$A[i, j] = \sum_{v=-k}^k \sum_{u=-k}^k H[u, v] K[i - u, j - v] + b_j \quad (2.1)$$

Keterangan:

- A: feature map
- H: matriks masukan
- K: matriks filter
- b: bias
- u: kolom
- v: baris
- k:  $\left[\frac{k}{2}\right]$

Dalam convolutional layer terdapat beberapa parameter penting yang memengaruhi hasil ekstraksi fitur, yaitu:

1. **Kernel/Filter:** matriks berukuran kecil dengan nilai tertentu yang digunakan untuk mengekstraksi pola dari citra masukan.
2. **Stride:** menentukan seberapa jauh filter bergeser pada setiap langkah konvolusi. Nilai stride yang kecil akan menghasilkan *feature map* lebih detail, namun meningkatkan beban komputasi.
3. **Padding:** penambahan piksel di tepi citra masukan untuk mempertahankan ukuran output agar tetap sesuai dengan ukuran input.

Ukuran keluaran (output feature map) dari lapisan konvolusi dapat dihitung dengan persamaan berikut:

$$W_{out} = \frac{W_{in} - K + 2P}{S} + 1 \quad (2.2)$$

$$H_{out} = \frac{H_{in} - K + 2P}{S} + 1 \quad (2.3)$$

Keterangan:

- W: Lebar dari matriks masukan yang berukuran WxH
- H: Tinggi dari matriks masukan yang berukuran WxH
- K: Ukuran matriks filter yang berukuran nxn
- P: Padding
- S: Stride

#### 2.4.2 Non-Linear Activation Functions (Swish)

Fungsi aktivasi Swish merupakan salah satu fungsi aktivasi non-linear yang digunakan pada arsitektur deep learning modern, termasuk EfficientNet. Swish memiliki karakteristik lebih halus dibandingkan ReLU dan dapat mengurangi masalah dying neurons yang sering terjadi pada ReLU[17]. Fungsi ini bekerja dengan cara mengalikan nilai masukan dengan fungsi sigmoid dari masukan tersebut. Persamaan Swish ditunjukkan pada Persamaan berikut:

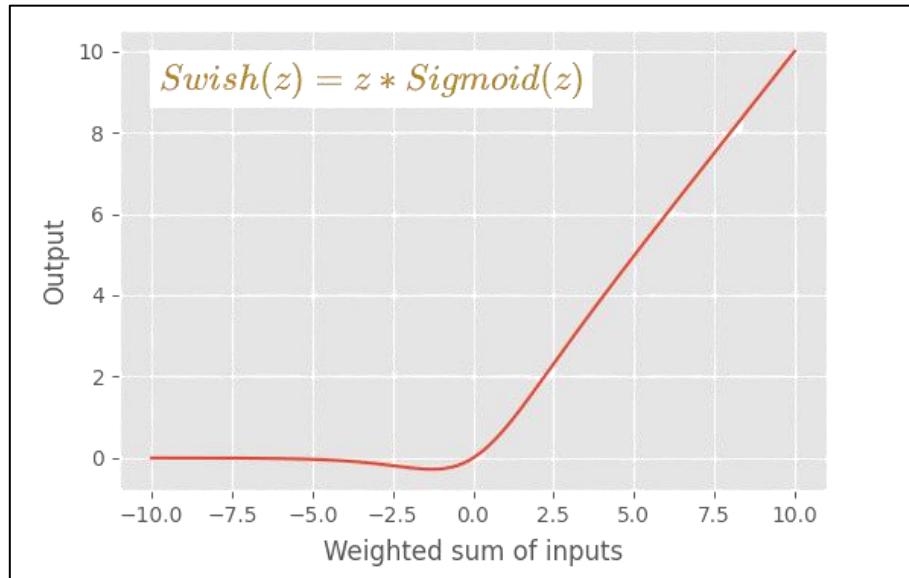
$$f(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \quad (2.4)$$

Keterangan:

- $\sigma(x)$ : fungsi sigmoid
- $x$ : nilai input yang masuk ke fungsi aktivasi (output dari layer sebelumnya).

Dimana  $x$  adalah nilai masukkan. Untuk grafik dari Swish aktivasi dapat dilihat pada grafik berikut:

**Gambar 2. 3** Grafik fungsi aktivasi swish

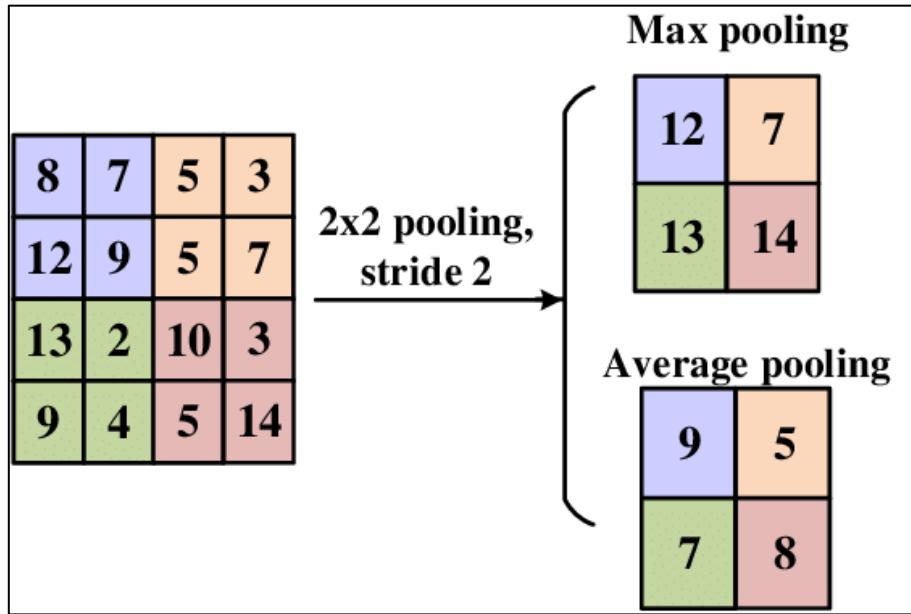


#### 2.4.3 Pooling Layer

Pooling Layer merupakan salah satu lapisan pada CNN yang berfungsi untuk mengurangi ukuran spasial citra dan jumlah parameter pada feature maps. Lapisan ini sering disebut juga sebagai subsampling layer dan biasanya ditempatkan setelah convolutional layer[13]. Tujuan utama pooling adalah menekan kompleksitas komputasi dengan tetap mempertahankan informasi penting serta mengurangi risiko overfitting[18].

Pooling bekerja dengan memilih nilai tertentu dari area pada feature map. Terdapat dua jenis pooling yang umum digunakan, yaitu average pooling, yang mengambil nilai rata-rata dari area tertentu, dan max pooling, yang mengambil nilai maksimum dari area tersebut[15].

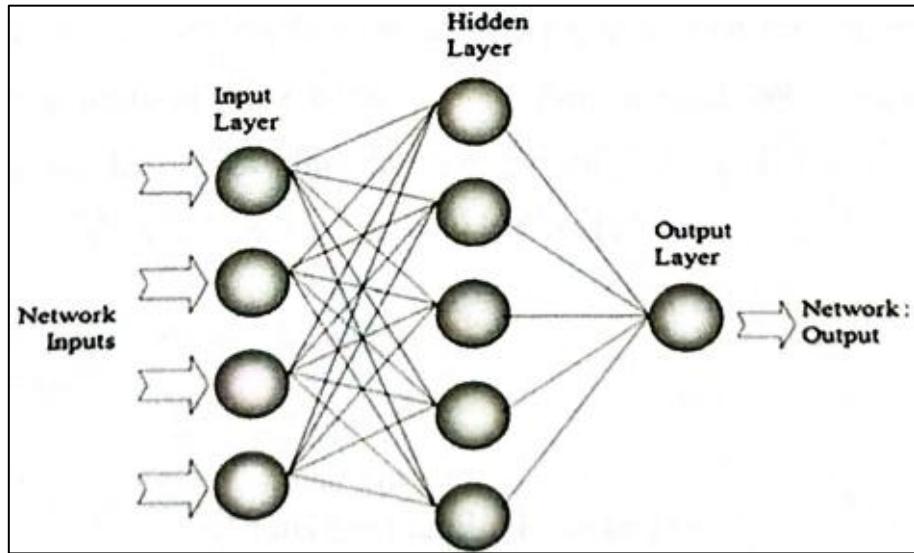
**Gambar 2. 4 Pooling Layer**



#### 2.4.4 Fully Connected Layer

Fully Connected Layer merupakan lapisan pada CNN yang tersusun dari neuron-neuron yang saling terhubung secara penuh. Lapisan ini berfungsi untuk mengubah hasil feature maps dari proses ekstraksi fitur yang masih berbentuk matriks menjadi vektor melalui proses flattening atau reshape[13], sehingga dapat dijadikan masukan bagi neuron pada FC layer.

**Gambar 2.5 Fully Connected Layer**



Berbeda dengan convolutional layer yang hanya terhubung pada area tertentu dari citra masukan, pada FC layer setiap neuron terhubung dengan seluruh neuron pada lapisan sebelumnya. Arsitektur FC layer terdiri dari tiga bagian utama, yaitu input layer, hidden layer, dan output layer. Berikut adalah operasi dari FC layer

$$y_j = \sigma \left( \sum_{i=1}^n x_i w_{ij} + b_j \right) \quad (2.5)$$

Keterangan:

- $y_j$  : output hidden layer
- $x_i$  : nilai input ke-i
- $w_{ij}$  : nilai bobot dari input ke-i menuju hidden ke-j
- $b_j$  : bias menuju hidden ke-j
- $n$  : jumlah neuron/fitur input
- $\sigma$  : fungsi aktivasi output hidden layer

Pada FC Layer, setiap nilai input ( $x_i$ ) akan dikalikan dengan bobot ( $w_{ij}$ ) yang telah ditentukan, kemudian ditambahkan dengan nilai bias ( $b_j$ ).

Hasil perhitungan tersebut menjadi masukan bagi neuron pada hidden layer ( $y_j$ ), yang selanjutnya dilewatkan melalui fungsi aktivasi tertentu untuk menghasilkan keluaran yang lebih representatif.

Selanjutnya, hidden layer melakukan proses perhitungan dengan cara mengalikan nilai pada lapisan input dengan bobot yang telah diinisialisasi, kemudian menambahkan hasilnya dengan nilai bias. Hubungan matematis pada perhitungan hidden layer dapat dituliskan dalam persamaan berikut.

$$net_k = \sum_{j=1}^m y_j w_{jk} + b_k \quad (2.6)$$

Keterangan:

- $net_k$  : output layer
- $y_j$  : hasil hidden layer ke-j
- $w_{jk}$  : bobot dari hidden layer ke-j menuju output layer ke-k
- $b_k$  : bias menuju hidden layer ke-k
- $m$  : jumlah neuron pada hidden layer

Setelah melalui proses pada hidden layer, tahap berikutnya adalah menghasilkan output layer. Pada lapisan ini dilakukan aktivasi untuk seluruh neuron dengan menggunakan fungsi aktivasi sigmoid. Fungsi sigmoid akan memetakan nilai keluaran ke dalam rentang 0 hingga 1, sehingga sesuai digunakan untuk kasus klasifikasi biner. Berikut adalah persamaan sigmoid.

$$y_j = \frac{1}{1 + e^{-x}} \quad (2.7)$$

Keterangan:

- $y_j$  : probabilitas
- $x$  : output layer

Selanjutnya, nilai keluaran tersebut digunakan dalam perhitungan loss function dengan metode binary cross entropy, yang umum dipakai ketika jumlah kelas pada dataset hanya terdiri dari dua label

$$loss = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (2.8)$$

Keterangan:

- $y_i$  : nilai target
- $\hat{y}_i$  : nilai prediksi
- $N$  : total data

Untuk menentukan hasil akhir prediksi, digunakan persamaan klasifikasi biner yang memutuskan apakah suatu data masuk ke kelas 0 atau kelas 1 berdasarkan nilai probabilitas yang dihasilkan.

$$y_j = \{0 \quad (\hat{y}_j \leq 0.5), 1 \quad (\hat{y}_j \geq 0.5)\} \quad (2.9)$$

Dimana  $\hat{y}_i$  adalah hasil dari kelas prediksi

## 2.5 EfficientNet

EfficientNet pertama kali diperkenalkan oleh Tan dan Le pada tahun 2019 dari Google Research melalui publikasi berjudul EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks[19]. Kemunculan arsitektur ini didorong oleh kebutuhan akan model jaringan saraf konvolusional (CNN) yang tidak hanya akurat, tetapi juga efisien dalam penggunaan sumber daya komputasi. Pada saat itu, tren pengembangan arsitektur CNN lebih banyak berfokus pada peningkatan kedalaman (depth) atau lebar (width) jaringan untuk memperoleh akurasi lebih tinggi. Pendekatan tersebut seringkali menghasilkan model dengan jumlah parameter yang sangat besar dan waktu komputasi yang lama. EfficientNet menghadirkan solusi dengan memperkenalkan konsep compound scaling yang memungkinkan proses penskalaan model dilakukan secara seimbang pada tiga dimensi utama, yaitu kedalaman, lebar, dan resolusi input citra[20].

Pada tahun 2021, generasi penerusnya yaitu EfficientNetV2 diperkenalkan dengan sejumlah perbaikan penting. Jika EfficientNet generasi pertama menekankan pada efisiensi ukuran model, maka EfficientNetV2 hadir untuk menjawab kebutuhan akan proses pelatihan yang lebih cepat dan efisiensi parameter yang lebih tinggi[21]. Perkembangan ini penting mengingat skala data yang semakin besar serta kebutuhan pelatihan model dalam waktu yang lebih singkat.

### 2.5.1 Konsep Compound Scaling

EfficientNet merupakan arsitektur convolutional neural network yang dibangun berdasarkan konsep compound scaling. Konsep ini menjawab permasalahan klasik mengenai trade-off antara ukuran model, akurasi, dan efisiensi komputasi. Gagasan utama compound scaling adalah melakukan penskalaan secara bersamaan pada tiga dimensi penting jaringan saraf, yaitu width, depth, dan resolution[22].

- Width: Pensakalan lebar merujuk pada jumlah channel pada setiap lapisan jaringan. Dengan menambah lebar jaringan, model dapat menangkap pola dan fitur yang lebih kompleks sehingga meningkatkan akurasi. Sebaliknya, pengurangan lebar menghasilkan model yang lebih ringan, cocok untuk lingkungan dengan keterbatasan sumber daya
- Depth: Pensakalan kedalaman terkait dengan jumlah total lapisan dalam jaringan. Model yang lebih dalam mampu merepresentasikan data secara lebih kompleks, namun juga membutuhkan sumber daya komputasi yang lebih besar. Model yang lebih dangkal lebih efisien, tetapi sering kali mengorbankan akurasi
- Resolution: Pensakalan resolusi mengacu pada ukuran citra masukan. Gambar dengan resolusi tinggi membawa lebih banyak detail sehingga berpotensi meningkatkan performa. Namun, hal ini juga menuntut memori dan daya komputasi lebih besar. Sebaliknya, gambar resolusi rendah lebih ringan diproses, tetapi kehilangan detail halus.

Salah satu kekuatan utama EfficientNet adalah kemampuannya menyeimbangkan ketiga dimensi ini dengan pendekatan yang sistematis. Dimulai dari sebuah baseline model, peneliti melakukan pencarian grid untuk menemukan kombinasi optimal antara width, depth, dan resolution. Pencarian ini dipandu oleh sebuah koefisien skala komposit yang disebut  $\phi$  (*compound coefficient*)[23], yang secara seragam mengatur kompleksitas dan kebutuhan sumber daya model.

Rumus penskalaan komposit dapat dituliskan sebagai berikut:

$$d = \alpha^\phi, \quad w = \beta^\phi, \quad r = \gamma^\phi \quad (2.10)$$

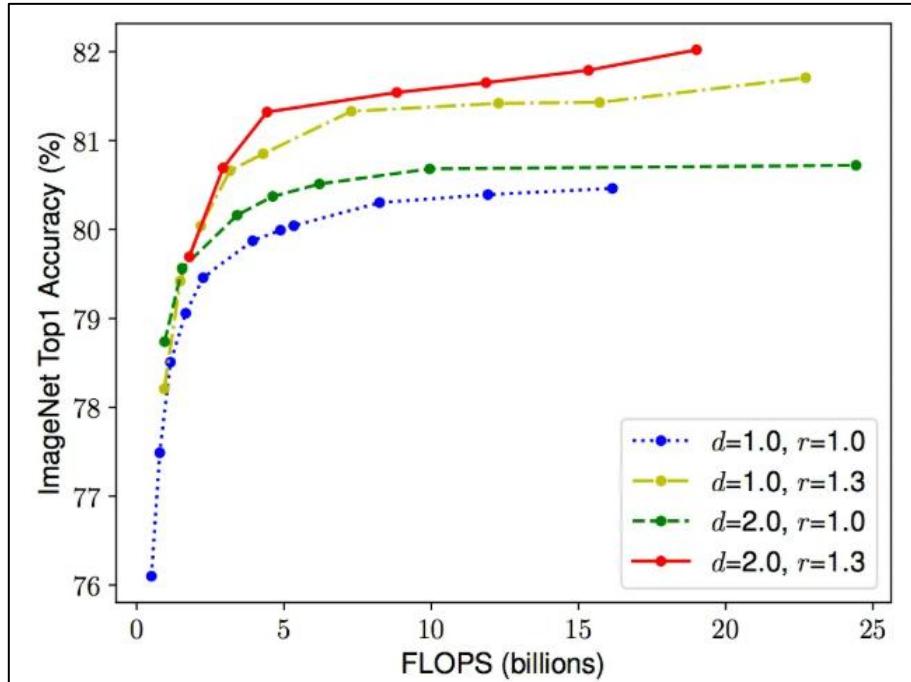
Dengan ketentuan bahwa  $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

Keterangan:

- $d$  : kedalaman jaringan (depth).
- $w$  : lebar jaringan (width).
- $r$  : resolusi citra masukan (resolution).
- $\phi$  : koefisien skala komposit (compound coefficient).
- $\alpha, \beta, \gamma$  : konstanta skala hasil pencarian grid.

Pada penelitian asli, diperoleh nilai empiris terbaik yaitu  $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$ . Nilai ini digunakan untuk membangun model *baseline* EfficientNet-B0, sedangkan variasi B1–B7 diperoleh dengan meningkatkan  $\phi$

**Gambar 2.6** Grafik penskalaan depth dan resolution terhadap akurasi dan kompleksitas



Gambar 2.6 menunjukkan dampak penskalaan terhadap akurasi dan kompleksitas model. Terlihat bahwa kombinasi penskalaan pada depth dan resolution (garis merah:  $d=2, r=1.3$ ) menghasilkan akurasi lebih tinggi dibanding hanya menskalakan salah satu dimensi saja.

### 2.5.2 Cara Kerja Compound Scaling

Proses compound scaling diawali dengan sebuah baseline model yang berfungsi sebagai titik awal. Model baseline ini biasanya merupakan jaringan saraf berukuran sedang yang telah menunjukkan performa baik pada suatu tugas, tetapi belum sepenuhnya optimal dalam hal efisiensi komputasi[22].

Langkah berikutnya adalah memperkenalkan koefisien komposit \phi, yaitu sebuah parameter skalar yang ditentukan pengguna. Nilai \phi mengatur seberapa besar penskalaan dilakukan terhadap dimensi width, depth, dan resolution[22].

Dengan mengubah nilai  $\phi$ , kompleksitas dan kebutuhan sumber daya model dapat dikendalikan secara terukur.

Setelah itu, ketiga dimensi diskalakan secara seimbang dan terkoordinasi:

- Width Scaling: jumlah channel jaringan diperbesar dengan mengalikan faktor skala yang ditentukan oleh  $\beta^\phi$
- Depth Scaling: jumlah lapisan jaringan ditingkatkan sesuai faktor  $\beta^\phi$
- Resolution Scaling: ukuran input citra diperbesar sesuai faktor  $\gamma^\phi$

Nilai konstanta  $\alpha, \beta, \gamma$  ditentukan melalui proses pencarian grid empiris untuk menemukan kombinasi terbaik yang memberikan keseimbangan antara akurasi dan efisiensi. Setelah ketiga faktor penskalaan ini diterapkan pada *baseline model*, dihasilkanlah varian EfficientNet dengan kompleksitas tertentu yang dikendalikan oleh nilai  $\phi$ .

Dengan demikian, pengguna dapat memilih varian EfficientNet yang sesuai dengan kebutuhan dan keterbatasan perangkat keras. Nilai  $\phi$  yang lebih kecil menghasilkan model ringan dan hemat sumber daya, sedangkan nilai  $\phi$  yang lebih besar menghasilkan model lebih kuat namun membutuhkan komputasi lebih tinggi.

Metode penskalaan terkoordinasi inilah yang membuat EfficientNet mampu mencapai performa *state-of-the-art* pada berbagai tugas visi komputer, sekaligus tetap adaptif terhadap beragam keterbatasan perangkat keras.

### 2.5.3 Arsitektur EfficientNet

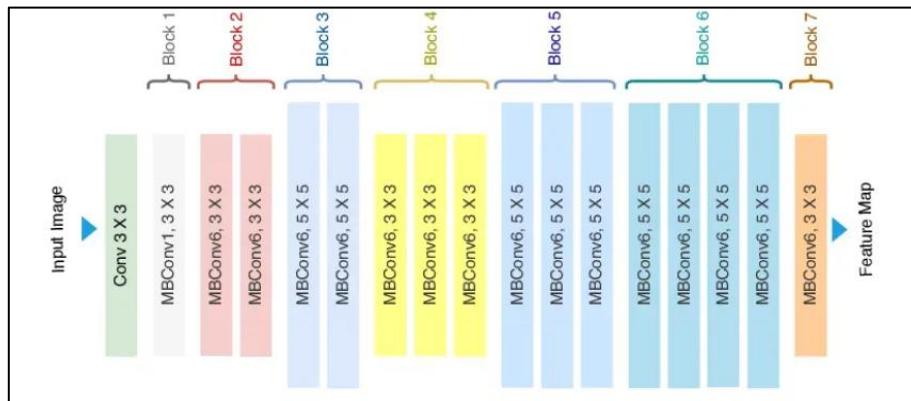
EfficientNet menggunakan Mobile Inverted Bottleneck (MBConv) sebagai blok dasar arsitektur, yang merupakan kombinasi dari depth-wise separable convolution dan inverted residual block. Pendekatan ini diadaptasi dari MobileNetV2 dengan beberapa modifikasi untuk meningkatkan efisiensi dan kemampuan representasi[19]. Selain itu, arsitektur ini juga mengintegrasikan

Squeeze-and-Excitation (SE) block untuk memperkuat performa dengan cara menekankan fitur penting dan menekan fitur yang kurang relevan.

MBConv diawali dengan depth-wise convolution untuk mengekstraksi informasi spasial dari setiap saluran, kemudian dilanjutkan dengan point-wise convolution (konvolusi 1x1) yang memperluas jumlah saluran sehingga dapat merepresentasikan pola lebih kompleks. Setelah itu, jumlah saluran dikembalikan ke ukuran semula menggunakan konvolusi 1x1 lain. Desain bottleneck ini memungkinkan jaringan untuk belajar secara efisien tanpa meningkatkan kompleksitas model secara signifikan[23].

SE block berperan penting dalam memberi bobot pada saluran tertentu agar model dapat fokus pada informasi yang paling relevan. Mekanismenya dimulai dengan global average pooling untuk mereduksi dimensi spasial menjadi satu vektor, kemudian melewati dua fully connected layer yang menghasilkan bobot atensi. Bobot ini dikalikan dengan fitur asli sehingga informasi penting lebih menonjol dalam proses pembelajaran.

**Gambar 2. 7** Arsitektur EfficientNet



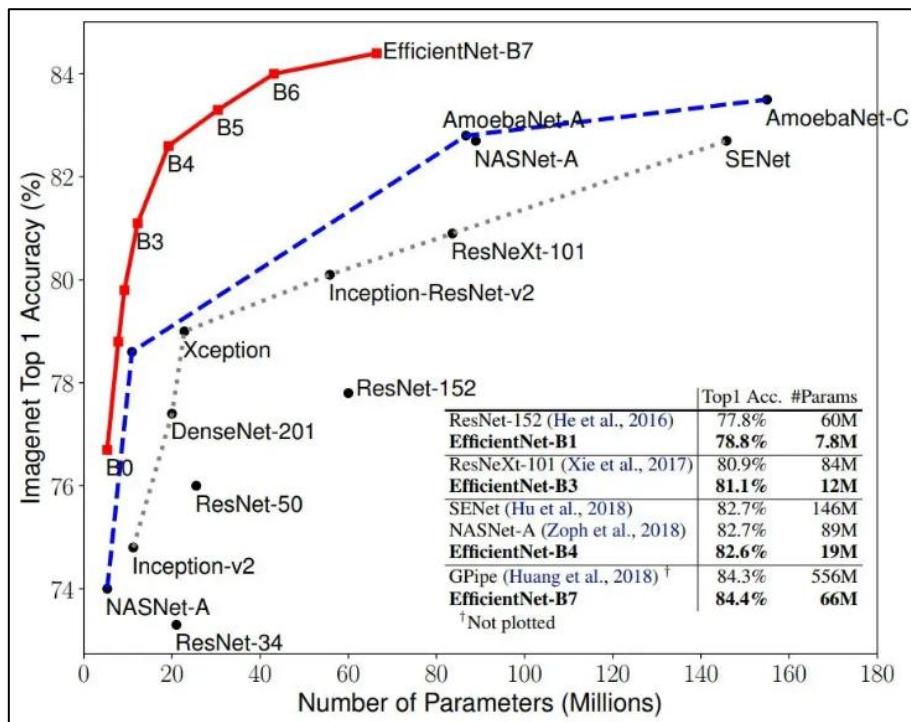
EfficientNet tersedia dalam berbagai varian seperti B0, hingga B7, dan V2. Setiap varian memiliki skala berbeda yang menyeimbangkan ukuran model, akurasi, dan kebutuhan komputasi. Dengan adanya pilihan varian ini, pengguna dapat menyesuaikan arsitektur EfficientNet sesuai kebutuhan spesifik, baik untuk

perangkat dengan sumber daya terbatas maupun untuk sistem dengan kapasitas komputasi tinggi[23].

#### 2.5.4 Performa EfficientNet

Performa EfficientNet dapat dilihat dari perbandingan dengan berbagai arsitektur jaringan saraf konvolusional yang lebih dahulu dikembangkan. Gambar di bawah menunjukkan kurva EfficientNet yang ditandai dengan garis merah. Pada sumbu horizontal ditampilkan jumlah parameter model, sedangkan sumbu vertikal menggambarkan tingkat akurasi pada dataset ImageNet Top-1[24].

**Gambar 2. 8** Grafik Performa EfficientNet



Dari Gambar 2.8 terlihat bahwa EfficientNet mampu memberikan kinerja yang unggul dengan jumlah parameter yang jauh lebih efisien. Sebagai contoh, model EfficientNet-B7 berhasil mencapai akurasi sebesar 84,4% dengan hanya menggunakan 66 juta parameter. Sebagai perbandingan, GPipe yang sebelumnya merupakan state-of-the-art membutuhkan 556 juta parameter untuk mencapai akurasi 84,3%. Walaupun perbedaannya hanya sekitar 0,1% dari sisi akurasi,

efisiensi parameter yang delapan kali lebih kecil membuat EfficientNet jauh lebih ringan dan cepat untuk diterapkan pada skenario dunia nyata[24].

Kemampuan ini menjadikan EfficientNet tidak hanya unggul dalam hal akurasi, tetapi juga sangat praktis untuk berbagai kebutuhan industri. Dengan kebutuhan komputasi yang lebih rendah, model ini dapat diimplementasikan secara lebih luas, baik pada perangkat dengan sumber daya terbatas maupun pada sistem berskala besar.

## 2.6 Optimizer

Optimizer merupakan salah satu komponen penting dalam proses pembelajaran jaringan saraf tiruan. Fungsinya adalah mengatur pembaruan bobot agar model dapat mencapai konvergensi lebih cepat dan efisien. Setiap algoritma optimasi umumnya memerlukan pengaturan nilai learning rate, yaitu parameter yang menentukan seberapa besar langkah pembaruan bobot pada setiap iterasi[25].

Pada penelitian ini digunakan optimizer Adam (Adaptive Moment Estimation), yaitu algoritma optimasi yang mengombinasikan kelebihan dari Momentum dan RMSprop. Adam bekerja dengan menghitung rata-rata gradien (momen pertama) dan rata-rata kuadrat gradien (momen kedua), lalu melakukan penyesuaian adaptif terhadap learning rate pada setiap parameter[25]. Dengan demikian, setiap bobot memiliki kecepatan pembaruan yang berbeda-beda sesuai dengan kondisi gradien yang dihadapinya. Proses pembaruan bobot dengan Adam dapat dijelaskan sebagai berikut:

- Menghitung estimasi momen pertama dan kedua:

$$m_t \leftarrow \rho_1 m_{t-1} + (1 - \rho_1) g_t \quad (2.11)$$

$$u_t \leftarrow \rho_2 u_{t-1} + (1 - \rho_2) g_t^2 \quad (2.12)$$

- Menghitung gradien bobot dan bias:

$$g_{t(weight)} \leftarrow (t - y) \cdot y(1 - y) \cdot x \quad (2.13)$$

$$g_{t(bias)} \leftarrow (t - y) \cdot y(1 - y) \cdot 1 \quad (2.14)$$

- Melakukan bias correction:

$$\widehat{m}_t \leftarrow \frac{m_t}{1 - \rho_1^t} \quad (2.15)$$

$$\widehat{u}_t \leftarrow \frac{u_t}{1 - \rho_2^t} \quad (2.16)$$

d. Menghitung bobot baru:

$$w_t = w_{t-1} - \sigma \frac{\widehat{m}_t}{\sqrt{\widehat{u}_t} - \varepsilon} \quad (2.17)$$

Keterangan:

- $m_t$  : rata-rata gradien pada waktu  $t$  (momen pertama)
- $u_t$  : rata-rata kuadrat gradien pada waktu  $t$  (momen kedua)
- $\widehat{m}_t, \widehat{u}_t$  : nilai gradien baru setelah bias correction
- $g_t$  : nilai gradien pada waktu  $t$
- $w_t$  : bobot pada waktu  $t$
- $\rho_1, \rho_2$  : konstanta decay untuk momen pertama dan kedua
- $\sigma$  : learning rate
- $\varepsilon$  : konstanta kecil ( $10^{-8}$ ) untuk menghindari pembagian dengan nol

Dengan mekanisme tersebut, Adam mampu menyeimbangkan kecepatan konvergensi dan kestabilan pembaruan bobot. Selain itu, Adam relatif lebih mudah digunakan karena sudah memiliki nilai default hyperparameter yang umumnya bekerja baik ( $\rho_1 = 0.9, \rho_2 = 0.999, \varepsilon = 10^{-8}$ )

## 2.7 Area Under Curve

*Area Under the Curve* (AUC) adalah salah satu metrik evaluasi yang fundamental untuk mengukur performa model klasifikasi biner. Secara harfiah, AUC mengukur keseluruhan area di bawah kurva *Receiver Operating Characteristic* (ROC). Nilai AUC merepresentasikan kemampuan sebuah model dalam membedakan antara kelas positif dan kelas negatif[26]. Dengan kata lain, AUC menjawab pertanyaan: "Seberapa

baik model ini dalam membedakan antara sel darah yang terindikasi Acute Lymphoblastic Leukemia (ALL) dan sel darah yang sehat?"

Nilai AUC berada dalam rentang 0 hingga 1, di mana:

- AUC = 1: Menandakan model yang sempurna. Model ini mampu membedakan kelas positif dan negatif dengan 100% benar.
- AUC = 0.5: Menandakan model yang tidak memiliki kemampuan diskriminatif sama sekali, setara dengan tebakan acak (seperti melempar koin).
- AUC < 0.5: Menandakan performa yang lebih buruk dari tebakan acak.

### 2.7.1 Kurva Receiver Operating Characteristic (ROC)

Untuk memahami AUC, penting untuk terlebih dahulu memahami kurva ROC. Kurva ROC adalah plot grafis yang mengilustrasikan kemampuan diagnostik dari sebuah sistem klasifikasi biner seiring dengan variasi ambang batas (threshold) klasifikasinya[26]. Kurva ini memetakan dua parameter penting:

- a. True Positive Rate (TPR) pada sumbu Y, yang juga dikenal sebagai Sensitivitas atau Recall. TPR mengukur proporsi kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model.

$$TPR = \frac{TP}{TP + FN} \quad (2.18)$$

Keterangan:

- TP (True Positive): Jumlah sel ALL yang diklasifikasikan dengan benar sebagai ALL.
  - FN (False Negative): Jumlah sel ALL yang salah diklasifikasikan sebagai sel sehat.
- b. False Positive Rate (FPR) pada sumbu X. FPR mengukur proporsi kasus negatif aktual yang salah diidentifikasi sebagai kasus positif

$$FPR = \frac{FP}{FP + TN} \quad (2.19)$$

Keterangan:

- FP (False Positive): Jumlah sel sehat yang salah diklasifikasikan sebagai sel ALL.
- TN (True Negative): Jumlah sel sehat yang diklasifikasikan dengan benar sebagai sel sehat.

Kurva ROC yang ideal akan melengkung tajam menuju sudut kiri atas, yang merepresentasikan TPR tinggi dan FPR rendah. Titik di sudut kiri atas (koordinat [0,1]) adalah klasifikator yang sempurna.

## 2.8 Weighted Ensemble of Network

Ensemble learning adalah sebuah pendekatan dalam machine learning yang menggabungkan prediksi dari beberapa model individual untuk menghasilkan satu prediksi akhir yang tunggal. Tujuannya adalah untuk memperoleh performa yang lebih baik, lebih stabil, dan lebih robust (tangguh) daripada yang dapat dicapai oleh model tunggal manapun[27].

Dalam penelitian ini, metode Weighted Ensemble of Networks (WEN) digunakan. Berbeda dengan ensemble sederhana yang memberi bobot sama pada setiap model, WEN mengalokasikan "bobot" atau "tingkat kepentingan" ( $w_k$ ) yang berbeda-beda untuk setiap model. Alokasi bobot ini didasarkan pada performa kuantitatif dari masing-masing model. Model yang memiliki kinerja lebih baik akan secara otomatis memberikan kontribusi yang lebih besar terhadap keputusan akhir.

Penerapan weighted ensemble ini secara sistematis melibatkan tiga langkah utama. Pertama, menghitung bobot ternormalisasi ( $w_k$ ) untuk setiap model. Bobot ini dihitung berdasarkan nilai metrik performa ( $M_k$ ) yang diperoleh model tersebut, kemudian dinormalisasi sehingga jumlah total dari semua bobot adalah 1, sesuai dengan rumus: Penerapan weighted ensemble ini secara sistematis melibatkan tiga langkah utama. Pertama, menghitung bobot ternormalisasi ( $w_k$ ) untuk setiap model.

Bobot ini dihitung berdasarkan nilai metrik performa ( $M_k$ ) yang diperoleh model tersebut, kemudian dinormalisasi sehingga jumlah total dari semua bobot adalah 1, sesuai dengan rumus:

$$w_k = \frac{M_k}{\sum_{i=1}^N M_i} \quad (2.20)$$

Keterangan:

- $w_k$  = Bobot ternormalisasi untuk model ke-k.
- $M_k$  = Nilai metrik performa (skor) dari model ke-k.
- $N$  = Jumlah total model dalam ensemble (dalam penelitian ini,  $N=3$ ).

Kedua, melakukan agregasi prediksi ( $P_{final}$ ). Probabilitas prediksi akhir untuk sebuah kelas dihitung sebagai rata-rata tertimbang (weighted average) dari probabilitas prediksi ( $P_k$ ) setiap model, menggunakan bobot yang telah dihitung sebelumnya.

$$P_{final} = \sum_{k=1}^N (w_k \cdot P_k) \quad (2.21)$$

Keterangan:

- $P_{final}$  = Probabilitas prediksi akhir (hasil ensemble).
- $P_k$  = Probabilitas prediksi yang dihasilkan oleh model ke-k.
- $w_k$  = Bobot ternormalisasi dari model ke-k.

Ketiga, mengambil keputusan klasifikasi akhir ( $y_{final}$ ). Probabilitas akhir ( $P_{final}$ ) dikonversi menjadi keputusan kelas biner (misalnya 0 atau 1) menggunakan nilai ambang batas (threshold), umumnya 0.5.

$$y_{final} = \begin{cases} 1 & \text{Jika } y_{final} \geq 0.5 \\ 0 & \text{Jika } y_{final} < 0.5 \end{cases} \quad (2.22)$$

Perbedaan fundamental dari investigasi dalam penelitian ini terletak pada metrik ( $M_k$ ) yang digunakan untuk menghitung bobot  $w_k$  pada langkah pertama. Penelitian ini akan menguji tiga skema pembobotan yang berbeda.

### 2.8.1 Pembobotan Berdasarkan F1-Score

Skema ini menggunakan F1-Score sebagai metrik performa  $M_k$ . F1-Score adalah rata-rata harmonik dari Precision dan Recall, yang sangat efektif untuk menangani kasus imbalanced dataset[12].

Dalam skema ini, nilai  $M_k$  didefinisikan sebagai:

$$M_k = \text{F1-Score}_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \quad (2.23)$$

Dimana:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} ; \quad \text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (2.24)$$

### 2.8.2 Pembobotan Berdasarkan Akurasi

Skema ini menggunakan Akurasi sebagai metrik performa  $M_k$ . Akurasi mengukur proporsi prediksi yang benar (True Positive dan True Negative) dari keseluruhan data.

Dalam skema ini, nilai  $M_k$  didefinisikan sebagai:

$$M_k = \text{Akurasi}_k = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k} \quad (2.25)$$

### 2.8.3 Pembobotan Berdasarkan Cohen's Kappa

Skema ini menggunakan koefisien Cohen's Kappa ( $\kappa_k$ ) sebagai metrik performa  $M_k$ . Kappa adalah metrik statistik yang mengukur kesepakatan (agreement) antara prediksi dan nilai aktual, dengan memperhitungkan faktor kesepakatan yang mungkin terjadi secara kebetulan[12]. Ini menjadikannya metrik yang sangat robust untuk imbalanced dataset.

Dalam skema ini, nilai  $M_k$  didefinisikan sebagai:

$$M_k = \kappa_k \quad (2.26)$$

dimana  $\kappa_k$  dihitung dengan rumus:

$$\kappa_k = \frac{p_o - p_e}{1 - p_e} \quad (2.27)$$

Keterangan:

- $p_o$  (Observed Accuracy): Proporsi kesepakatan yang diamati (sama dengan Akurasi).

$$p_o = \frac{TP_k + TN_k}{\text{Total Sampel}} \quad (2.28)$$

- $p_e$  (Expected Accuracy): Proporsi kesepakatan yang diharapkan terjadi secara kebetulan, dihitung dari total marjinal aktual dan prediksi.

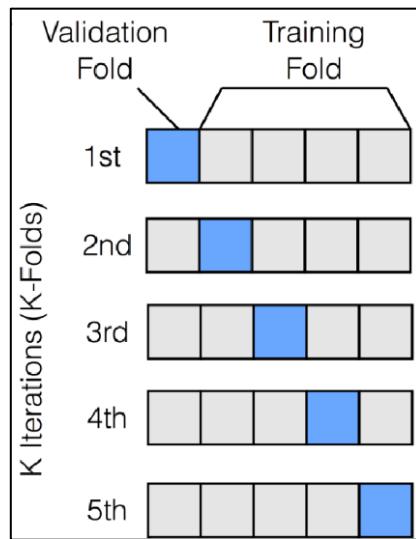
$p_e = \frac{(TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{(TP + TN + FP + FN)^2}$	(2.29)
---	--------

## 2.9 Cross Validation

Cross-validation adalah sebuah teknik fundamental dalam evaluasi model yang bertujuan untuk mengukur seberapa efektif dan general suatu model[28]. Dalam praktiknya, alur kerja yang paling umum dimulai dengan memisahkan dataset utama menjadi dua bagian: set latih (training set) yang lebih besar, dan set uji (test set) yang disimpan dan tidak digunakan hingga tahap evaluasi akhir. Metode cross-validation, khususnya K-fold cross-validation, kemudian diterapkan secara eksklusif pada set latih tersebut. Proses ini bekerja dengan membagi set latih menjadi  $k$  subset (disebut folds)[29]. Secara iteratif, model akan dilatih menggunakan  $k-1$  folds, sementara satu fold sisanya digunakan sebagai data validasi (validation set) untuk mengukur performa sementara. Proses ini diulang sebanyak  $k$  kali, memastikan setiap fold pernah berfungsi sebagai data validasi. Hasil performa dari  $k$  iterasi tersebut kemudian dirata-rata untuk mendapatkan estimasi kinerja yang stabil, yang sangat berguna untuk memilih

arsitektur model terbaik atau menyetel hyperparameter. Dengan memisahkan set uji dari awal, metode ini menjamin bahwa evaluasi akhir terhadap model yang terpilih benar-benar objektif dan tidak terpengaruh oleh data leakage, sehingga memberikan gambaran akurat tentang kemampuan generalisasi model pada data yang sepenuhnya baru. Berikut adalah Gambar 5-fold cross validation.

**Gambar 2.9 K-Fold Cross Validation**



## 2.10 Evaluasi Kinerja Sistem

Evaluasi kinerja model merupakan tahap penting untuk mengukur seberapa baik model klasifikasi dalam mengenali dan membedakan antara kelas Acute Lymphoblastic Leukemia (ALL) dan Healthy (HEM). Tujuan dari proses evaluasi ini adalah untuk menilai tingkat keakuratan model terhadap data uji dan mengetahui kemampuan model dalam memprediksi setiap kelas secara tepat.

Pada penelitian ini, proses evaluasi dilakukan menggunakan metrik evaluasi berbasis Confusion Matrix, yang mencakup Accuracy, Precision, Recall, dan F1-Score. Metrik-metrik ini banyak digunakan dalam permasalahan klasifikasi biner karena mampu memberikan gambaran menyeluruh terhadap performa model, baik dari sisi prediksi positif maupun negatif[30].

### 2.10.1 Confusion Matrix

Confusion Matrix merupakan tabel yang menggambarkan hasil prediksi model terhadap data uji dengan membandingkan nilai aktual dan nilai prediksi[12].

Tabel ini terdiri dari empat komponen utama, yaitu:

- True Positive (TP): Jumlah data di mana model dengan benar memprediksi sel sebagai ALL, dan kenyataannya memang sel tersebut adalah ALL.
- True Negative (TN): Jumlah data di mana model dengan benar memprediksi sel sebagai HEM, dan kenyataannya memang sel tersebut adalah HEM.
- False Positive (FP): Jumlah data di mana model salah memprediksi sel sebagai ALL, padahal kenyataannya sel tersebut adalah HEM.
- False Negative (FN): Jumlah data di mana model salah memprediksi sel sebagai HEM, padahal kenyataannya sel tersebut adalah ALL. Kesalahan jenis ini dianggap paling kritis dalam konteks diagnosis medis.

**Table 2. 2** Confusion Matrix

		Nilai Aktual	
		Positif	Negatif
Nilai Prediksi	Positif	TP	FP
	Negatif	FN	TN

### 2.10.2 Accuracy

Accuracy menunjukkan tingkat keseluruhan ketepatan prediksi model terhadap seluruh data uji. Nilai akurasi diperoleh dari rasio jumlah prediksi yang benar (TP dan TN) terhadap total jumlah data uji[3], yang dapat dihitung dengan rumus sebagai berikut:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.30)$$

Semakin tinggi nilai akurasi, semakin baik model dalam melakukan klasifikasi secara umum. Namun akurasi bisa menjadi metrik yang menyesatkan jika jumlah data antar kelas tidak seimbang (imbalanced dataset)[6], yang sering terjadi pada data medis.

### **2.10.3      Precision**

Presisi mengukur proporsi prediksi positif yang benar dari total prediksi yang diklasifikasikan sebagai positif[7], [31]. Metrik ini menjawab pertanyaan: "Dari semua data yang diprediksi sebagai kelas positif, berapa persen yang sebenarnya positif?"

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.31)$$

Presisi yang tinggi mengindikasikan rendahnya jumlah False Positive, yang penting dalam meminimalkan kesalahan diagnosis pada subjek yang sehat.

### **2.10.4      Recall**

Recall, yang juga dikenal sebagai Sensitivitas atau True Positive Rate (TPR), mengukur kemampuan model untuk mengidentifikasi semua sampel positif yang sebenarnya[4]. Metrik ini menjawab pertanyaan: "Dari semua data yang sebenarnya positif, berapa persen yang berhasil diprediksi dengan benar oleh model?"

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.32)$$

Dalam konteks diagnosis medis, Recall merupakan metrik yang sangat krusial karena nilai yang tinggi menunjukkan rendahnya jumlah False Negative, sehingga meminimalkan risiko gagal mendeteksi adanya penyakit.

### **2.10.5      F1-Score**

F1-Score adalah rata-rata harmonik dari Presisi dan Recall. Metrik ini digunakan untuk mencari keseimbangan antara kedua metrik tersebut, sehingga memberikan sebuah nilai tunggal yang representatif, terutama pada kasus dataset yang tidak seimbang.

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.33)$$

Nilai F1-Score yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara Presisi dan Recall.

## 2.10.6 Cohen's Kappa

Cohen's Kappa (atau Koefisien Kappa,  $\kappa$ ) adalah sebuah metrik statistik yang digunakan untuk mengukur tingkat kesepakatan (agreement) antara dua penilai[12]. Dalam evaluasi model ini, "penilai" tersebut adalah hasil prediksi model dan nilai aktual (ground truth).

Keunggulan utama Cohen's Kappa dibandingkan dengan Akurasi adalah kemampuannya untuk memperhitungkan kesepakatan yang terjadi murni secara kebetulan (by chance). Hal ini membuatnya menjadi metrik evaluasi yang jauh lebih robust, terutama untuk kasus dataset yang tidak seimbang (imbalanced dataset), di mana metrik Akurasi saja bisa memberikan skor yang tinggi namun menyesatkan[12].

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (2.34)$$

Nilai Kappa ( $\kappa$ ) dihitung menggunakan rumus berikut:

- $p_o$  (Observed Accuracy): Proporsi kesepakatan yang diamati, yang nilainya identik dengan Akurasi.

$$p_o = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.35)$$

- $p_e$  (Expected Accuracy): Proporsi kesepakatan yang diharapkan terjadi secara kebetulan. Nilai ini dihitung dari total aktual dan total prediksi untuk setiap kelas dengan rumus:

$$p_e = \frac{(TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{(TP + TN + FP + FN)^2} \quad (2.36)$$

Keterangan:

Nilai kappa berkisar antara -1 hingga 1, di mana:

- $\kappa = 1$  : Menunjukkan kesepakatan yang sempurna.

- $\kappa = 0$ : Menunjukkan kesepakatan yang terjadi tidak lebih baik dari tebakan acak.
- $\kappa < 1$ : Menunjukkan kesepakatan yang lebih buruk daripada tebakan acak.

## 2.11 Learning Rate

Learning Rate (Laju Pembelajaran) adalah salah satu hyperparameter paling krusial dalam pelatihan deep neural networks dan model machine learning yang dioptimasi menggunakan gradient descent. Secara formal, Learning Rate adalah nilai skalar positif ( $\alpha$ ) yang menentukan magnitudo atau ukuran langkah (step size) dari pembaruan bobot (weights) model pada setiap iterasi pelatihan[32].

Proses pelatihan sebuah neural network pada dasarnya adalah masalah optimasi, di mana tujuannya adalah untuk menemukan sekumpulan nilai bobot ( $W$ ) yang dapat meminimalkan Loss Function (Fungsi Kerugian,  $J(W)$ ). Untuk melakukan ini, optimizer (seperti Adam) menghitung gradient (turunan parsial,  $\nabla J(W)$ ) dari loss function terhadap setiap bobot. Gradient ini menunjukkan arah di mana loss akan meningkat paling tajam, sehingga model harus bergerak ke arah yang berlawanan untuk mengurangi loss[32].

Learning Rate bertindak sebagai pengali (multiplier) untuk gradient tersebut dalam rumus pembaruan bobot:

$$W_{\text{baru}} = W_{\text{lama}} - \alpha \cdot \nabla J(W_{\text{lama}}) \quad (2.37)$$

Keterangan:

- $W_{\text{baru}}$  : Bobot (weight) baru yang telah diperbarui.
- $W_{\text{lama}}$  : Bobot (weight) lama sebelum pembaruan.
- $\alpha$  : Learning Rate, yaitu ukuran langkah untuk pembaruan.
- $\nabla J(W_{\text{lama}})$  : Gradient dari loss function ( $J$ ) terhadap bobot lama ( $W_{\text{lama}}$ ), yang menunjukkan arah dan besaran kesalahan.

Kaitannya sangat langsung: Loss menghasilkan Gradient (sinyal kesalahan), dan Learning Rate ( $\alpha$ ) menentukan seberapa kuat sinyal tersebut akan memengaruhi perubahan Bobo[33]t. Pemilihan nilai Learning Rate memiliki dampak fundamental terhadap keberhasilan dan efisiensi pelatihan:

- Jika Learning Rate terlalu tinggi (misalnya 0.1 atau 0.01): Langkah pembaruan bobot akan terlalu besar. Alih-alih menuruni "lembah" loss secara bertahap, model dapat "melompati" titik minimum optimal. Hal ini menyebabkan proses pelatihan menjadi tidak stabil (unstable), di mana nilai loss akan berfluktuasi secara liar, dan model gagal converge (mencapai titik optimal), atau bahkan diverge (nilai loss meledak menjadi tak terhingga)[34].
- Jika Learning Rate terlalu rendah (misalnya 0.000001): Langkah pembaruan bobot akan terlalu kecil. Meskipun pelatihan akan cenderung lebih stabil, prosesnya akan menjadi sangat lambat dan membutuhkan jumlah epoch yang jauh lebih banyak untuk mencapai convergence. Selain itu, dengan langkah yang terlalu kecil, model berisiko lebih tinggi untuk terjebak dalam local minimum (titik optimal yang suboptimal) karena tidak memiliki "momentum" yang cukup untuk keluar dan menemukan global minimum (titik optimal yang sebenarnya)[34].

Oleh karena itu, menemukan Learning Rate yang optimal adalah sebuah kompromi: harus cukup besar untuk menjamin convergence yang efisien, namun cukup kecil untuk memungkinkan model "menetap" secara stabil di dekat titik minimum yang baik.

## 2.12 Class Weight

Class Weight (Pembobotan Kelas) adalah sebuah teknik yang digunakan untuk menangani masalah imbalanced dataset (dataset tidak seimbang). Dataset tidak seimbang adalah kondisi di mana jumlah sampel pada satu kelas (kelas mayoritas) jauh lebih banyak daripada jumlah sampel pada kelas lainnya (kelas minoritas)[35].

Masalah ini sangat umum terjadi pada dataset medis, di mana jumlah data "normal" atau "sehat" (HEM) mungkin lebih sedikit atau lebih banyak daripada data "sakit" (ALL). Jika model dilatih pada dataset yang tidak seimbang tanpa penanganan khusus, model akan cenderung bias terhadap kelas mayoritas. Model akan mencapai akurasi yang tinggi hanya dengan selalu menebak kelas mayoritas, namun akan memiliki performa yang sangat buruk dalam mengidentifikasi kelas minoritas yang seringkali lebih penting.

Metode Class Weight mengatasi masalah ini dengan cara memodifikasi Loss Function (Fungsi Kerugian) dengan memberikan "denda" atau penalti yang berbeda untuk kesalahan klasifikasi, tergantung pada kelasnya. Secara spesifik kelas minoritas akan diberikan bobot yang lebih tinggi sedangkan untuk kelas minoritas akan diberikan bobot yang lebih rendah[35].

Selama proses pelatihan, ketika model menghitung loss (kesalahan), loss dari setiap sampel akan dikalikan dengan bobot kelasnya. Akibatnya, kesalahan klasifikasi (misclassification) pada satu sampel kelas minoritas akan menghasilkan nilai loss yang jauh lebih besar (karena dikalikan bobot tinggi) daripada kesalahan pada satu sampel kelas mayoritas.

Hal ini "memaksa" optimizer (seperti Adam) untuk memberikan perhatian lebih dan bekerja lebih keras untuk mengklasifikasikan sampel-sampel dari kelas minoritas dengan benar, demi menekan total loss seefektif mungkin.

Bobot ini biasanya dihitung secara otomatis berdasarkan invers (kebalikan) dari frekuensi kelas. Rumus umum untuk menghitung bobot ( $W_j$ ) untuk kelas  $j$  adalah:

$$W_j = \frac{N}{k \times N_j} \quad (2.38)$$

Keterangan:

- $W_j$  : Bobot untuk kelas  $j$ .

- $N$  : Jumlah total seluruh sampel dalam dataset.
- $k$  : Jumlah total kelas (dalam penelitian ini,  $k = 2$ )
- $N_j$  : Jumlah sampel dalam kelas  $j$

## **BAB III**

### **METODE USULAN**

#### **3.1 Dataset**

Dataset yang digunakan dalam penelitian ini adalah C-NMC 2019 Dataset, yang merupakan bagian dari ALL Challenge – ISBI 2019 dan dipublikasikan melalui The Cancer Imaging Archive (TCIA) Public Access serta tersedia juga pada platform Kaggle. Dataset ini dirancang secara khusus untuk tujuan klasifikasi sel darah putih pada pasien dengan Acute Lymphoblastic Leukemia (ALL) terhadap sel darah normal (Healthy). Seluruh data berupa citra mikroskopis apusan darah yang telah melalui proses segmentasi sel secara manual, sehingga tiap citra berfokus pada satu sel tunggal.

Dataset ini pertama kali digunakan dalam IEEE ISBI 2019 Challenge: Classification of Normal vs Malignant Cells in B-ALL White Blood Cancer Microscopic Images, yang dapat diakses melalui laman resmi ISBI 2019 Challenge dan CodaLab Competition melalui link <https://www.cancerimagingarchive.net/collection/c-nmc-2019/>. Dataset ini juga dipublikasikan ulang di Kaggle melalui tautan: <https://www.kaggle.com/datasets/andrewmvd/leukemia-classification>.

Secara keseluruhan, dataset C-NMC 2019 terdiri atas 15.114 citra sel, yang berasal dari 118 individu subjek dengan distribusi 91 pasien ALL dan 72 pasien normal. Akan tetapi, dalam penelitian ini hanya digunakan 12.528 citra, sementara sebagian lainnya dialokasikan khusus untuk keperluan evaluasi kompetisi di CodaLab. Komposisi dataset berdasarkan kelas dan pembagiannya dapat dilihat pada table berikut:

**Table 3. 1 Komposisi Struktur Asli Dataset**

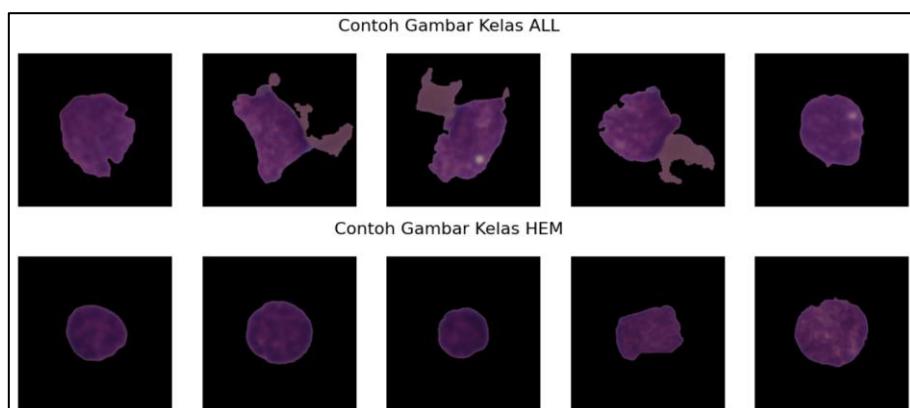
<b>Komposisi</b>	<b>Sel ALL</b>	<b>Sel Normal</b>	<b>Pasien ALL</b>	<b>Pasien Normal</b>
Train set composition	7272	3389	69	49

Preliminary test set composition	1219	648	13	15
Final test set composition		2586	9	8

Setiap citra disimpan dalam format BMP dengan ukuran resolusi  $450 \times 450$  piksel. Label ground truth ditentukan oleh pakar onkologi sehingga memiliki validitas yang tinggi. Penamaan file mengikuti konvensi tertentu, misalnya UID\_P\_N\_C\_all untuk sel ALL (kanker) dan UID\_HS\_N\_C\_hem untuk sel normal (healthy).

Karakteristik dari dataset ini menunjukkan adanya ketidakseimbangan kelas (imbalanced dataset), karena jumlah citra sel ALL lebih banyak dibanding citra sel normal. Kondisi ini perlu ditangani dengan teknik penyeimbang data.

**Gambar 3. 1** File Gambar ALL

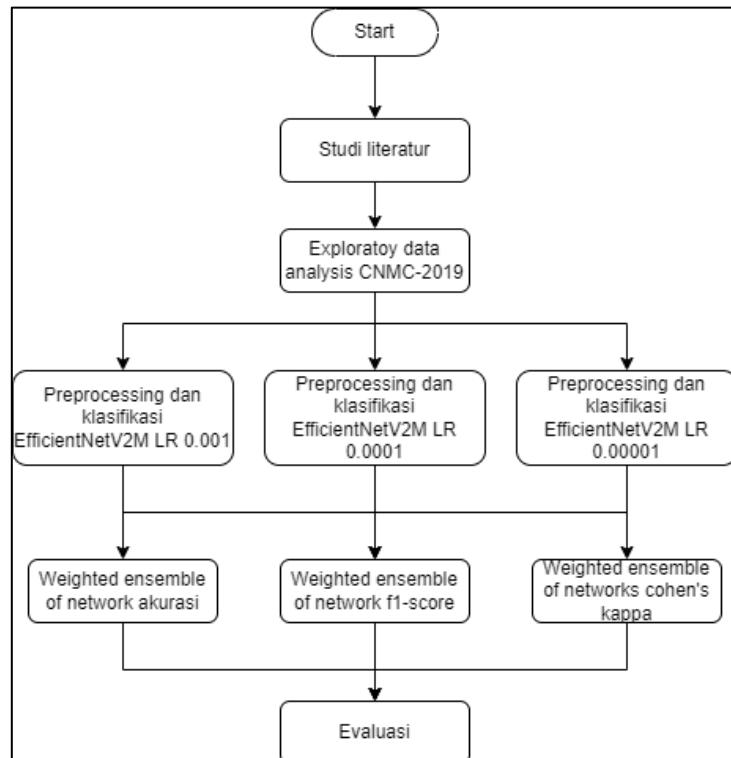


Secara morfologis dapat dilihat pada gambar diatas, sel ALL dan sel normal memiliki kemiripan yang tinggi sehingga sulit dibedakan hanya dengan pengamatan visual. Analisis laboratorium menunjukkan bahwa sel ALL cenderung memiliki ukuran bervariasi, bentuk inti sel tidak beraturan, serta terdapat nucleolus yang menonjol. Sitoplasma pada sel ALL sering kali memiliki volume besar dengan vakuola yang jelas terlihat. Perbedaan lebih nyata dapat ditemukan ketika dilakukan pewarnaan Romanowsky atau Giemsa, di mana sel ALL menampilkan warna limfoblas lebih gelap dibandingkan sel limfosit normal.

### 3.2 Rancangan Arsitektur Penelitian

Rancangan sistem pada penelitian ini dirancang secara sistematis untuk menjawab rumusan masalah yang telah ditetapkan. Bagian ini menjelaskan arsitektur penelitian secara keseluruhan (high-level) dari tahap awal hingga tahap evaluasi akhir. Alur kerja penelitian secara umum diilustrasikan pada Gambar 3.2.

**Gambar 3. 2 Rancangan Arsitektur Penelitian**



Penelitian dimulai dengan dua tahap fundamental. Pertama adalah Studi Literatur, yang bertujuan untuk mengumpulkan pengetahuan mengenai penelitian terdahulu yang sudah dilakukan dibidang Acute Lymphoblastic Leukemia serta menganalisis penggunaan metode dan hasil yang didapatkan. Tahap kedua adalah Exploratory Data Analysis (EDA) pada dataset C-NMC 2019 untuk memahami karakteristik, distribusi kelas, dan kualitas data yang akan digunakan.

Setelah tahap persiapan data selesai, penelitian masuk ke tahap inti, yaitu eksperimen pelatihan model. Seperti ditunjukkan pada Gambar 3.2, penelitian akan

menjalankan tiga alur eksperimen klasifikasi secara paralel dan independen. Ketiga alur ini menggunakan arsitektur dasar yang sama, yaitu EfficientNetV2-M, dan metodologi pelatihan yang identik (termasuk preprocessing, batch size, epoch, stage pelatihan, dan 5-Fold CV).

Perbedaan fundamental di antara ketiga alur tersebut adalah hyperparameter Learning Rate (LR) yang digunakan: 0.001, 0.0001, dan 0.00001. Setiap alur ("Preprocessing dan klasifikasi") ini akan menghasilkan serangkaian model yang telah dilatih. Proses teknis yang mendetail untuk setiap alur pelatihan paralel ini akan dijabarkan lebih lanjut pada Sub-bab 3.3.

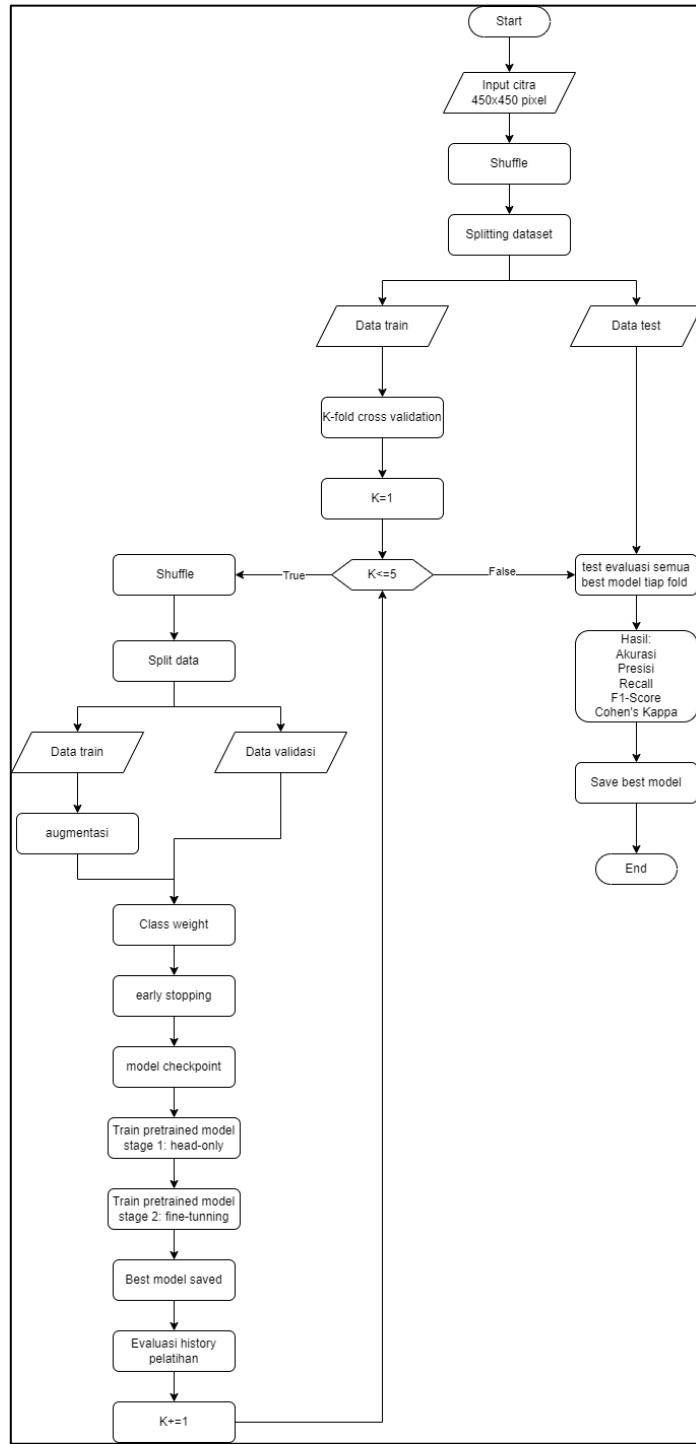
Setelah ketiga alur pelatihan menghasilkan model-model terbaiknya, tahap selanjutnya adalah pengujian Weighted Ensemble of Networks (WEN). Pada tahap ini, model-model yang telah dilatih akan digabungkan untuk menguji apakah performa gabungan dapat melampaui performa model tunggal terbaik. Seperti yang ditunjukkan pada alur, pengujian WEN akan diinvestigasi menggunakan tiga skema pembobotan yang berbeda, yaitu berdasarkan Akurasi, F1-Score, dan Cohen's Kappa. Rancangan detail untuk implementasi tahap WEN ini akan dijelaskan lebih lanjut pada Sub-bab 3.4.

Tahap terakhir dari arsitektur penelitian adalah Evaluasi. Pada tahap ini, seluruh hasil performa—baik dari model-model tunggal (hasil perbandingan 3 skenario LR) maupun dari berbagai skenario WEN—akan dianalisis dan dibandingkan secara komprehensif menggunakan data uji. Tujuannya adalah untuk mengidentifikasi hyperparameter dan strategi ensemble yang paling optimal untuk menjawab rumusan masalah.

### **3.3 Rancangan Sistem Pelatihan Model**

Bagian ini merinci arsitektur teknis dan alur proses untuk melatih model klasifikasi. Proses yang dijelaskan di sini berlaku untuk setiap alur eksperimen Learning Rate (0.001, 0.0001, dan 0.00001) yang telah dipaparkan pada sub-bab sebelumnya. Rancangan sistem pelatihan model dapat dilihat pada Gambar 3.3.

**Gambar 3. 3 Rancangan Sistem Pelatihan Model**



Proses pelatihan diawali dengan Input Data Citra. Dataset yang digunakan adalah C-NMC 2019, berisi citra mikroskopis sel darah putih berukuran 450x450 piksel. Data ini kemudian melalui tahap Preprocessing Data yang meliputi:

- Resize Citra: Mempertahankan ukuran asli 450x450 piksel untuk menjaga detail citra.
- Normalisasi: Dilakukan secara otomatis melalui fungsi preprocessing bawaan EfficientNetV2, yang mengubah nilai piksel dari rentang 0-255 menjadi 0-1.
- Batching dan Prefetching: Menyiapkan data dalam batch berukuran 64 untuk mempercepat proses pelatihan.

Selanjutnya, dilakukan Shuffle dan Splitting Dataset. Data diacak lalu dibagi menjadi dua bagian: 80% data latih (training data) dan 20% data uji (testing data). Data uji ini akan disimpan dan tidak akan pernah digunakan selama proses pelatihan untuk memastikan evaluasi akhir yang objektif.

Inti dari metodologi pelatihan ini adalah penerapan 5-Fold Cross Validation ( $K=5$ ) pada 80% data latih tersebut<sup>13</sup>. Seperti yang ditunjukkan pada Gambar 3.3, data latih akan masuk ke dalam loop validasi silang sebanyak 5 kali ( $K=1$  hingga  $K=5$ ). Pada setiap iterasi fold:

1. Shuffle dan Split: Data latih (80% total data) diacak dan dibagi lagi menjadi dua set: data latih baru (4 fold) dan data validasi (1 fold).
2. Augmentasi: Penerapan transformasi acak seperti flip horizontal, flip vertikal, perubahan kecerahan, dan kontras diterapkan hanya pada data latih baru untuk meningkatkan variasi data dan mencegah overfitting.
3. Penanganan Imbalance: Teknik Class Weight diterapkan. Teknik ini memberikan bobot yang lebih besar pada kelas minoritas (sel normal/HEM) selama perhitungan loss function, sehingga model tidak bias terhadap kelas mayoritas (sel ALL).

4. Callback Pelatihan: Dua callback penting diaktifkan, yaitu Early Stopping untuk menghentikan pelatihan secara otomatis jika performa pada data validasi tidak lagi meningkat, dan Model Checkpoint untuk memantau dan menyimpan bobot model terbaik berdasarkan nilai validation accuracy.
5. Pelatihan 2 Tahap: Proses pelatihan dilakukan dalam dua tahap berurutan:
  - a. Tahap 1 (Head-only Training): Seluruh lapisan dasar (base model) EfficientNetV2-M dibekukan (frozen). Lapisan head klasifikasi (yang baru ditambahkan) dilatih selama 50 epoch. Secara spesifik, head lama diganti dengan lapisan Dense yang memiliki 1 neuron dan fungsi aktivasi Sigmoid, untuk menghasilkan probabilitas dua kelas (ALL dan HEM).
  - b. Tahap 2 (Fine-tuning): Model terbaik dari Tahap 1 dimuat kembali. Sebagian lapisan atas dari base model dibuka (unfrozen), dan seluruh model dilatih kembali (di-fine-tuning) dengan learning rate yang sama selama 50 epoch.
6. Penyimpanan Model Fold: Model terbaik yang dihasilkan dari iterasi fold ini (Best model saved) disimpan.

Setelah loop K=5 selesai, akan ada 5 model terbaik (satu dari setiap fold). Langkah selanjutnya adalah mengevaluasi kelima model ini menggunakan data uji (Data test) yang telah disisihkan. Model yang menunjukkan performa tertinggi pada data uji (berdasarkan metrik Akurasi, Presisi, Recall, F1-Score, dan Cohen's Kappa) akan dipilih dan disimpan sebagai satu model "juara" (Save best model). Model "juara" inilah yang akan mewakili skenario *Learning Rate* tersebut dalam tahap ensemble di sub-bab berikutnya.

### **3.4 Rancangan Weighted Ensemble of Networks (WEN)**

Tahap *Weighted Ensemble of Networks* (WEN) bertujuan untuk menggabungkan hasil prediksi dari beberapa model tunggal yang telah dilatih.

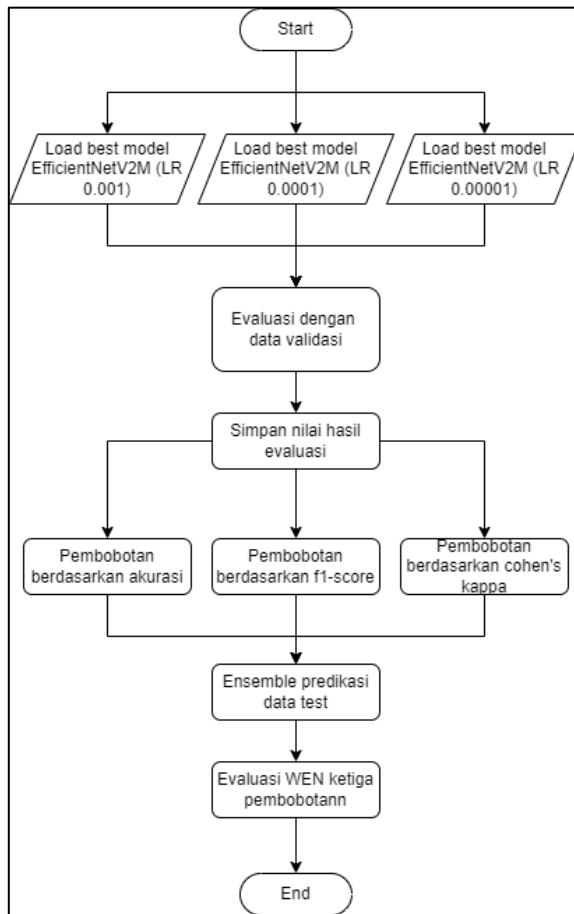
Tujuannya adalah untuk menghasilkan keputusan akhir yang lebih akurat dan stabil (robust) daripada yang dapat dicapai oleh model tunggal terbaik sekalipun.

Dalam penelitian ini, dua skenario WEN utama akan diinvestigasi:

1. WEN Inter-LR (Ensemble per Skenario LR): Skenario ini dilakukan di dalam setiap grup Learning Rate. Contohnya, 5 model terbaik yang dihasilkan dari 5-Fold Cross Validation pada skenario LR 0.001 akan digabungkan. Proses ini diulang untuk skenario LR 0.0001 dan LR 0.00001. Tujuannya adalah untuk melihat apakah gabungan dari 5 fold lebih stabil daripada satu fold "juara".
2. WEN Intra-LR (Ensemble Gabungan Juara): Skenario ini adalah inti dari pengujian ensemble penelitian ini. Di sini, tiga model "juara" (satu model terbaik yang mewakili setiap skenario LR: 0.001, 0.0001, dan 0.00001) akan digabungkan. Hipotesisnya adalah bahwa model-model yang dilatih dengan Learning Rate berbeda akan mempelajari fitur dengan cara yang berbeda, dan menggabungkan "keahlian" yang beragam ini akan menghasilkan model yang paling unggul.

Alur proses untuk skenario WEN Intra-LR (Gabungan Juara) diilustrasikan secara rinci pada Gambar 3.4.

**Gambar 3. 4 Rancangan Weighted Ensemble of Networks (Intra-LR)**



Tahapan prosesnya dijelaskan sebagai berikut:

1. Load Model: Sistem memanggil tiga model "juara" yang telah disimpan dari tahap 3.3, yaitu: Best model EfficientNetV2M (LR 0.001), Best model EfficientNetV2M (LR 0.0001), dan Best model EfficientNetV2M (LR 0.00001).
2. Evaluasi Awal Tiap Model: Masing-masing dari ketiga model tersebut dievaluasi menggunakan data validasi.
3. Simpan Nilai Hasil Evaluasi: Hasil dari evaluasi ini (nilai metrik Akurasi, F1-Score, dan Cohen's Kappa) disimpan variable. Nilai-nilai inilah yang akan menjadi dasar untuk perhitungan bobot.

4. Perhitungan Bobot (Tiga Skema): Proses ensemble dilakukan sebanyak tiga kali secara paralel untuk menguji tiga skema pembobotan yang berbeda:
  - a. Pembobotan berdasarkan akurasi ( $WEN^{acc}$ ): Bobot untuk setiap model (LR 0.001, 0.0001, 0.00001) diambil dari nilai Akurasi yang diperoleh pada langkah 2
  - b. Pembobotan berdasarkan f1-score ( $WEN^{f1}$ ): Bobot untuk setiap model diambil dari nilai F1-Score yang diperoleh pada langkah 2.
  - c. Pembobotan berdasarkan cohen's kappa ( $WEN^{kappa}$ ): Bobot untuk setiap model diambil dari nilai Cohen's Kappa yang diperoleh pada langkah 2.
5. Ensemble Prediksi Data Test: Untuk setiap skema pembobotan, prediksi probabilitas dari ketiga model digabungkan menggunakan rata-rata tertimbang (weighted average) sesuai bobotnya masing-masing untuk menghasilkan prediksi akhir.
6. Evaluasi WEN Ketiga Pembobotan: Hasil akhir dari ketiga model WEN ( $WEN^{acc}$ ,  $WEN^{f1}$ , dan  $WEN^{kappa}$ ) dievaluasi kembali performanya menggunakan data test. Tujuannya adalah untuk membandingkan performa dari ketiga skema pembobotan tersebut dan menentukan skema mana yang menghasilkan model ensemble paling optimal, sekaligus untuk mengukur peningkatannya dibandingkan dengan performa model tunggal "juara" terbaik.

### 3.5 Skenario Uji Coba

Skenario uji coba dilakukan untuk memperoleh model klasifikasi citra leukemia dengan performa terbaik melalui investigasi hyperparameter yang optimal. Proses pelatihan model EfficientNetV2-M dilakukan dalam dua tahap, yaitu Head Training dan Fine-Tuning, masing-masing selama 50 epoch.

Optimizer yang digunakan adalah Adam, dengan nilai batch size sebesar 64 dan untuk menangani ketidakseimbangan kelas, digunakan teknik Class Weight. Seluruh pelatihan model tunggal dilakukan menggunakan metode 5-Fold Cross Validation.

Fokus utama dari skenario uji coba ini adalah untuk membandingkan tiga skenario pelatihan yang berbeda berdasarkan Learning Rate (LR). Detail dari ketiga skenario pelatihan model tunggal dapat dilihat pada Tabel 3.2.

**Table 3. 2** Skenario Uji Coba Pelatihan

Model	Learning Rate	Epoch	Metode Validasi	Balancing Data
EfficientNetV2-M	0.001	50 (Head) + 50 (Fine-tuning)	5-Fold Cross Validation	Class Weight
	0.0001			
	0.00001			

Setelah ketiga skenario pelatihan model tunggal di atas selesai dan menghasilkan model-model terbaik, skenario uji coba berikutnya adalah menerapkan *Weighted Ensemble of Networks* (WEN). Skenario *ensemble* dibagi menjadi dua jenis investigasi:

1. WEN Inter-LR: Menggabungkan 5 model yang dihasilkan dari 5-Fold CV di dalam satu skenario LR yang sama (contoh: 5 model dari LR 0.001 digabung).
2. WEN Intra-LR: Menggabungkan 3 model "juara" (satu model terbaik yang mewakili setiap skenario LR).

Untuk setiap skenario ensemble ini, akan diuji tiga skema pembobotan (Akurasi, F1-Score, dan Cohen's Kappa). Rincian skenario ensemble dapat dilihat pada Tabel 3.3.

**Table 3. 3** Skenario Ensemble Pembobotan

Skenario	Model yang digabungkan	Dasar Pembobotan
Inter-LR 0.001	5 Model Fold (dari LR 0.001)	Akurasi
		F1-Score
		Cohen's Kappa
Inter-LR 0.0001	5 Model Fold (dari LR 0.0001)	Akurasi
		F1-Score

		Cohen's Kappa
Inter-LR 0.00001	5 Model Fold (dari LR 0.00001)	Akurasi
		F1-Score
		Cohen's Kappa
Intra-LR	3 Model "Juara" (terbaik dari tiap LR)	Akurasi
		F1-Score
		Cohen's Kappa

Pada skenario Intra-LR, setiap model (yang mewakili LR-nya) memberikan output probabilitas yang kemudian dikombinasikan menggunakan bobot sesuai metrik masing-masing model pada data uji. Hasil akhir dari ensemble merupakan nilai rata-rata tertimbang dari ketiga model tersebut.

Pada skenario Inter-LR, proses yang sama dilakukan, namun model yang digabungkan adalah 5 model dari cross-validation dalam satu grup LR yang sama. Evaluasi ini akan menentukan strategi hyperparameter dan strategi ensemble yang paling optimal.

## DAFTAR PUSTAKA

- [1] A. Muntasa, R. T. Wahyuningrum, Z. Tuzzahra, A. Motwakel, M. Yusuf, and W. F. Mahmudi, "A Pyramid Model of Convolutional Neural Network to Classify Acute Lymphoblastic Leukemia Images," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 6, pp. 576–588, Dec. 2022, doi: 10.22266/ijies2022.1231.51.
- [2] 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, Oct. 2022, doi: 10.22266/ijies2022.1031.54.
- [3] L. Rodrigues Moreira, A. Backes, B. Travençolo, and G. Oliveira, "Optimizing a Deep Residual Neural Network with Genetic Algorithm for Acute Lymphoblastic Leukemia Classification," *J Digit Imaging*, vol. 35, pp. 623–637, Feb. 2022, doi: 10.1007/s10278-022-00600-3.
- [4] K. Abdulameer, F. Najjar, A. A. Waad, I. Al-Kharsan, Z. N. Khudhair, and A. Salim, "Leukemia Classification using a Convolutional Neural Network of AML Images," *Malaysian Journal of Fundamental and Applied Sciences*, vol. 19, pp. 306–312, May 2023.
- [5] K. Wibowo, N. Rianto, and J. Unjung, "Improved convolutional neural network model for leukemia classification using EfficientNetV2M and bayesian optimization," *Journal of Soft Computing Exploration*, vol. 5, pp. 192–199, Jun. 2024, doi: 10.52465/josce.v5i2.378.
- [6] A. Batool and Y. Byun, "Lightweight EfficientNetB3 Model Based on Depthwise Separable Convolutions for Enhancing Classification of Leukemia White Blood Cell Images," *IEEE Access*, vol. PP, p. 1, Jan. 2023, doi: 10.1109/ACCESS.2023.3266511.
- [7] M. Zakir Ullah *et al.*, "An Attention-Based Convolutional Neural Network for Acute Lymphoblastic Leukemia Classification," *Applied Sciences*, vol. 11, p. 10662, Nov. 2021, doi: 10.3390/app112210662.
- [8] A. Muntasa *et al.*, "A Novel Hybrid Convolutional and Network Encapsulation Approach in EfficientNetV2-S Architecture for Acute Lymphoblastic Leukemia Classification," *International Journal of Intelligent Engineering and Systems*, 2024, [Online]. Available: <https://api.semanticscholar.org/CorpusID:273622375>
- [9] I. Ahmed, E. Senan, H. Shatnawi, Z. Alkhraisha, and M. Al-Azzam, "Hybrid Techniques for the Diagnosis of Acute Lymphoblastic Leukemia Based on Fusion of CNN Features," *Diagnostics*, vol. 13, p. 1026, Mar. 2023, doi: 10.3390/diagnostics13061026.
- [10] M. Ghadezadeh, M. Aria, A. Hosseini, F. Asadi, D. Bashash, and H. Abolghasemi, "A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images," *International Journal of Intelligent Systems*, vol. 37, Nov. 2021, doi: 10.1002/int.22753.

- [11] D. Baby, S. Juliet, and M. Raj, "An efficient lymphocytic leukemia detection based on EfficientNets and ensemble voting classifier," *Int J Imaging Syst Technol*, vol. 33, p. n/a-n/a, Jul. 2022, doi: 10.1002/ima.22787.
- [12] C. Mondal *et al.*, "Ensemble of Convolutional Neural Networks to diagnose Acute Lymphoblastic Leukemia from microscopic images," *Inform Med Unlocked*, vol. 27, p. 100794, Nov. 2021, doi: 10.1016/j.imu.2021.100794.
- [13] H. Yu, L. T. Yang, Q. Zhang, D. Armstrong, and M. J. Deen, "Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives," *Neurocomputing*, vol. 444, pp. 92–110, 2021, doi: <https://doi.org/10.1016/j.neucom.2020.04.157>.
- [14] I. D. Mienye, T. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," *Information*, p., 2025, doi: 10.3390/info16030195.
- [15] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of Image Classification Algorithms Based on Convolutional Neural Networks," *Remote. Sens.*, vol. 13, p. 4712, 2021, doi: 10.3390/rs13224712.
- [16] Md. S. I. Khan *et al.*, "Accurate brain tumor detection using deep convolutional neural network," *Comput Struct Biotechnol J*, vol. 20, pp. 4733–4745, 2022, doi: 10.1016/j.csbj.2022.08.039.
- [17] R. Dasgupta, Y. S. Chowdhury, and S. Nanda, "Performance Comparison of Benchmark Activation Function ReLU, Swish and Mish for Facial Mask Detection Using Convolutional Neural Network," *Algorithms for Intelligent Systems*, p., 2021, doi: 10.1007/978-981-16-2248-9\_34.
- [18] S. Niyas, S. Pawan, A. Kumar, and J. Rajan, "Medical image segmentation with 3D convolutional neural networks: A survey," *Neurocomputing*, vol. 493, pp. 397–413, 2021, doi: 10.1016/j.neucom.2022.04.065.
- [19] S. Puligadda, G. Karthik, U. Polina, S. Subbarao, V. P. K. Sistla, and V. K. K. Kolli, "A Novel Transfer Learning-based Efficient-Net for Visual Image Tracking," *2025 International Conference on Pervasive Computational Technologies (ICPCT)*, pp. 531–537, 2025, doi: 10.1109/icpct64145.2025.10941256.
- [20] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, "Efficient densely connected convolutional neural networks," *Pattern Recognit.*, vol. 109, p. 107610, 2021, doi: 10.1016/j.patcog.2020.107610.
- [21] R. Fu, M. Cao, D. Novák, X. Qian, and N. Alkayem, "Extended efficient convolutional neural network for concrete crack detection with illustrated merits," *Autom Constr*, p., 2023, doi: 10.1016/j.autcon.2023.105098.

- [22] C. Lin, P. Yang, Q. Wang, Z. Qiu, W. Lv, and Z. Wang, "Efficient and accurate compound scaling for convolutional neural networks," *Neural Netw*, vol. 167, pp. 787–797, 2023, doi: 10.1016/j.neunet.2023.08.053.
- [23] V.-T. Hoang, N. D. Quang, T. M. Phuong, K.-H. Jo, and V.-D. Hoang, "A compact version of EfficientNet for skin disease diagnosis application," *Neurocomputing*, vol. 620, p. 129166, 2024, doi: 10.1016/j.neucom.2024.129166.
- [24] M. Di, Z. Peng, and S. Chen, "Performance comparison and application of deep-learning-based image recognition models," vol. 13550, p. 135504, 2025, doi: 10.1111/12.3059903.
- [25] F. Mehmood, S. Ahmad, and T. Whangbo, "An Efficient Optimization Technique for Training Deep Neural Networks," *Mathematics*, p., 2023, doi: 10.3390/math11061360.
- [26] A. Carrington *et al.*, "Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation," *IEEE Trans Pattern Anal Mach Intell*, vol. 45, pp. 329–341, 2021, doi: 10.1109/tpami.2022.3145392.
- [27] X. Zhou, J. He, and C. Yang, "An ensemble learning method based on deep neural network and group decision making," *Knowl. Based Syst.*, vol. 239, p. 107801, 2021, doi: 10.1016/j.knosys.2021.107801.
- [28] L. Yates, Z. Aandahl, S. Richards, and B. Brook, "Cross validation for model selection: a review with examples from ecology," *Ecol Monogr*, p., 2022, doi: 10.1002/ecm.1557.
- [29] V. Lumumba, D. Kiprotich, M. Mpaine, N. Makena, and M. Kavita, "Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models," *American Journal of Theoretical and Applied Statistics*, p., 2024, doi: 10.11648/j.ajtas.20241305.13.
- [30] J. Obi, "A comparative study of several classification metrics and their performances on data," *World Journal of Advanced Engineering Technology and Sciences*, p., 2023, doi: 10.30574/wjaets.2023.8.1.0054.
- [31] M. Z. R. Azhary and A. R. Ismail, "Comparative Performance of Different Convolutional Neural Network Activation Functions on Image Classification," *International Journal on Perceptive and Cognitive Computing*, p., 2024, doi: 10.31436/ijpcc.v10i2.490.
- [32] Y. Wu and L. Liu, "Selecting and Composing Learning Rate Policies for Deep Neural Networks," *ACM Trans Intell Syst Technol*, vol. 14, pp. 1–25, 2022, doi: 10.1145/3570508.
- [33] Z. Wang, H.-B. Gao, X. Wang, S. Zhao, H. Li, and X.-Q. Zhang, "Adaptive learning rate optimization algorithms with dynamic bound based on Barzilai-Borwein method," *Inf. Sci.*, vol. 634, pp. 42–54, 2023, doi: 10.1016/j.ins.2023.03.050.
- [34] S.-H. Cha and D.-S. Kim, "Efficient Training of Deep Spiking Neural Networks Using a Modified Learning Rate Scheduler," *Mathematics*, p., 2025, doi: 10.3390/math13081361.

- [35] B. Bakırarar and A. Elhan, "Class Weighting Technique to Deal with Imbalanced Class Problem in Machine Learning: Methodological Research," *Turkiye Klinikleri Journal of Biostatistics*, p., 2023, doi: 10.5336/biostatic.2022-93961.