

**PENERAPAN *ROBUST INCREMENTAL LEARNING*  
DENGAN RESNET-50 DAN BIAS CORRECTION LAYER  
UNTUK KLASIFIKASI CITRA**



**ARTIKEL ILMIAH**

**Disusun Sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Komputer  
pada Program Studi Informatika**

**Disusun oleh:  
Axelliano Rafael Situmeang  
24060121130089**

**DEPARTEMEN INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO**

**2025**

## HALAMAN PENGESAHAN

Judul : Penerapan *Robust Incremental Learning* dengan *Resnet-50* dan *Bias Correction Layer* untuk Klasifikasi Citra

Nama : Axelliano Rafael Situmeang

NIM : 24060121130089

Telah diujikan pada sidang tugas akhir dan dinyatakan lulus pada tanggal 13 Maret 2025

Semarang, 21 Maret 2025

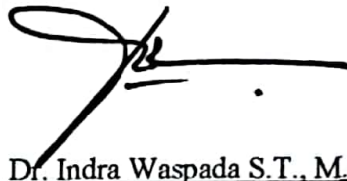
Pembimbing 1,



Rismiyati, B.Eng., M.Cs

NIP. 19851125 201803 2 001

Pembimbing 2,



Dr. Indra Waspada S.T., M.T.I

NIP. 19790212 200812 1 002



# Penerapan Robust Incremental Learning dengan ResNet-50 dan Bias Correction Layer Untuk Klasifikasi Citra

Axelliano Rafael Situmeang<sup>\*1)</sup>, Rismiyati<sup>2)</sup>, dan Indra Waspada<sup>3)</sup>

<sup>1,2,3)</sup>Fakultas Sains dan Matematika, Universitas Diponegoro, Semarang, Indonesia

\* Corresponding author: shikakunxd@students.undip.ac.id

## Abstrak

Model deep learning konvensional dengan pendekatan pelatihan closed world classification memiliki keterbatasan untuk mengakomodasi pengetahuan kelas baru. Penggunaannya mengharuskan pelatihan ulang untuk menambah pengetahuan kelas baru, yang bertentangan dengan pembelajaran inkremental. Tantangan utama pembelajaran inkremental adalah catastrophic forgetting, yakni hilangnya kemampuan model dalam mempertahankan pengetahuan lama saat mempelajari informasi baru serta bias terhadap kelas baru. Penelitian ini mengevaluasi metode Bias Correction (BiC) dalam kerangka Class Incremental Learning (CIL) yang memfasilitasi pembelajaran inkremental pada arsitektur ResNet-50 menggunakan 20 kelas dari ImageNet100 yang bernoise rendah variasi hyperparameter meliputi pixel size, learning rate, dan step size. Hasil penelitian menunjukkan penerapan BiC menghasilkan akurasi model sebesar 88,7% dengan peningkatan sebesar 0,7% pada skenario 2 step size, serta peningkatan akurasi terbesar yakni 8,79% pada skenario 4 step size. Penerapan BiC juga efektif menurunkan forgetting rate dari 12,13% menjadi 8,47% pada skenario 5 step size serta 8,94% menjadi 4,50% pada skenario 4 step size. Temuan ini mengkonfirmasi efektivitas BiC dalam mengatasi catastrophic forgetting dan mengatasi bias pada skenario data yang lebih kecil.

**Kata Kunci :** BiC Method, Class Incremental Learning, Image Classification, ResNet, ImageNet

## Abstract

*Conventional deep learning models with closed world classification training approaches have limitations in accommodating new class knowledge. Their use necessitates complete retraining to add new class knowledge, which contradicts the incremental nature of natural learning processes. The primary challenge in incremental learning is catastrophic forgetting—the model's inability to retain previous knowledge while learning new information and bias toward new classes. This research evaluates the Bias Correction (BiC) method within the Class Incremental Learning (CIL) framework that facilitates incremental learning on ResNet-50 architecture using 20 low-noise classes from ImageNet100 with various hyperparameter scenarios including pixel size, learning rate, and step size. Results demonstrate that BiC implementation achieved 88.7% model accuracy with a 0.7% improvement in the 2-step size scenario and a significant 8.79% accuracy increase in the 4-step size scenario. BiC effectively reduced the forgetting rate from 12.13% to 8.47% in the 5-step size scenario and from 8.94% to 4.50% in the 4-step size scenario. These findings confirm BiC's effectiveness in addressing catastrophic forgetting and mitigating bias in smaller data scenarios.*

**Keywords :** BiC Method, Class Incremental Learning, Image Classification, ResNet, ImageNet

## 1 Pendahuluan

Perkembangan *deep learning* telah mencapai tahap dimana teknologi ini mampu menyamai performa manusia di berbagai bidang [1]. Namun, performa tersebut kerap dihasilkan dari pendekatan *Closed World Classification* yang membatasi model untuk mampu mengenali kelas di luar dataset pelatihan. Keterbatasan ini mendorong pengembangan pembelajaran inkremental yang bertujuan menambah pengetahuan baru secara bertahap sembari mempertahankan pengetahuan yang telah diperoleh model tanpa pelatihan ulang secara keseluruhan [2], [3], [4].

*Class Incremental Learning* (CIL) hadir sebagai solusi yang memungkinkan model mengintegrasikan pengetahuan tentang kelas-kelas baru tanpa mengorbankan kemampuan identifikasi terhadap kelas-kelas sebelumnya. Tantangan utama dalam implementasi CIL adalah *catastrophic forgetting*, yaitu hilangnya kemampuan model dalam mempertahankan pengetahuan lama saat mempelajari informasi baru [5]. Untuk mengatasi masalah ini, beberapa strategi telah dikembangkan seperti pemanfaatan *Knowledge Distillation* [6], penggunaan *exemplar* [7], [8], dan kombinasi keduanya [9]. Namun, pendekatan tersebut belum sepenuhnya mengatasi kecenderungan bias klasifikasi terhadap kelas baru.

Bias Correction Method (BiC) yang diusulkan Wu dkk. [5] diimplementasikan dengan menambahkan *linear layer* pada akhir model untuk mengoreksi bias terhadap kelas baru. Pendekatan ini terbukti efektif untuk dataset besar seperti ImageNet (1000 kelas) dan MS-Celeb-1M (10.000 kelas), dengan peningkatan performa model sebesar 11,1% dan 13,2% dibandingkan algoritma iCaRL [9] dan EEIL secara berturut-turut pada skenario inkremen 20 kelas. Namun, efektivitas BiC pada inkremen kelas yang terbatas belum diteliti secara komprehensif, mengindikasikan adanya celah penelitian yang perlu dikaji.

Penelitian ini mengimplementasikan BiC dengan arsitektur ResNet-50 pada subset 20 kelas dari ImageNet-100 yang telah diseleksi berdasarkan nilai *F1-Score* dan jumlah citra berformat *grayscale*. Tujuan penelitian adalah mengevaluasi efektivitas BiC dalam memperbaiki performa klasifikasi dan mengurangi dampak *catastrophic forgetting* pada skenario inkremen kelas kecil, sehingga dapat memberikan kontribusi pada pengembangan sistem pembelajaran yang lebih berkelanjutan dan adaptif dalam konteks keterbatasan daya komputasi untuk mencapai kelas yang besar.

## 2 Penelitian Terkait

Metode CIL terus berkembang sejak 2016 sejalan dengan inovasi arsitektur model dan perkembangan parameter di dalamnya [10]. Metode-metode yang telah berkembang dikelompokkan sebagai *Knowledge Distillation*, *Data Replay*, dan *Model Rectify* [10]. Implementasi dari kelompok tersebut ialah algoritma *Learning Without Forgetting* untuk *Knowledge Distillation* [6], *Replay* [7], [8] untuk *Data Replay*, dan *Bias Correction Method* [5] untuk *Model Rectify*.

*Learning Without Forgetting* (LwF) yang dikembangkan oleh Li & Hoiem menjadi pelopor perkembangan CIL dengan memperkenalkan teknik *Knowledge Distillation* dan skema *Teacher-*

*Student* [6]. *Knowledge Distillation* berupaya menyalin pengetahuan model lama (Teacher) ke model baru (Student) dengan menyamakan *output* kedua model dalam sebuah pelatihan. Implementasi LwF menggunakan arsitektur ResNet-18 dengan pelatihan hanya menggunakan data kelas baru mampu menghasilkan skor akurasi 53% pada kasus ImageNet100 dengan 20 inkremental, menunjukkan peningkatan signifikan dibandingkan dengan *finetune* standar [10].

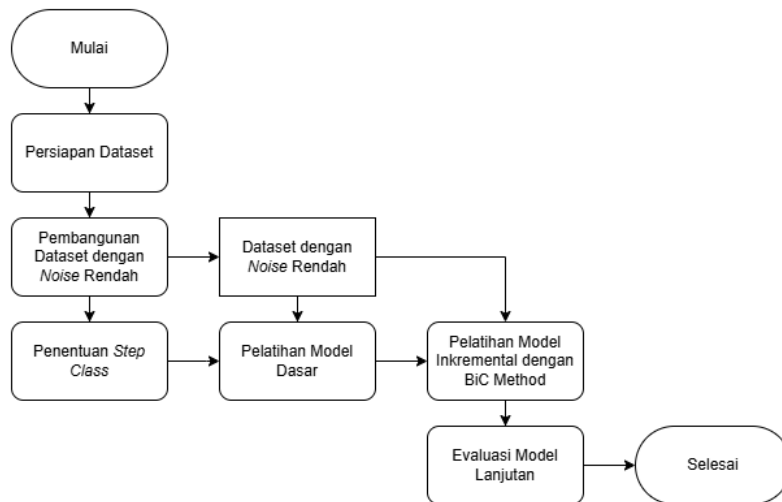
Perkembangan selanjutnya dalam CIL adalah penggunaan *exemplar*, yakni penyisipan sejumlah data kelas lama yang digabungkan dengan data kelas baru, yang dikenal dengan metode *Replay* [7], [8]. Teknik *Replay* yang diimplementasikan menggunakan arsitektur ResNet-18 menghasilkan skor akurasi 48,76% pada kasus ImageNet100 dengan 20 inkremental, menunjukkan kinerja yang mendekati performa LwF meskipun menggunakan pendekatan yang berbeda [10].

Integrasi *Knowledge Distillation* dan penggunaan *exemplar* melahirkan metode iCaRL yang dikembangkan oleh Rebuffi dkk. [9]. iCaRL membawa inovasi dalam pemilihan *exemplar* dengan merata-rata nilai *pixel* dari suatu kelas dan mengambil sejumlah gambar dengan kedekatan terhadap rata-rata yang merepresentasikan kelas tersebut. Selain itu, iCaRL menggunakan *Nearest-Mean-of-Example* (NME) tanpa *softmax* untuk menentukan hasil klasifikasi. Implementasi iCaRL dengan arsitektur ResNet-18 menunjukkan peningkatan kinerja dengan skor akurasi mencapai 61,50% pada dataset ImageNet100 dengan 20 inkremental [10], menjadikannya acuan bagi penelitian selanjutnya meskipun masih terdapat bias terhadap kelas yang terakhir dipelajari.

Untuk mengatasi bias terhadap kelas terakhir, Wu dkk. mengajukan Bias Correction Method (BiC) [5]. BiC diimplementasikan dengan menempatkan sebuah linear layer pada akhir model yang bertujuan mengurangi *logits* kelas baru sebelum diprobabilitaskan dengan fungsi *softmax*. BiC memanfaatkan *Knowledge Distillation* dan *Data Replay* dalam fase pelatihan pertama, dengan perbedaan adanya penambahan *linear layer* pada akhir model serta pemilihan acak pada data gambar kelas lama. Penerapan BiC dengan arsitektur ResNet-18 menunjukkan peningkatan sebesar 8% dibandingkan metode iCaRL, dengan skor akurasi mencapai 66,92% pada dataset ImageNet100 dengan 20 inkremental [10].

### 3 Metode Penelitian

Metode penyelesaian masalah serta implementasi BiC dalam penelitian ini dilakukan dalam beberapa tahapan. Tahapan umum tersebut ditunjukkan pada Gambar 1.



Gambar 1 Diagram Alur Garis Besar Penelitian

Tahapan penelitian diawali dengan persiapan dataset, dilanjutkan proses pembangunan dataset dengan *noise* rendah untuk menyerupai komposisi dataset penelitian dalam tinjauan pustaka, dilanjutkan dengan penentuan *step class*, pelatihan model dasar, pelatihan model inkremental dengan BiC, dan diakhiri dengan evaluasi model.

### 3.1 Persiapan dataset

Dataset yang digunakan dalam penelitian ini adalah ImageNet100 yang bersumber dari *Kaggle*. Dataset ImageNet100 memiliki 1250 gambar per kelasnya dengan variasi format citra dari sisi warna dan ukuran *pixel*. Variasi tersebut membutuhkan *preprocessing* agar citra dapat diolah oleh model. *Preprocessing* yang digunakan berupa standarisasi ukuran citra menjadi 192x192, penyeragaman format citra gambar menjadi RGB, normalisasi piksel, serta diakhiri dengan pembagian dataset menjadi 3 bagian yakni *Dataset Training*, *Dataset Validation*, dan *Dataset Testing*. Adapun untuk *Dataset Training* akan dikenakan *augmentasi* berupa *rotation* sebesar 20°, *random brightness* sebesar 25%, *horizontal flip*, dan *random contrast* sebesar 20%.

### 3.2 Pembangunan Dataset dengan Noise Rendah

Dataset ImageNet100 yang didapatkan akan dievaluasi agar menghasilkan performa CIL yang maksimal. Suatu kelas dikatakan memiliki *noise* rendah bila memiliki nilai *F1-Score* diatas 0,88 pada pelatihan 5 kelas dalam suatu model dan memiliki jumlah gambar berformat *grayscale* yang kurang dari 10 buah. Model yang digunakan untuk mengevaluasi dataset adalah ResNet-50 dengan lanjutan arsitektur layer *GlobalAveragePooling2D* yang dilanjutkan dengan satu lapisab *Dense* layer sebesar 512 neuron, satu lapisan *BatchNormalization*, dan diakhiri dengan satu lapisan *Dense* layer sebesar 5 neuron yang menggunakan fungsi aktivasi *softmax*. Model tersebut dilatih menggunakan *hyperparameter learning rate*  $10^{-4}$  dan *optimizer* Adam dan menghasilkan 42 kelas dengan *noise* rendah yang disajikan pada Tabel 1.

Tabel 1 Daftar Kelas dengan Noise Rendah

Dataset dengan Noise Rendah						
n01440764	n01601694	n01770081	n01806143	n01843383	n01950731	n02013706

n01443537	n01667778	n01770393	n01818515	n01860187	n01955084	n02018207
n01531178	n01695060	n01773157	n01820546	n01910747	n01986214	n02027492
n01537533	n01698640	n01774750	n01824575	n01914609	n02006656	n02037110
n01582220	n01751748	n01795545	n01828970	n01924916	n02007558	n02051845
n01592084	n01756291	n0179848	n01833805	n01930112	n02011460	n02077923

### 3.3 Penentuan Step Class

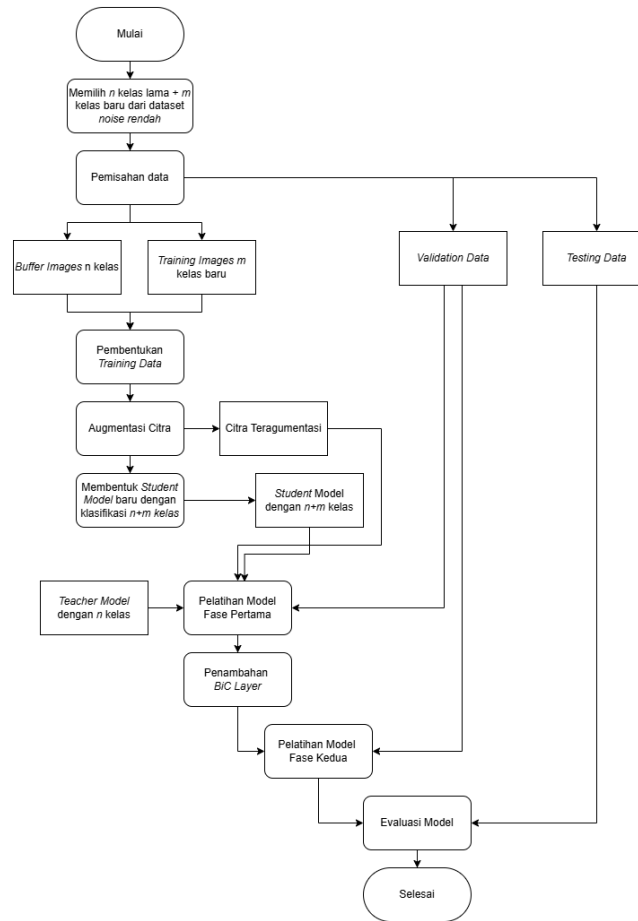
*Step class* adalah parameter pelatihan yang menentukan penambahan jumlah kelas per iterasi. Keterbatasan komputasi membatasi deteksi maksimal 20 kelas, sehingga digunakan faktor bilangan 20 sebagai *step class* yakni 2, 4, dan 5. *Step class* berpengaruh pada ingatan model melalui parameter *Lambda* dalam *Knowledge Distillation*, jumlah *exemplar* per iterasi, dan jumlah iterasi pelatihan inkremental. Nilai *step class* yang lebih tinggi mengurangi pembobotan untuk mempertahankan pengetahuan kelas lama, mengurangi jumlah *exemplar* per kelas, dan menurunkan total pelatihan yang diperlukan untuk mencapai klasifikasi kelas maksimal.

### 3.4 Pelatihan Model Dasar

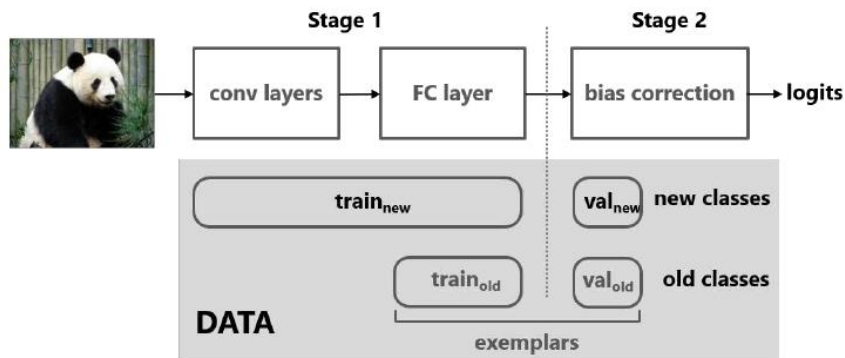
Proses pelatihan model dasar hanya dilakukan terhadap  $n$  kelas pertama dengan proses implementasi metodologi standar dalam *image classification*. Model dasar dilatih menggunakan data folder *training* yang telah teraugmentasi dengan alur pelatihan model dasar. Model dasar memiliki arsitektur yang serupa dengan model untuk membentuk dataset dengan noise rendah. Namun, model dasar memiliki perbedaan utama pada *hyperparameter* yang digunakan yakni *learning rate* sebesar  $10^{-5}$  dan *EarlyStopping* dengan parameter henti pada *validation loss*. Pelatihan model dasar memegang peranan penting dalam menjaga pengetahuan kelas-kelas terdahulu yang mana pengetahuan tersebut akan terdegradasi seiring bertambahnya kelas baru. Oleh karena itu, optimalisasi proses pelatihan model dasar berperan krusial dalam mempertahankan integritas pengetahuan kelas-kelas terdahulu selama fase-fase pelatihan inkremental berikutnya.

### 3.5 Pelatihan Model Inkremental

Proses pelatihan model inkremental akan menerapkan metode BiC yang membagi pelatihan menjadi dua fase. Fase pertama akan berfokus untuk membentuk *classifier* model baru yang mampu menangani  $n + m$  kelas serta menyalin pengetahuan dari model lama yang hanya mampu menangani  $n$  kelas. Fase kedua akan berfokus pada pengoreksian *bias* terhadap kelas baru yang ditimbulkan dari *imbalance training* saat pelatihan fase pertama. Alur pelatihan model inkremental tergambar pada Gambar 2 dengan visualisasi kedua fase pelatihan pada Gambar 3.



Gambar 2 Alur Pelatihan Inkremental Model dengan BiC Method



Gambar 3 Fase Pelatihan dalam BiC Method

Fase pertama pelatihan menggunakan skema *Teacher-Student* dengan *Knowledge Distillation* sebagaimana ditunjukkan pada Gambar 4. Data pelatihan menggabungkan *buffer images* dengan yang memiliki 2000 gambar terdistribusi merata pada  $n$  kelas lama dengan seluruh data pelatihan dari  $m$  kelas baru. Metode BiC mengimplementasikan *Knowledge Distillation Loss* ( $L_{KD}$ ) yang merupakan kombinasi dari *Distillation Loss* dan *Categorical Crossentropy Loss* dengan pembobotan berdasarkan Lambda ( $\lambda$ ) yakni rasio antara  $n$  kelas lama dan  $m$  total kelas. *Distillation Loss* ( $L_D$ ) mentransfer pengetahuan dari model lama (*teacher*) ke model baru (*student*) menggunakan output *logits* dari kedua model ( $\hat{\pi}_{k_{teacher}}, \hat{\pi}_{k_{student}}$ ) untuk  $n$  kelas lama yang dibagi dengan *temperature*. Sementara itu, *Categorical Crossentropy* ( $L_{CE}$ ) digunakan untuk mempelajari kelas-kelas baru dengan membandingkan distribusi kelas sebenarnya dan prediksi.



$$\lambda = \frac{n}{m} \quad (1)$$

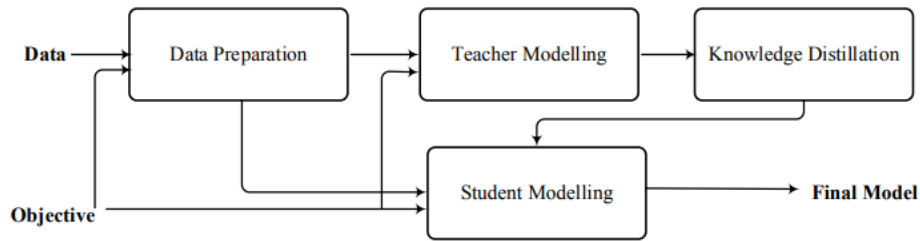
$$\hat{\pi}_k(x) = \frac{e^{\frac{z_k(x)}{T}}}{\sum_{j=1}^N e^{\frac{z_j(x)}{T}}} \quad (2)$$

$$\pi_k(x) = \frac{e^{z_k(x)}}{\sum_{j=1}^N e^{z_j(x)}} \quad (3)$$

$$L_D = \sum_{x \in \mathcal{X}^n \cup \mathcal{X}^m} \sum_{k=1}^n \hat{\pi}_{k_{teacher}}(x) \log[\hat{\pi}_{k_{student}}(x)] \quad (4)$$

$$L_{CE} = \sum_{x \in \mathcal{X}^n \cup \mathcal{X}^m} \sum_{k=1}^{n+m} -\delta_k \log[\pi_{k_{student}}(x)] \quad (5)$$

$$L_{KD} = \lambda L_D + (1 - \lambda) L_{CE} \quad (6)$$



Gambar 4 Skema Pelatihan Fase Pertama BiC Method dengan *Teacher-Student*

Setelah pelatihan fase pertama selesai, sebuah *linear layer* berparameter  $\alpha$  ( $\alpha$ ) dan  $\beta$  ( $\beta$ ) yang ditambahkan di akhir model yang dinamai BiC Layer. BiC Layer berfungsi mengoreksi nilai *logits* dari *student model* sebelum dikenai fungsi *softmax* [10]. Nilai *logits* yang terkena terkoreksi ( $q_k$ ) hanyalah kelas yang baru dipelajari saja sesuai dengan persamaan 7. Pelatihan fase kedua akan memanfaatkan sebagian data latih yang tidak digunakan pada fase pertama untuk mengoreksi *bias* terhadap kelas baru. Pelatihan fase kedua akan memanfaatkan *categorical crossentropy loss* untuk mengoreksi bobot  $\alpha$  dan  $\beta$  milik BiC Layer.

$$q_k = \begin{cases} o_k, & 1 \leq k \leq n \\ \alpha * o_k + \beta, & n + 1 \leq k \leq n + m \end{cases} \quad (7)$$

### 3.6 Evaluasi Model

Evaluasi dilakukan untuk mengukur kemampuan model sesuai dengan *hyperparameter* dan konfigurasi penggunaan BiC Layer pada masing-masing fase pelatihan terhadap data uji dari ImageNet100. Evaluasi model dilakukan pasca pelatihan fase kedua selesai di masing-masing *step size* untuk mendapatkan nilai akurasi akhir model saat klasifikasi 20 kelas, *F1-Score* model saat klasifikasi 20 kelas, *average accuracy* ( $\bar{A}$ ), dan *forgetting rate* ( $F$ ) pada masing-masing skenario. Metrik *average accuracy* serta *forgetting rate* mampu memberikan informasi lebih khususnya untuk hal kestabilan model serta efek *catastrophic forgetting* di masing-masing skenario pelatihan [11]. Persamaan yang digunakan dalam perhitungan kinerja model dijelaskan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

$$Accuracy = \frac{\sum TP + \sum TN}{N} \quad (11)$$

$$\bar{A} = \frac{1}{B} \sum_{b=1}^B A_b \quad (12)$$

$$f_{j,k} = \max_{l \in \{1, \dots, k-1\}} a_{l,j} - a_{k,j} \quad (13)$$

$$F = \frac{1}{B-1} \sum_{j=1}^{B-1} f_{j,B} \quad (14)$$

#### 4 Hasil dan Pembahasan

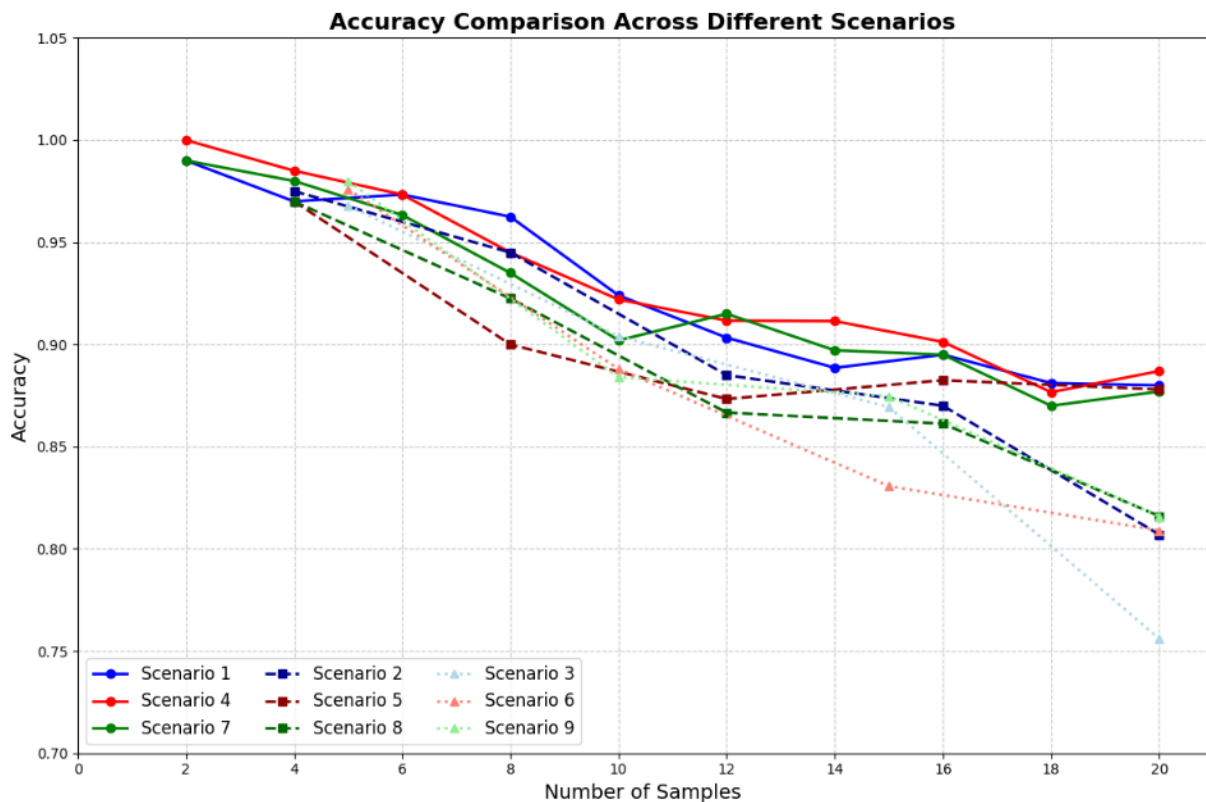
Hasil dan pembahasan berisi pembahasan analisis dari hasil skenario pelatihan pada penelitian penerapan *robust incremental learning* dengan ResNet-50 dan Bias Correction Layer untuk Klasifikasi Citra. Pembahasan akan berfokus pada *hyperparameter* berupa penggunaan BiC Layer pada fase pelatihan di masing-masing *step size* serta melihat dampak penerapan BiC Layer terhadap metrik evaluasi khususnya *average accuracy* dan *forgetting rate*. Kedua metrik tersebut mampu untuk mengukur efektivitas model dalam mempertahankan pengetahuan sebelumnya sembari mempelajari konsep baru [11] dengan data pada Tabel 3 dan visualisasi *average accuracy* pada Gambar 5.

Tabel 2 Detail Skenario Pelatihan

No.	Skenario	Resolusi Citra	BiC Layer Fase 1	BiC Layer Fase 2	Step class
1.	<i>Skenario_1</i>	192x192	Tidak	Tidak	2
2.	<i>Skenario_2</i>	192x192	Tidak	Tidak	4
3.	<i>Skenario_3</i>	192x192	Tidak	Tidak	5
4.	<i>Skenario_4</i>	192x192	Tidak	Ya	2
5.	<i>Skenario_5</i>	192x192	Tidak	Ya	4
6.	<i>Skenario_6</i>	192x192	Tidak	Ya	5
7.	<i>Skenario_7</i>	192x192	Ya	Ya	2
8.	<i>Skenario_8</i>	192x192	Ya	Ya	4
9.	<i>Skenario_9</i>	192x192	Ya	Ya	5

Tabel 3 Hasil Pengujian Model

No	Skenario	Akurasi Validasi	F1-Score	Average Accuracy	Forgetting Rate
1.	Skenario_1	88%	0,8809	0,92678	5,17%
2.	Skenario_2	80,7%	0,8209	0,89640	8,94%
3.	Skenario_3	75,6%	0,7640	0,87433	12,13%
4.	Skenario_4	88,7%	0,8874	0,93133	4,67%
5.	Skenario_5	87,8%	0,8782	0,90076	4,50%
6.	Skenario_6	80,9%	0,8205	0,87591	8,53%
7.	Skenario_7	87,7%	0,8781	0,92244	4,4%
8.	Skenario_8	81,6%	0,8215	0,88728	8,50%
9.	Skenario_9	81,6%	0,8240	0,88867	8,47%



Gambar 5 Average Accuracy pada masing-masing Skenario

Sesuai dengan pernyataan Chaudhry dkk. [11], nilai *forgetting loss* yang kecil mempunyai arti model mampu untuk mempertahankan pengetahuan kelas lama dengan baik dan mencegah fenomena *catastrophic forgetting*. Di sisi lain, *average accuracy* juga memberikan informasi kestabilan model dalam menjaga akurasi yang dimilikinya terhadap semua kelas [11].

#### 4.1 Analisa 2 Step Class

Hasil analisis pada Tabel 3 menunjukkan bahwa penerapan *step class* kecil (2 kelas) cukup efektif dalam menangani *catastrophic forgetting* bahkan tanpa BiC Layer, dengan *forgetting rate* hanya 5.17% pada Skenario 1. Pada ukuran inkremen ini, BiC Layer memberikan dampak bervariasi dengan Skenario 4 menunjukkan peningkatan baik pada *average accuracy* (93.133%) maupun penurunan *forgetting rate* (4.67%), sementara Skenario 7 mengalami *overcorrection* yang menurunkan *average accuracy* menjadi 0,92244 meskipun mencapai *forgetting rate* terendah (4.4%). Fenomena ini menunjukkan bahwa pada *step class* kecil, efektivitas model lebih dipengaruhi oleh faktor-faktor seperti frekuensi pelatihan tinggi per kelas, penggunaan exemplar yang efisien, dan dominasi *Distillation Loss* yang mencapai 0.90 pada pelatihan akhir. Ketiganya memberikan dampak secara alamiah untuk membantu model mempertahankan pengetahuan kelas lama.

#### 4.2 Analisa 4 Step Class

Analisis komparatif pada Tabel 3 memperlihatkan bahwa implementasi *step size* 4 mulai menunjukkan fenomena *catastrophic forgetting* yang signifikan, dengan nilai *forgetting rate* mencapai 8,94% pada Skenario 2 (tanpa BiC Layer). Nilai tersebut menandakan ketidakstabilan model dalam

mempertahankan pengetahuan kelas sebelumnya. Temuan tersebut didukung dengan perbedaan antara rerata akurasi Skenario 2 (0,89640) dan akurasi validasi akhir Skenario 2 yang lebih rendah (80,7%). Faktor tersebut didukung oleh frekuensi pelatihan yang lebih terbatas yakni hanya 5 sesi dan proporsi data exemplar yang berkurang menjadi 125 per kelas pada pelatihan terakhir. Sebaliknya, implementasi BiC Layer membawa dampak yang beragam. Pada Skenario 5, terdapat perbaikan substansial dengan peningkatan akurasi validasi menjadi 87,8% dan penurunan *forgetting rate* menjadi 4,50%. Namun, pada Skenario 8, hasil yang didapatkan tidak berbeda jauh dengan Skenario 2. Hal tersebut menunjukkan adanya *overcorrection* pada penerapan BiC layer di kedua fase pelatihan. Walaupun begitu, pada *step class* menengah dapat disimpulkan bahwa BiC Layer berperan krusial dalam memitigasi degradasi pengetahuan dan menstabilkan performa model, sebagaimana tervisualisasi pada Gambar 5 yang menunjukkan kurva penurunan akurasi yang lebih moderat dibandingkan dengan model tanpa BiC Layer.

#### 4.3 Analisa 5 Step Class

Analisis data pada Tabel 3 menunjukkan korelasi positif antara ukuran *step class* dengan tingkat *forgetting rate*. Temuan tersebut terbukti dari peningkatan nilai *forgetting rate* dari 5,17% (Skenario 1) menjadi 8,94% (Skenario 2) pada model tanpa BiC Layer. Sama seperti Skenario 2, besarnya *forgetting rate* pada 5 *step class* disebabkan oleh rendahnya frekuensi pelatihan (hanya 4 sesi) dan minimnya data exemplar per kelas (133 gambar pada pelatihan terakhir). Implementasi BiC Layer memberikan dampak positif signifikan, ditunjukkan oleh penurunan *forgetting rate* pada Skenario 6 dan 9 menjadi 8,53% dan 8,47% dari nilai awal yang sebesar 12,13% pada Skenario 3. Pada kasus 5 *step class*, penerapan BiC Layer pada kedua fase pelatihan menghasilkan performa yang superior dibandingkan implementasi hanya pada fase kedua, sejalan dengan implementasi oleh Wu dkk. (2019) dan Zhou dkk. (2024) yang menggunakan *step class* 10 dan 25. Klaim tersebut didukung dengan visualisasi pada Gambar 4.7 yang mendemonstrasikan mitigasi degradasi akurasi selama pembelajaran inkremental.

### 5 Kesimpulan

Penelitian ini menghasilkan beberapa kesimpulan penting mengenai penerapan *Class Incremental Learning* (CIL) dengan arsitektur ResNet-50 dan Bias Correction Layer pada klasifikasi citra menggunakan dataset 20 kelas dengan *noise* rendah dari ImageNet100. Hasil analisis menunjukkan bahwa ResNet-50 dapat diimplementasikan secara efektif untuk CIL dengan kemampuan mempertahankan pengetahuan kelas lama sambil mengakomodasi kelas baru, dibuktikan dengan *forgetting rate* yang konsisten di bawah 13%. Performa model dapat dioptimalkan melalui konfigurasi *hyperparameter* yang tepat, khususnya kombinasi BiC Layer dan *step class*, dengan pengaruh BiC Layer yang lebih signifikan pada *step class* berukuran besar dibandingkan pada *step class* kecil. Konfigurasi optimal ditemukan pada model dengan BiC Layer yang diterapkan hanya pada fase kedua dengan 2 *step class*, menghasilkan *F1-score* 0,887, akurasi 0,8874, *average accuracy* 0,93133, dan *forgetting rate* sebesar 4,67%. Hasil diatas membuktikan bahwa penerapan BiC Layer yang selektif pada fase pelatihan dengan *step class* kecil dapat menghasilkan model klasifikasi yang lebih efisien dan akurat untuk kasus penggunaan spesifik dalam penelitian ini.

### Ucapan Terima Kasih

Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada Ibu Rismiyati, B.Eng., M.Cs, dan Bapak Dr. Indra Waspada S.T., M.T.I., atas bimbingan, dukungan, serta masukan berharga yang diberikan selama proses penyusunan artikel ini. Penulis juga menghargai segala bentuk kontribusi dari berbagai pihak, baik secara langsung maupun tidak langsung, yang turut membantu dalam penyelesaian penulisan ini.

## Daftar Pustaka

- [1]D. Silver et al., “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, pp. 484–489, Feb. 2016, doi: 10.1038/nature16961.
- [2]G. van de Ven, T. Tuytelaars, and A. Tolias, “Three types of incremental learning,” *Nat Mach Intell*, vol. 4, pp. 1–13, Feb. 2022, doi: 10.1038/s42256-022-00568-3.
- [3]M. Delange et al., “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Trans Pattern Anal Mach Intell*, p. 1, 2021, doi: 10.1109/tpami.2021.3057446.
- [4]S. T. Grossberg, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, vol. 70. in *Boston Studies in the Philosophy and History of Science*, vol. 70. Springer, 2012. doi: 10.1007/978-94-009-7758-7.
- [5]Y. Wu et al., “Large Scale Incremental Learning,” 2019. [Online]. Available: <https://arxiv.org/abs/1905.13260>
- [6]Z. Li and D. Hoiem, “Learning without Forgetting,” *IEEE Trans Pattern Anal Mach Intell*, vol. 40, no. 12, pp. 2935–2947, 2018, doi: 10.1109/TPAMI.2017.2773081
- [7]J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, “Rainbow Memory: Continual Learning with a Memory of Diverse Samples,” 2021. [Online]. Available: <https://arxiv.org/abs/2103.17230>
- [8]H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual Learning with Deep Generative Replay,” 2017. [Online]. Available: <https://arxiv.org/abs/1705.08690>
- [9]S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” 2017. [Online]. Available: <https://arxiv.org/abs/1611.07725>
- [10]D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, “Class-Incremental Learning: A Survey,” *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 12, pp. 9851–9873, Dec. 2024, doi: 10.1109/tpami.2024.3429383.
- [11]A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, “Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence,” in *Computer Vision – ECCV 2018*, Springer International Publishing, 2018, pp. 556–572. doi: 10.1007/978-3-030-01252-6\_33.