

**PERBANDINGAN KINERJA *DEEP LEARNING ALEXNET* DAN
EFFICIENTNET DENGAN *AUGMENTASI DATA* UNTUK IDENTIFIKASI
KUPU-KUPU**

(Skripsi)

Oleh

ATIKA ISTIQOMAH

1817051004



FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2022

**PERBANDINGAN KINERJA *DEEP LEARNING ALEXNET* DAN
EFFICIENTNET DENGAN *AUGMENTASI DATA* UNTUK IDENTIFIKASI
KUPU-KUPU**

Oleh

ATIKA ISTIQOMAH

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRAK

PERBANDINGAN KINERJA *DEEP LEARNING ALEXNET* DAN *EFFICIENTNET* DENGAN AUGMENTASI DATA UNTUK IDENTIFIKASI KUPU-KUPU

Oleh

ATIKA ISTIQOMAH

Kupu-kupu merupakan makhluk hidup yang berpengaruh dalam ekosistem. Hewan ini membantu dalam proses penyerbukan pada tanaman, serta dapat menjaga keseimbangan ekosistem sekitar. Identifikasi terhadap kupu-kupu diperlukan untuk membantu orang awam atau pengunjung taman kupu-kupu, dalam menambah pengetahuannya terkait macam-macam kupu-kupu. Proses identifikasi dimudahkan dengan adanya teknologi *deep learning*, serta munculnya algoritma *Convolutional Neural Network* (CNN) dan beragam arsitektur yang dibentuk oleh algoritma tersebut. Arsitektur *AlexNet* dan *EfficientNet* digunakan untuk membantu proses klasifikasi kelas kupu-kupu. Kedua arsitektur digunakan, karena dianggap sebagai arsitektur yang mudah diimplementasikan serta mudah untuk dilakukan perubahan, tetapi dapat menghasilkan akurasi yang baik pada beberapa penelitian terdahulu. Penelitian dilakukan menggunakan 800 gambar kupu-kupu dengan 8 kelas, masing-masing kelas terdiri dari 100 gambar. Percobaan menggunakan 2 macam mesin, yaitu *google colab (online)* dan komputer dengan GPU NVIDIA Tesla K20 (*offline*). Hasil yang diberikan ialah, perbandingan kedua arsitektur pada masing-masing mesin. Hasil percobaan menggunakan *google colab* menghasilkan akurasi terbesar pada arsitektur *EfficientNet*, yaitu sebesar 99.69% dengan *augmentasi flip vertical*. Percobaan menggunakan mesin tesla didapatkan akurasi terbesar pada kedua arsitektur, yaitu sebesar 99.37% dengan *augmentasi flip horizontal*.

Kata Kunci: kupu-kupu, *deep learning*, akurasi

Judul Skripsi : **PERBANDINGAN KINERJA *DEEP LEARNING ALEXNET* DAN *EFFICIENTNET* DENGAN *AUGMENTASI DATA* UNTUK IDENTIFIKASI KUPU-KUPU**

Nama Mahasiswa : **Atika Istiqomah**

Nomor Pokok Mahasiswa : 1817051004

Program Studi : S1 Ilmu Komputer

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing



Rico Andrian, S.Si., M.Kom.
NIP 19750627 200501 1 001

2. Ketua Jurusan Ilmu Komputer



Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

Ketua

: **Rico Andrian, S.Si., M.Kom.**



Penguji

Bukan Pembimbing : **Dr. rer. nat. Akmal Junaidi, M.Sc.**



Penguji

Bukan Pembimbing : **Didik Kurniawan, S.Si., M.T.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Supto Dwi Yuwono, S.Si., M.T.
NIP. 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **11 Agustus 2022**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Atika Istiqomah

NPM : 1817051004

Dengan ini menyatakan bahwa skripsi saya yang berjudul “**Perbandingan Kinerja *Deep Learning AlexNet* dan *EfficientNet* dengan *Augmentasi Data* untuk Identifikasi Kupu-Kupu**” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 19 Agustus 2022



Atika Istiqomah

NPM. 1817051004

RIWAYAT HIDUP



Penulis lahir di Bandar Lampung, 03 April 2000, dilahirkan sebagai anak keempat dari empat bersaudara. Pendidikan yang telah ditempuh oleh penulis diantaranya, menyelesaikan pendidikan dasar di SD Al-Azhar 2 Bandar Lampung pada tahun 2012. Penulis menyelesaikan pendidikan menengah pertama di SMP IT Ar-Raihan Bandar Lampung pada tahun 2015.

Penulis melanjutkan pendidikan menengah atas di SMA S Al Kautsar Bandar Lampung dan lulus pada tahun 2018. Perjalanan pendidikan penulis dilanjutkan dengan terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2018 melalui jalur SNMPTN. Penulis turut aktif mengikuti beberapa kegiatan di kampus selama menjadi mahasiswa, antara lain.

1. Menjadi anggota bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) periode 2019/2020.
2. Menjadi anggota bidang Informasi dan Komunikasi (Infokom) Rohani Islam (Rois) FMIPA Unila periode 2019/2020.
3. Menjadi anggota bidang Kaderisasi Bina Rohani Mahasiswa (Birohmah) Unila periode 2019/2020.

4. Menjadi Bendahara Himakom periode 2020/2021.
5. Menjadi Sekretaris Komisi 2 Keuangan Dewan Perwakilan Mahasiswa (DPM) FMIPA Unila periode 2021/2022.
6. Menjadi Asisten Dosen di Jurusan Ilmu Komputer tahun 2019 hingga 2021.
7. Melaksanakan Kerja Praktik pada bulan Februari periode 2019/2020 di PDDI-LIPI.
8. Melaksanakan Kuliah Kerja Nyata (KKN) di Kelurahan Bakung, Kecamatan Teluk Betung Barat, Kota Bandar Lampung periode 2020/2021.
9. Memenangkan juara ke-3 dalam perlombaan LKTI-A Nasional FPPI FKIP Unila pada tahun 2020.

MOTO

1. "Cara terbaik untuk memprediksi masa depan adalah dengan mempersiapkannya"

(Alan Kay)

2. "Kesulitan dimaksudkan untuk membangkitkan, bukan mengecilkan hati. Semangat manusia tumbuh kuat melalui sebuah masalah"

(William E. Channing)

3. "Membandingkan hidup dan kesuksesanmu dengan yang lain, hanya akan memberatkan jalan menuju tujuanmu sendiri"

(Author)

4. "Kau boleh merasa lelah untuk mencoba, tetapi jangan sampai menyerah untuk mencoba"

(Author)

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji dan syukur atas kehadiran Allah SWT atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tersampaikan kepada Nabi Muhammad SAW.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku

Yang selalu mendoakan segala hal baik, selalu mendukung dan memberikan segala yang terbaik untukku. Terima kasih telah mendidik dan membesarkanku dengan cara terbaik kalian, dan selalu dipenuhi kasih sayang. Terima kasih atas dukungan dan pengorbanan kalian yang belum bisa terbalaskan, karena begitu tak terkira jika dirasakan.

Kakak-Kakak Tercinta

Yang selalu memberikan dukungan, motivasi, dan saran atas segala kondisi dan keadaan yang sedang ku hadapi. Kalian sangat berarti dalam membangkitkan semangatku.

Seluruh Keluarga Besar Ilmu Komputer 2018

Yang selalu memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat bernaung mengemban semua ilmu untuk menjadi bekal hidup.

SANWACANA

Puji syukur kehadiran Allah SWT atas berkah, rahmat dan hidayat-Nya, serta petunjuk dan pedoman dari Rasulullah Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perbandingan Kinerja *Deep Learning AlexNet* dan *EfficientNet* dengan *Augmentasi Data* untuk Identifikasi Kupu-Kupu” dengan baik dan lancar.

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan besar dalam menyusun skripsi ini, antara lain.

1. Kedua orang tua serta kakak-kakak tercinta yang selalu memberi dukungan, do'a, semangat, motivasi, dan kasih sayang yang luar biasa tak terhingga. Semua yang telah kalian berikan tidak akan pernah mampu untukku balas. Semoga Allah SWT selalu memberikan kebahagiaan dan keberkahan dalam kehidupan kalian di dunia dan akhirat.
2. Bapak Rico Andrian, S.Si., M.Kom. sebagai pembimbing utama yang telah memberikan segala arahan, ide, motivasi, kritik serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik
3. Bapak Didik Kurniawan, S.Si., M.T. sebagai pembahas yang telah memberikan masukan bermanfaat dalam perbaikan skripsi ini, serta selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
4. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. sebagai pembahas yang telah memberikan kritik dan masukan yang sangat membantu dalam perbaikan skripsi ini, serta selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Bapak Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T. selaku Dekan FMIPA Universitas Lampung.

6. Ibu Ade Nora Maela, Bang Zainuddin dan Mas Nofal yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu bermanfaat dan pengalaman dalam hidup untuk menjadi lebih baik.
8. Sahabat-sahabat saya Wulan Warohma dan Noni Susanti yang sudah seperti keluarga sendiri, senantiasa menjadi tempat berkeluh kesah, selalu menemani di segala kondisi, dan tak luput dalam memberikan semangat, dukungan, serta doa. Terima kasih atas segala warna dan lika liku kehidupan yang sangat bernilai dari kalian.
9. Teman-teman yang selalu ada sejak awal perkuliahan, Anisa Raden, Mutiara Widdi, dan Livia Ayu, yang tak pernah lupa untuk saling mendukung, menemani, dan menyemangati satu sama lain.
10. Teman-teman yang selalu menghibur disela kehidupan kampus, Andira Rahma, Aniisah Nurfaizah, dan Ramona Rahmawati.
11. Teman-teman seperjuangan grup Spartan, Kak Kasandra Cahyani, Kak Eggi Amandara Sari, Kak Dian Indah Lestari, Kak Noverina Ika Tama, Kak Reda Meiningtiyas, Kak Zuhri Nopriyanto, dan Pandi Barep Arianza, yang selalu mendoakan dan saling membantu dalam segala hal baik.
12. Keluarga Ilmu Komputer 2018 yang tidak bisa penulis sebut satu persatu. Keluarga kedua penulis, rekan kelompok, rekan diskusi, rekan bercanda, dan telah memberi arti dan warna serta pengalaman tak ternilai semasa duduk di bangku kuliah.
13. Teman-teman Himakom yang sudah mengajarkan banyak hal dalam berorganisasi, memberikan banyak pengalaman.
14. Teman-teman Harmoni yang sudah mengajarkan banyak hal dalam segala bentuk keharmonian yang penuh warna.
15. Teman-teman organisasi lainnya yang telah memberikan banyak pengalaman dalam keorganisasian kampus maupun hidup, dan telah mengajarkan banyak hal yang bermanfaat untuk kedepannya.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, semoga skripsi ini membawa manfaat dan keberkahan bagi semua civitas Ilmu Komputer Universitas Lampung aamiin ya rabbal aalamiin.

Bandar Lampung, 19 Agustus 2022



Atika Istiqomah

NPM. 1817051004

DAFTAR ISI

	Halaman
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	viii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan.....	6
1.5 Manfaat.....	6
II. TINJAUAN PUSTAKA.....	7
2.1 Kupu-Kupu	7
2.2 <i>Deep Learning</i>	8
2.3 Convolutional Neural Network	9
2.4 <i>AlexNet</i>	14
2.5 <i>EfficientNet</i>	14
2.6 <i>Augmentasi Data</i>	15
2.7 <i>Hyperparameter</i>	18
2.8 <i>Confusion Matrix</i>	19
III. METODOLOGI PENELITIAN.....	22
3.1 Waktu dan Tempat	22

3.2 Bahan dan Alat	22
3.3 Metode.....	24
IV. HASIL DAN PEMBAHASAN.....	29
4.1 Metode Klasifikasi CNN	29
4.2 Pembahasan	43
V. SIMPULAN DAN SARAN.....	53
5.1 Simpulan.....	53
5.2 Saran	54
DAFTAR PUSTAKA	55

DAFTAR TABEL

Tabel	Halaman
1. Metrik utama	20
2. Nama kelas dan jumlah <i>dataset</i> awal	23
3. Nama kelas dan jumlah <i>dataset flip horizontal</i>	25
4. Nama kelas dan jumlah <i>dataset flip vertical</i>	26
5. <i>Hyperparameter</i> model	28
6. Hasil akurasi uji arsitektur AlexNet dengan mesin Colab	35
7. Hasil akurasi uji arsitektur AlexNet dengan mesin Tesla	35
8. Hasil akurasi arsitektur EfficientNetB4 menggunakan <i>colab</i>	39
9. Hasil akurasi arsitektur <i>EfficientNetB4</i> menggunakan <i>tesla</i>	40
10. <i>Confusion matrix</i> arsitektur <i>AlexNet</i> dengan mesin tesla.....	44
11. <i>Precision, recall, f1-score</i> arsitektur <i>AlexNet</i> dengan mesin tesla.....	46
12. <i>Confusion matrix</i> arsitektur <i>EfficientNetB4</i> dengan augmentasi <i>flip vertical</i> pada mesin <i>colab</i>	50
13. <i>Precision, recall, f1-score</i> arsitektur <i>EfficientNetB4</i> dengan mesin <i>colab</i>	51

DAFTAR GAMBAR

Gambar	Halaman
1. Kupu-kupu (<i>Rhopalocera</i>).	7
2. Lapisan <i>Convolutional Neural Network</i> (CNN).	10
3. Lapisan di dalam <i>AlexNet</i>	14
4. Lapisan di dalam <i>EfficientNet</i>	15
5. Hasil <i>flip</i> gambar.	16
6. Hasil <i>rotation</i>	16
7. Hasil perubahan warna.	17
8. Hasil pergeseran gambar.	18
9. Metode penelitian.	24
10. Kronologi <i>batch-size</i>	31
11. Kronologi iterasi, <i>forward propagation</i> , dan <i>backward propagation</i>	32
12. Arsitektur <i>AlexNet</i>	33
13. Grafik perbandingan tingkat akurasi arsitektur <i>AlexNet</i> menggunakan <i>colab</i> . ..	37
14. Grafik perbandingan tingkat akurasi arsitektur <i>AlexNet</i> menggunakan tesla. ..	38
15. Arsitektur <i>EfficientNet</i>	39
16. Grafik perbandingan tingkat akurasi tiga kategori dengan arsitektur <i>EfficientNetB4</i>	41
17. Perbandingan mesin tesla dan mesin <i>colab</i> pada arsitektur <i>EfficientNetB4</i> . ..	42
18. Kemiripan (a) Kupu Batik Cap dan (b) Kupu Jarak.	48
19. Kemiripan (a) Kupu Hijau Biru dan (b) Kupu Harimau Kuning Hijau.	49
20. Kemiripan (a) Kupu Raja Limau dan (b) Kupu Pantat Merah.	52

I. PENDAHULUAN

1.1 Latar Belakang

Indonesia sebagai negara kepulauan memiliki tingkat endemisitas yang tinggi dalam penyebaran faunanya. Fauna dengan sebaran endemisitas tinggi ialah kupu-kupu, yang mencapai lebih dari 35% dari jumlah spesiesnya. Spesies kupu-kupu di Indonesia diperkirakan mencapai 2.500 spesies berdasarkan data buku “Bioekologi : Kupu-Kupu”. Kupu-kupu termasuk ke dalam *ordo Lepidoptera* lebih tepatnya termasuk ke dalam *subordo Rhopalocera*. *Ordo Lepidoptera* secara garis besar berperan dalam mempertahankan keseimbangan ekosistem dan memperkaya keanekaragaman hayati saat menjalankan perannya sebagai pollinator. Kupu-kupu memiliki peran penting dalam proses penyerbukan tanaman, yaitu proses ketika tanaman akan menghasilkan buah dan berkembang biak. Seekor kupu-kupu biasanya memiliki habitat dengan tumbuhan pakan yang cocok bagi spesies kupu-kupu. Habitat kupu-kupu yang mengalami perubahan dapat mempengaruhi komposisi spesies dan kelimpahan kupu-kupu, sehingga penting untuk dilakukan pengawasan dan perawatan terhadap habitat kupu-kupu dan kupu-kupu itu sendiri.

Taman kupu-kupu Gita Persada Lampung adalah salah satu habitat pelestarian dan tempat penangkaran kupu-kupu di Sumatra, lebih tepatnya di provinsi Lampung. Taman ini mengonservasi sekitar 180 spesies kupu-kupu dan terus meningkat setiap tahunnya seiring bertambahnya penelitian terkait kupu-kupu (Butterfly Park). Taman kupu-kupu Gita Persada menyediakan 60% keanekaragaman kupu-kupu khas Sumatra dengan warna, corak, dan pola yang indah (Kartini, et al., 2021). Keragaman kupu-kupu cukup sulit dikenali jika hanya melihat pola, warna, dan coraknya dengan kasat mata. Proses mengidentifikasi jenis kupu-kupu secara

manual cukup menghabiskan waktu dan tenaga. Kupu-kupu yang diidentifikasi secara manual juga dapat membahayakan kupu-kupu tersebut (Theivaprakasham, 2020). Mereka harus ditangkap secara manual oleh perangkap atau jaring untuk waktu yang cukup lama sampai proses identifikasi selesai dilakukan. Identifikasi atau pengenalan terhadap kupu-kupu diperlukan kemudahan, sehingga orang awam maupun pengunjung taman sejenis ini dapat dengan mudah mengenali jenis kupu-kupu ketika berkunjung. Kemudahan ini dibutuhkan untuk menambah pengetahuan maupun pengalaman dari orang awam (selain peneliti) terhadap pengenalan kupu-kupu. Identifikasi atau pengenalan pola kupu-kupu membutuhkan teknologi untuk memudahkan prosesnya. Teknologi *deep learning* ialah salah satu yang dapat digunakan berkaitan dengan pengenalan pola.

Deep Learning merupakan bagian dari *computer vision* yang sering digunakan untuk mengenali dan mengklasifikasi sebuah objek gambar. *Computer vision* menggabungkan antara *artificial intelligence* dan *deep learning*, sehingga menghasilkan bidang baru, yang khusus digunakan untuk melatih komputer agar dapat bekerja layaknya manusia ketika melihat, mengenali, dan menginterpretasikan sebuah objek gambar (Borugadda, et al., 2021). *Deep learning* bisa disebut sebagai tahap lanjut dari *machine learning*, karena lebih canggih dan lebih mendalam komputasinya dibandingkan *machine learning*. Komputasi di dalam *deep learning* memungkinkan mesin untuk dapat belajar sendiri dengan data yang mentah atau tanpa diberikan instruksi spesifik oleh manusia ketika terjadi kesalahan prediksi. *Deep learning* menggunakan konsep cara kerja dari jaringan saraf (*neural network*) otak manusia dalam menangkap sebuah gambar (Almryad & Kutucu, 2020). *Neural network* dapat menghubungkan ciri-ciri yang ditangkap dan disimpan melalui proses *training* dengan gambar yang diuji, sehingga dapat mengidentifikasi gambar saat proses uji dilakukan. Cara kerja *deep learning* tersebut tersusun rapih membentuk sebuah algoritma yang dapat terus berkembang seiring kebutuhan dan pemanfaatan teknologi *deep learning*.

Algoritma *deep learning* yang efektif digunakan salah satunya adalah *Convolutional Neural Network* (CNN). CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP), yang memanfaatkan banyak lapisan untuk menyimpan dan memroses ciri objek gambar. Proses utama CNN terbagi menjadi *feature extraction layer* dan *fully-connected layer* (Borugadda, et al., 2021). *Feature extraction layer* digunakan untuk mengambil ciri objek dalam gambar dan menyimpannya untuk digunakan pada *fully-connected layer*. *Fully-connected layer* digunakan dalam proses klasifikasi objek pada gambar. CNN memiliki beberapa arsitektur yang beragam, beberapa diantaranya ialah arsitektur *AlexNet* dan *EfficientNet*. *Alexnet* menggunakan proses dasar dari CNN yaitu, *feature extraction* dan *fully-connected layer*. Arsitektur ini juga memenangkan kompetisi ImageNet LSVRC pada tahun 2012 dalam kategori *image classification* (Krizhevsky, et al., 2012). *AlexNet* dapat menerima *input* atau masukan gambar berupa RGB. Arsitektur lainnya ialah *EfficientNet*, yang memiliki susunan berbeda dengan *AlexNet*. *EfficientNet* merupakan pengembangan yang dilakukan oleh CNN untuk mendapatkan akurasi terbaik, serta meningkatkan efisiensi model dengan mengurangi parameter yang dapat dilatih.

Penelitian sebelumnya terkait penggunaan arsitektur *AlexNet* digunakan untuk mengklasifikasikan penyakit pada daun kapas. Kategori atau kelas yang terdapat pada *dataset* sebanyak 4 kategori. Hasil *macro F1_score* terbesar terdapat di angka 94.92%. Hasil *macro F1_score* tersebut didapatkan dengan parameter berupa *batch size* sebesar 64, *learning rate* sebesar 0.001, *dropout* di angka 0.5, *momentum* di angka 0.9, *epochs* senilai 50, dan menggunakan *optimizer* SGD. *Macro F1_score* digunakan sebagai nilai hasil dari klasifikasi karena *dataset* dalam penelitian ini kurang seimbang atau jumlah gambar masing-masing kategori di dalam *datasetnya* tidak seimbang (Borugadda, et al., 2021).

Penelitian terkait penggunaan arsitektur *EfficientNet* B0-B7 digunakan untuk klasifikasi telah digunakan sebelumnya. *Dataset* yang digunakan terdiri dari 7 kelas kanker kulit dan jumlah *dataset* yang digunakan sebanyak 10015 gambar. Penelitian ini menghasilkan *EfficientNet* B4 sebagai model yang paling bagus

performanya. *EfficientNet* B4 menghasilkan nilai *F1 score* sebesar 87% dan nilai *Top-1 Accuracy* sebesar 87.91%, hasil tersebut berdasarkan parameter nilai *image size* 380x380, *batch size* senilai 8, *learning rate* senilai 0.001, dan menggunakan *optimizer* SGD. *Top-k Accuracy* sendiri artinya salah satu prediksi *top-k* berdasarkan probabilitas harus cocok dengan kelas sebenarnya dari gambar yang dianggap sebagai prediksi yang benar (Ali, et al., 2021).

Penelitian terdahulu memanfaatkan *augmentasi data* pada dataset kupu-kupu menggunakan *Convolutional Neural Network* (CNN). Dataset kupu-kupu yang digunakan sebanyak 6.000 gambar dari 4 jenis kupu-kupu. *Augmentasi* serta *Pre-processing* yang digunakan berupa *cropping*, *scaling*, dan *gray-scale*. Hasil akurasi sebesar 62.5% didapatkan melalui proses training menggunakan model CNN dan dengan nilai *epochs* 500 Model CNN (Bakri, et al., 2019).

Penelitian terdahulu mengenai identifikasi kupu-kupu pernah dilakukan menggunakan fitur deteksi tepi (*edge detection*) dalam mengekstraksi ciri dari objek kupu-kupu. Klasifikasi terhadap objek kupu-kupu menggunakan *K-Nearest Neighbor* (KNN). *Dataset* yang digunakan sebanyak 600 gambar kupu-kupu dengan masing-masing kelas atau spesies kupu-kupu sebanyak 100 gambar. Tahapan diawali dengan *pre-processing* berupa *scaling*, *segmentation*, dan *grayscale*. Data dibagi menjadi 2 bagian yaitu data latih dan data uji sebesar 70% dan 30%. Hasil akurasi terbesar didapatkan pada nilai $k = 5$ yaitu sebesar 80% (Andrian, et al., 2019).

Penelitian terdahulu dalam identifikasi kupu-kupu pernah dilakukan juga menggunakan *K-Nearest Neighbor* (KNN), dengan *gray level co-occurrence matrix* (glcm) untuk ekstraksi fitur atau ciri objeknya. *Dataset* yang digunakan berjumlah 100 gambar untuk masing-masing kelas kupu-kupu. Kelas kupu-kupu pada penelitian ini sebanyak 6 kelas atau spesies. Hasil akurasi terbesar yaitu 91.1% dengan nilai $k = 5$ pada orientasi sudut 90° (Andrian, et al., 2020).

Penelitian ini akan menggunakan *dataset* kupu-kupu sebanyak 100 gambar untuk masing-masing 8 kelas kupu-kupu. *Dataset* kupu-kupu akan diberikan perlakuan *augmentasi data*. Penelitian menggunakan arsitektur *AlexNet* dan *EfficientNet* untuk melatih *dataset* dan juga mengujinya menggunakan data uji dari gambar kupu-kupu. Kedua arsitektur digunakan untuk melakukan perbandingan dan mengetahui tingkat akurasi yang paling baik diantara keduanya. *Augmentasi data* digunakan untuk memberi perlakuan tambahan kepada *dataset*, sehingga bisa melihat pengaruh *augmentasi data* terhadap tingkat akurasi.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu, bagaimana perbandingan tingkat akurasi identifikasi kupu-kupu dengan *augmentasi data* menggunakan arsitektur *AlexNet* dan *EfficientNet*.

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini diantaranya.

1. Menggunakan arsitektur *EfficientNet* dan *AlexNet* untuk mengidentifikasi kupu-kupu.
2. Menggunakan *dataset* kupu-kupu berjumlah 800 gambar dengan 8 kelas, yaitu *Ariadne ariadne*, *Cethosia penthesilea*, *Papilio peranthus*, *Pacliopta aristolochiae*, *Papilio memnon*, *Papilio nephelus*, *Parantica aspiasa*, dan *Troides Helena*.
3. Menggunakan *augmentasi data* pada *dataset* kupu-kupu.
4. Menggunakan *hyperparameter* dan perlakuan yang sama untuk kedua arsitektur.

1.4 Tujuan

Tujuan penelitian ini untuk membandingkan kinerja arsitektur *EfficientNet* dan *AlexNet* dalam identifikasi kupu-kupu dengan *augmentasi data* berdasarkan tingkat akurasi uji kedua arsitektur.

1.5 Manfaat

Manfaat dari penelitian ini agar dapat digunakan sebagai acuan dalam mengembangkan penelitian selanjutnya menggunakan *deep learning* dan parameter lainnya untuk identifikasi kupu-kupu.

II. TINJAUAN PUSTAKA

2.1 Kupu-Kupu



Gambar 1. Kupu-kupu (*Rhopalocera*).

Kupu-kupu merupakan salah satu jenis serangga yang termasuk ke dalam *ordo* *Lepidoptera* dan memiliki peranan penting dalam menjaga keseimbangan ekosistem, salah satunya sebagai satwa penyerbuk pada proses pembuahan bunga (Oktaviati, et al., 2019). Kupu-kupu berada dalam satu *ordo* dengan ngengat, tetapi keduanya berada dalam *subordo* yang berbeda. Kupu-kupu berada dalam *subordo* *Rhopalocera*, sedangkan ngengat berada dalam *subordo* *Heterocera*. Kupu-kupu merupakan salah satu jenis serangga di dalam daftar kekayaan hayati Indonesia, keberagaman bentuk, corak, ataupun jenis kupu-kupu menjadi salah satu keunikan dan keindahan dari fauna ini. Keberadaan jenis kupu-kupu di sebuah habitat dapat dipengaruhi oleh beberapa faktor, baik faktor biotik maupun

faktor abiotik seperti ketinggian tempat, suhu, kelembapan udara, intensitas cahaya, dan cuaca (Syahfitri, 2019).

Habitat kupu-kupu dapat mempengaruhi keanekaragamannya. Kondisi habitat yang sering berubah-ubah baik cuaca, pencahayaan, kelembapan, dan sebagainya, dapat mempengaruhi keragaman kupu-kupu tersebut. Serangga ini termasuk sebagai *poikiloterm*, yaitu hewan yang menyesuaikan suhu tubuhnya dengan suhu lingkungan. Beberapa jenis kupu-kupu memiliki corak dan warna yang tersesuaian dengan kondisi lingkungannya, mulai dari lingkungan yang kurang cahaya, lingkungan terbuka dengan banyak cahaya, lingkungan yang memiliki banyak pohon, dan sebagainya. Tanaman di sekitar kupu-kupu memiliki keterkaitan erat baik sebagai tempat tinggal, pakan, maupun sebagai media penyerbukan oleh kupu-kupu. Penyerbukan yang dilakukan kupu-kupu terhadap tumbuhan membuat kupu-kupu menjadi indikator perubahan ekologi di lingkungannya, terutama karena kupu-kupu sangat rentan terhadap gangguan yang terjadi di sekitarnya (Arzar, et al., 2019).

2.2 Deep Learning

Deep Learning ialah pengembangan lebih lanjut dari *machine learning* (pembelajaran mesin). *Deep learning* merupakan bagian dari *computer vision*, karena dapat belajar untuk bekerja layaknya manusia ketika menangkap gambar, suara, gerakan, serta menginterpretasikannya. Data yang dipelajari *deep learning* lebih mendalam dan membuat *deep learning* dapat belajar dari kesalahan dengan sendirinya, sehingga teknologi ini lebih kompleks dibandingkan *machine learning*. *Deep learning* bekerja dengan *artificial neural network* (jaringan saraf buatan) yang memang bertujuan untuk meniru bagaimana manusia berpikir dan belajar (Alzubaidi, et al., 2021). Susunan *neural network* di dalam *deep learning* membuatnya dapat menghubungkan antara satu ciri dengan ciri yang lain atau satu data dengan data yang lain sehingga memudahkan proses klasifikasi. *Neural network* (jaringan saraf) awalnya terbatas dalam segi komputasi dan kompleksitasnya, oleh karena itu *deep learning* masih sangat terbatas dalam

kinerjanya. *Big data* telah mengatasi hal tersebut seiring perkembangannya, sehingga *deep learning* dapat memiliki jaringan saraf yang lebih besar dan kompleks untuk membantu komputer dalam pengamatan, belajar, dan bereaksi terhadap hal yang kompleks lebih cepat dibandingkan manusia. Kinerja *deep learning* juga bisa meningkat dengan adanya perkembangan dari *big data*.

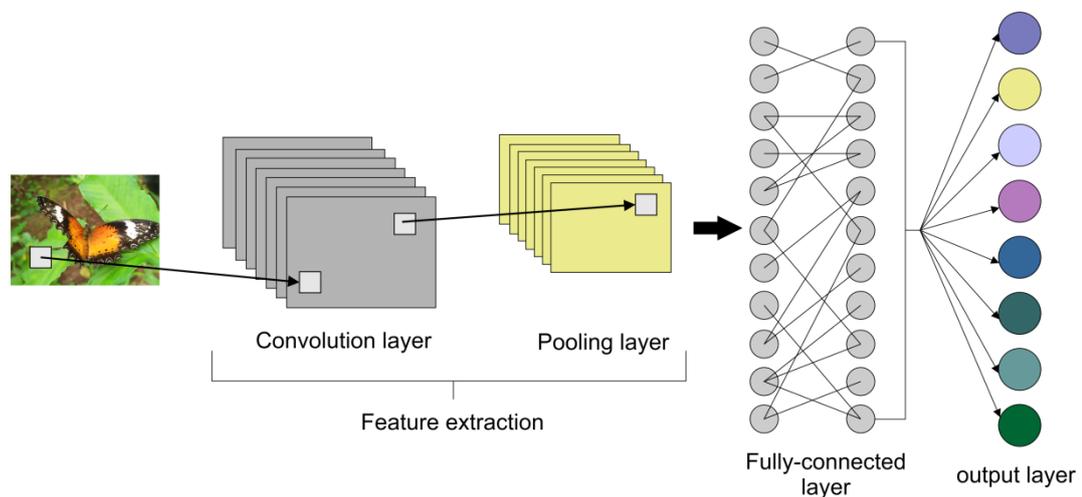
Data yang diterima oleh *deep learning* juga mempengaruhi kinerja mesin ini, semakin banyak data yang diterima maka akurasi atau ketepatan *deep learning* mengidentifikasi suatu objek lebih tinggi, begitupun sebaliknya. *Deep learning* akan mengekstraksi atau mengambil dan menyimpan ciri sebuah objek secara otomatis dengan bantuan algoritma dan komputasi yang telah tersusun. Model *deep learning* dilatih menggunakan data yang besar dan melalui banyak lapisan untuk membaca, menyimpan, memproses, dan mengklasifikasi sebuah masukan, sehingga seringkali kinerja *deep learning* melampaui kinerja manusia. *Deep learning* belajar melalui data dan contoh yang diberikan sebelumnya ke dalam komputer, *neural network* yang terdapat dalam *deep learning* akan mengekstraksi ciri dari data secara otomatis tanpa harus mendefinisikan ciri secara manual layaknya *machine learning* (Alzubaidi, et al., 2021). Data tersebut kemudian akan diproses untuk dapat diklasifikasi oleh *deep learning* melalui komputasi dan algoritma yang tersusun dalam *deep learning*.

Deep learning terus berkembang hingga memiliki beberapa macam algoritma yang dapat digunakan, diantaranya *Multilayer Perceptron* (MLP), *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), *Faster R-CNN*, *YOLO*, dan lain-lain. *Deep learning* dapat digunakan untuk mengenali tulisan, gambar, suara, maupun objek dalam sebuah video dengan adanya berbagai algoritma tersebut.

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu algoritma yang digunakan pada *deep learning*. CNN dikembangkan sekitar tahun 1988 oleh Yann

LeCun dan pertama kali digunakan untuk mengenali karakter seperti *zip codes* (kode pos) dan digit. Algoritma ini merupakan pengembangan lebih lanjut dari *Multilayer Perceptron* (MLP) yang memiliki banyak lapisan di dalamnya, untuk bisa memroses nilai-nilai *pixel* pada gambar. MLP awalnya hanya bisa digunakan untuk mengenali karakter dengan kinerja yang masih minim. CNN hadir dengan perkembangan lebih lanjut dari MLP, tidak hanya mengenali karakter, CNN juga dapat mengenali gambar dengan kinerja yang lebih baik dibandingkan MLP. Kinerja CNN dapat terlihat dengan adanya penambahan lapisan-lapisan tersembunyi untuk membantu identifikasi sebuah gambar, serta susunan dari *neural network* (jaringan saraf) yang saling terhubung untuk menyatukan ciri sebuah gambar (Alzubaidi, et al., 2021). Lapisan-lapisan awal pada CNN biasanya mendeteksi ciri dasar dari sebuah objek, seperti garis-garis *vertical*, *horizontal*, kemudian *output* dari lapisan pertama akan dijadikan masukan di lapisan selanjutnya untuk mengekstraksi ciri yang lebih kompleks. Lapisan yang semakin dalam akan mendeteksi ciri objek yang semakin sulit juga, seperti bentuk dan warna yang semakin detail. Lapisan-lapisan dan *neural network* dapat dibidang sangat membantu dalam keseluruhan proses CNN. Lapisan di dalam CNN dapat dilihat pada Gambar 2.



Gambar 2. Lapisan *Convolutional Neural Network* (CNN).

Neural network yang tersusun di dalam CNN membantu CNN untuk dapat bekerja layaknya manusia mengamati dan menganalisa sebuah objek. Interpretasi sebuah objek masih didapati kekeliruan selama dilakukan langsung oleh manusia, sehingga dengan adanya CNN dapat membantu proses interpretasi dan pengenalan terhadap objek lebih akurat dan cepat (Alzubaidi, et al., 2021). CNN bekerja secara otomatis dalam pengambilan ciri sebuah objek, begitu juga dalam pembentukan angka-angka *pixel* untuk menyimpan ciri dari gambar. Pekerjaan otomatis tersebut bisa terwujud dengan susunan komputasi dan algoritma CNN. Algoritma ini tersusun dari dua komponen utama, yaitu *feature extraction layer* dan *fully-connected layer*.

2.3.1 *Feature extraction layer*

Feature extraction layer (lapisan ekstraksi ciri) ialah layer atau lapisan di dalam CNN yang bertugas melakukan ekstraksi ciri dan menyimpannya menjadi nilai-nilai *pixel* dalam sebuah array. Gambar masukan tersusun atas nilai *pixel* berbentuk *array multidimensional*, karena ukuran array tersebut terlalu besar untuk diproses, sehingga diperlukan filter untuk mengambil beberapa ciri utama dari gambar tersebut. *Feature extraction* menyediakan *layer* atau lapisan-lapisan termasuk di dalamnya beberapa filter yang akan melalui proses perhitungan dengan perkalian *matrix* antara gambar masukan dan filternya. Proses perkalian akan diproses lagi untuk lebih menyederhanakan nilai ciri yang akan diproses dalam klasifikasi. Inti dari *feature extraction* selain mengambil ciri dari gambar, tetapi juga menyederhanakan bentuk dan nilai *pixel* yang dimiliki oleh gambar masukan tanpa menghilangkan ciri yang penting dari objek (Almryad & Kutucu, 2020). Lapisan yang dilewati dalam *feature extraction* diantaranya *convolution layer* dan *pooling layer*.

2.3.1.1 *Convolution Layer*

Convolution layer atau lapisan konvolusi merupakan proses ketika menyederhanakan nilai pada gambar masukan, sekaligus menyimpan ciri gambar melalui nilai *pixel* yang dihasilkan dari

penyederhanaan. Penyederhanaan dilakukan melalui perkalian antara *matrix* masukan dan filter. Gambar masukan akan memiliki nilai *pixel* berbentuk *array multidimensional* yang harus disederhanakan, baru kemudian dilakukan proses penyederhanaan nilai tersebut tanpa menghilangkan informasi utama ciri objeknya menggunakan sebuah filter. Hasil dari lapisan konvolusi ini bisa juga disebut sebagai *feature map* atau peta ciri.

2.3.1.2 *Pooling Layer*

Pooling layer atau lapisan *pooling* berfungsi untuk mengurangi ukuran dari kumpulan ciri hasil konvolusi atau mengurangi ukuran dari gambar masukan (Borza, et al., 2018). Pengurangan ukuran tersebut ditujukan agar proses komputasi lebih ringan untuk dijalankan saat memproses data (Alzubaidi, et al., 2021). Proses *pooling* biasanya mengambil nilai terbesar (*max pooling*) atau nilai rata-rata (*average pooling*) *pixel* dari *feature map* yang dihasilkan *convolution layer* (Irfansyah, et al., 2021). Nilai *pixel* yang diambil akan merepresentasikan ciri utama dari sebuah objek, sehingga tidak mengurangi informasi ciri objek meskipun ukuran kumpulan ciri gambar terus berkurang. *Pooling layer* juga ditujukan agar ciri yang akan diproses untuk klasifikasi, merupakan ciri utama dan tidak menyisakan ciri lain yang tidak berkaitan dengan objek yang dimaksudkan.

2.3.2 *Fully-Connected Layer*

Fully-connected layer ialah lapisan dalam CNN yang digunakan untuk tahapan klasifikasi objek gambar. *Fully-connected layer* juga merupakan tahapan terakhir dalam CNN hingga bisa didapatkan hasil klasifikasi terhadap kelas objek. Masukan yang bisa diterima *fully-connected layer* haruslah berbentuk vektor atau satu dimensi, sehingga perlu dilakukan *flatten* atau perataan bentuk *feature map* sebelum masuk ke tahap klasifikasi (Almryad & Kutucu, 2020). *Fully-connected layer* terdiri atas

weights (bobot) dan *bias*, serta terdapat *neuron* yang nantinya akan saling terhubung dengan menggunakan nilai bobot dan *bias*. *Neuron* yang terhubung sama seperti susunan *puzzle*, setiap ciri akan saling terhubung hingga bisa didapatkan hasil klasifikasi kelas objek.

2.3.3 *Rectified Linear Unit (ReLU) Activation*

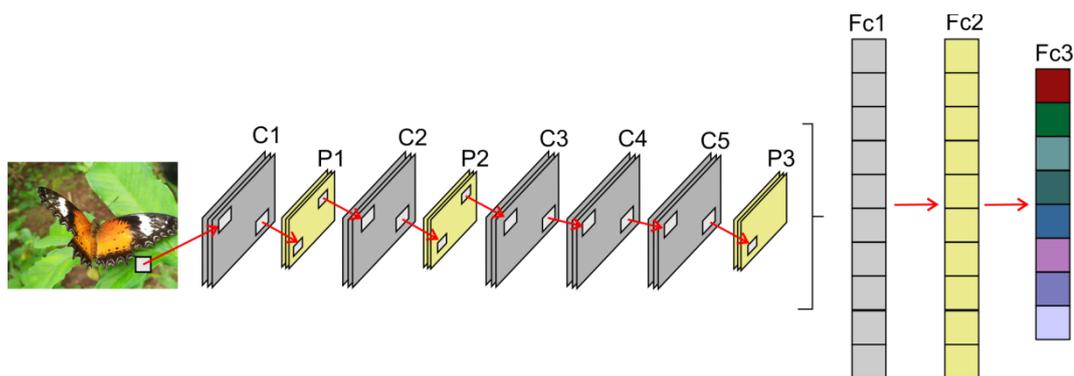
ReLU *activation* atau aktivasi ReLU digunakan untuk memperkenalkan dan mengajarkan fungsi *non-linear* ke dalam jaringan konvolusi. ReLU biasanya digunakan dalam *convolution layer* ataupun *pooling layer*. Fungsi *non-linear* ialah kelebihan dari CNN, sehingga bisa memisahkan ciri maupun kelas yang membutuhkan lebih dari sekedar garis lurus (*linear*). ReLU akan mengembalikan nilai negatif yang dimiliki konvolusi menjadi 0 dan mengembalikan nilai positif menjadi nilai itu sendiri. Proses tersebut yang menjadi ciri utama dari aktivasi ReLU, serta membuat performa ReLU lebih baik diantara aktivasi lainnya. Fungsi yang digunakan dalam ReLU yaitu $f(x) = \max(0, x)$ (Ilahiyah & Nilogiri, 2018).

2.3.4 *Softmax Activation*

Softmax activation atau aktivasi softmax ialah fungsi matematis yang digunakan untuk mengubah angka vektor menjadi sebuah probabilitas vektor. *Softmax* biasanya digunakan saat proses klasifikasi dan penentuan kelas objek, setiap *node neuron* atau ciri objek yang terhubung dan telah dijumlahkan bobot-bobotnya, akan memiliki nilai probabilitas yang menentukan kelas objek tersebut. *Softmax* akan mengubah *output* berupa jumlah bobot menjadi sebuah nilai probabilitas. *Output* dari *softmax* diartikan sebagai probabilitas atau kemungkinan kelas dari sebuah objek. *Softmax* sesuai digunakan untuk *multiclass classification* atau klasifikasi banyak kelas yang biasanya lebih dari dua kelas.

2.4 AlexNet

AlexNet pernah dinobatkan sebagai pemenang dalam kategori *image classification* pada acara *ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012 contest)*. *AlexNet* merupakan salah satu arsitektur CNN dengan metode, konsep, dan teknik yang mudah dipahami dan dipelajari oleh pemula dalam *deep learning*. Arsitektur ini menggunakan *convolution layer*, *pooling*, *dropout*, *activation ReLU*, *fully-connected layer*, dan beberapa parameter lain. *Memory* atau kapasitas penyimpanan dan pemrosesan yang dibutuhkan *AlexNet* pada awalnya cukup besar, hingga dibutuhkan dua GPU untuk menyimpan dan memproses sebagian modelnya. Kapasitas GPU yang juga semakin melimpah, membuat arsitektur ini dapat bekerja lebih optimal dengan hanya satu GPU. Arsitektur ini bisa dibilang cukup mirip dengan *LeNet*, tetapi memiliki *convolution layer* yang lebih banyak dan parameter latih model yang lebih besar juga. Lapisan di dalam *AlexNet* dapat dilihat pada Gambar 3.

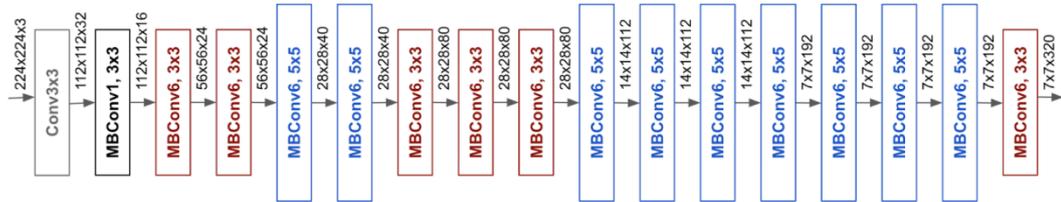


Gambar 3. Lapisan di dalam *AlexNet*.

2.5 EfficientNet

EfficientNet ialah salah satu arsitektur CNN yang dibentuk untuk memudahkan CNN dalam membesarkan skala pelatihannya tanpa harus meningkatkan skalanya secara manual. CNN pada awalnya membutuhkan peningkatan model skala secara manual dan dengan *manual tuning*, untuk dapat membaca resolusi masukan gambar yang besar dan memperluas atau memperdalam skala CNN. *EfficientNet* merupakan hasil pembelajaran lebih lanjut untuk meningkatkan skala dan efisiensi

dari CNN (Tan & Le, 2020). Arsitektur ini lebih efisien terutama dari segi ukuran yang kecil dan kecepatan pemrosesannya. *EfficientNet* kurang lebih mengadaptasi dari *MobileNetV2* dan *MnasNet* dengan menggunakan MBConv (*Mobile Inverted Bottleneck Convolution*) sebagai lapisannya. Model *EfficientNet* mendapatkan akurasi yang besar dan efisiensi yang lebih baik dibandingkan CNN, serta mengurangi ukuran parameter latih (Tan & Le, 2020). Lapisan *EfficientNet* dapat dilihat pada Gambar 4.



Gambar 4. Lapisan di dalam *EfficientNet*.

2.6 Augmentasi Data

Augmentasi data digunakan untuk menambah variasi *dataset* yang akan digunakan untuk pelatihan data nantinya. Penambahan variasi tersebut ditujukan agar data memiliki banyak variasi dan lebih meningkatkan kemungkinan dan kemampuan klasifikasi objek (Shorten & Khoshgoftaar, 2019). *Augmentasi data* bisa menjadi salah satu solusi untuk menghindari terjadinya *overfitting* (Borza, et al., 2018). *Overfitting* merupakan kondisi ketika model bekerja sangat baik dalam mengingat dan mempelajari objek pada proses *training data*, sedangkan memberikan hasil yang sangat buruk pada proses *testing data*. Kondisi tersebut dapat diminimalisir salah satunya dengan menambah data melalui *augmentasi*, sehingga model dapat belajar bervariasi data untuk bisa diuji dengan baik. *Augmentasi data* dalam prosesnya hanya akan mengembalikan data hasil *augmentasi* ke dalam mesin, sedangkan data asli tidak akan ikut dikembalikan. Model yang diberikan data baru dengan sedikit modifikasi untuk diuji, dapat secara otomatis mempelajarinya dengan lebih baik ketika variasi data ditambahkan dalam proses *training* atau belajar. *Augmentasi data* yang cukup

sering digunakan diantaranya, *flipping*, *rotation*, *color space transformations*, *translation*, dan sebagainya.

2.6.1 *Flipping*

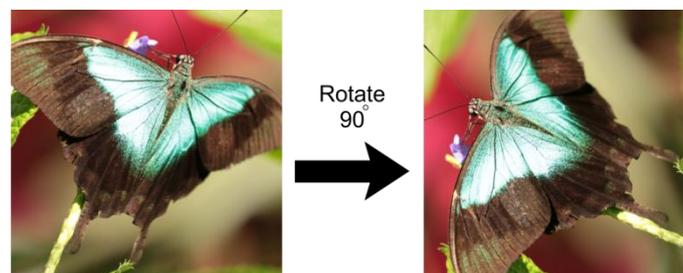
Flipping merupakan salah satu teknik *augmentasi data* dengan cara membalik gambar ke arah *vertical* maupun *horizontal*. *Augmentasi* ini termasuk mudah diterapkan dan terbukti mampu meningkatkan kinerja model klasifikasi saat menguji data (Shorten & Khoshgoftaar, 2019). Hasil *flipping* dapat dilihat pada Gambar 5.



Gambar 5. Hasil *flip* gambar.

2.6.2 *Rotation*

Rotation (rotasi) atau perputaran ialah teknik *augmentasi data* yang digunakan untuk memutar gambar. Gambar bisa diputar dengan nilai 1° - 359° baik ke arah kanan maupun kiri. *Augmentasi* rotasi harus disandingkan dengan nilai derajatnya agar hasil *augmentasinya* lebih baik dan aman (Shorten & Khoshgoftaar, 2019). Hasil *rotation* dapat dilihat pada Gambar 6.



Gambar 6. Hasil *rotation*.

2.6.3 *Color space transformation*

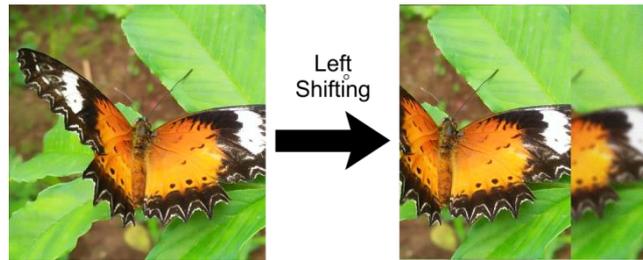
Color space transformation atau perubahan komposisi warna ditujukan untuk menambah variasi data dengan mengubah komposisi warna pada gambar. Gambar masukan biasanya memiliki 3 channel warna utama yaitu RGB (*Red, Green, Blue*), dari ketiga warna utama tersebut dapat diatur sehingga gambar akan dominan ke salah satu warna utama. Perubahan komposisi warna dapat menambah variasi data apabila data uji nantinya memiliki perbedaan komposisi warna maupun pencahayaan dari data latih, sehingga dapat juga meningkatkan akurasi uji model. Hasil perubahan warna gambar dapat dilihat pada Gambar 7.



Gambar 7. Hasil pengubahan warna.

2.6.4 *Translation*

Translation dilakukan dengan cara menggeser (*shift*) gambar ke arah atas, bawah, kanan, atau kiri. Menggeser gambar berguna untuk menghindari *bias* posisi yang terjadi pada data. *Bias* posisi terjadi ketika objek dominan berada di posisi tengah gambar, hal ini menyebabkan objek data uji nantinya harus tepat berada di tengah untuk bisa terproses dengan baik. Posisi objek tersebut divariasikan dengan *translation*, sehingga objek data uji tidak harus berada di tengah untuk dapat diproses dan mendapat akurasi yang baik. Hasil pergeseran gambar dapat dilihat pada Gambar 8.



Gambar 8. Hasil pergeseran gambar.

2.7 *Hyperparameter*

Hyperparameter merupakan jenis konfigurasi atau pengaturan terhadap model sebelum melakukan proses pelatihan data atau *training*. Nilai *hyperparameter* tidak bisa dimunculkan secara otomatis oleh data dalam model, tetapi didefinisikan secara manual oleh praktisi, itulah yang membedakannya dengan *parameter* (Afaq & Rao, 2020). *Parameter* akan muncul selama proses *training* dan tidak bisa ditentukan secara manual nilainya, seperti nilai bobot pada *neural network*. *Hyperparameter* yang telah diatur sebelum proses *training*, akan menentukan kemunculan nilai-nilai *parameter* yang digunakan selama proses *training* agar bisa melakukan prediksi data dengan baik. Nilai *parameter* hanya muncul melalui data-data yang diberikan termasuk di dalamnya *hyperparameter*, sehingga *hyperparameter* juga menjadi penentu performa dan keberhasilan model dalam memprediksi. *Hyperparameter* yang paling umum dan sering digunakan dalam model CNN yaitu, *learning-rate*, *batch-size*, *optimizer*, dan *epoch*.

2.7.1 *Learning-rate*

Learning-rate digunakan untuk menghitung nilai koreksi terhadap bobot pada saat proses *training* berjalan (Rochmawati, et al., 2021). *Hyperparameter* ini mengacu kepada tingkat kecepatan algoritma atau ukuran langkah yang perlu diambil oleh *gradient descent* untuk mencapai titik optimal model.

2.7.2 *Batch-size*

Batch-size merupakan pengambilan sejumlah sample dari data *training* yang akan dibentuk menjadi kelompok atau *batch*. Tiap *batch* diambil untuk digunakan dalam melakukan proses satu iterasi yaitu tahap pembacaan data secara *forward* (ke depan) dan *backward* (ke belakang) (Rochmawati, et al., 2021). *Batch-size* mempengaruhi kecepatan model dalam belajar dan kestabilan model tersebut.

2.7.3 *Optimizer Adam*

Optimizer Adam termasuk *optimizer* yang efektif dan dapat melatih data dalam waktu yang cukup singkat. *Optimizer* ini merupakan pengembangan dari algoritma *Stochastic Gradient Descent* (SGD) klasik yang telah mengalami pembaharuan bobot jaringannya (Rochmawati, et al., 2021).

2.7.4 *Epoch*

Epoch merupakan nilai yang menentukan jumlah proses *training* dilakukan. *Epoch* terhitung ketika model telah berhasil membaca keseluruhan data *training* (Afaq & Rao, 2020). Satu *epoch* dihitung ketika semua data *training* tiap *batch* selesai dipelajari oleh model.

2.8 *Confusion Matrix*

Confusion matrix biasanya digunakan untuk menunjukkan ringkasan performa dan hasil prediksi dari klasifikasi objek. Perhitungan *confusion matrix* dapat memberikan gambaran seberapa akurat hasil klasifikasi objek dan *error* yang dihasilkan oleh model. Jumlah prediksi benar dan salah akan dihitung dalam *confusion matrix* untuk kemudian menghasilkan prediksi keseluruhan maupun tiap kelasnya. *Confusion matrix* terdiri dari *accuracy*, *precision*, *recall*, dan *f1-score*. *Confusion matrix* memiliki ukuran penilaian melalui empat metrik utama, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Maxwell, et al., 2021). Gambaran umum metrik tersebut dapat dilihat pada Tabel 1.

Tabel 1. Metrik utama

		Hasil Prediksi	
		<i>Positive</i>	<i>Negative</i>
Data Sebenarnya	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

Keterangan :

- *True Positive* : model dengan benar memprediksi sebuah kelas (positive) sebagai kelas yang sebenarnya (positive).
- *True Negative* : model dengan benar memprediksi sebuah kelas (negative) sebagai kelas lain (negative).
- *False Positive* : model salah memprediksi kelas lain (negative) sebagai kelas yang sebenarnya (positive).
- *False Negative* : model salah memprediksi kelas yang sebenarnya (positive) sebagai kelas lain (negative).

2.8.1 Accuracy

Accuracy ialah penilaian yang dapat menggambarkan performa model dalam klasifikasi semua kelasnya. Penghitungan dilakukan dengan menjumlahkan metrik prediksi benar dan membaginya dengan total keseluruhan prediksi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (Maxwell, et al, 2021)$$

2.8.2 Precision

Precision digunakan untuk menghitung hasil prediksi, berupa prediksi yang *benar* terhadap sebuah kelas yang sebenarnya dan prediksi yang keliru atau salah sasaran terhadap sebuah kelas yang sebenarnya. *Precision* mengukur akurasi model dalam mengklasifikasi sebuah kelas sebagai kelas yang sebenarnya, maupun kelas lain yang terklasifikasi sebagai kelas yang sebenarnya. Model akan memiliki *precision* yang besar ketika hasil

prediksi benar (TP) semakin besar dan hasil prediksi salah (FP) semakin kecil.

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (Maxwell, et al, 2021)$$

2.8.3 *Recall*

Recall digunakan untuk mengukur kemampuan model mendeteksi kelas yang sebenarnya. Hasil deteksi yang dilihat berupa kelas sebenarnya yang terprediksi sebagai kelas lain maupun kelas yang benar terprediksi sebagai kelas sebenarnya. *Recall* hanya mementingkan tingkat keberhasilan kelas sebenarnya terprediksi, baik sebagai kelas tersebut maupun sebagai kelas lain.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (Maxwell, et al, 2021)$$

2.8.4 *F1-Score*

F1-Score digunakan untuk mengukur *mean* dari *precision* dan *recall*. Indikasi yang ditampilkan oleh *F1-Score* yaitu model klasifikasi memiliki nilai *precision* dan *recall* yang baik. *F1-Score* biasanya digunakan ketika dataset yang digunakan dalam pelatihan model tidak seimbang jumlahnya. Perhitungan ini dapat sebagai nilai pembanding antara algoritma yang digunakan dalam melatih data.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \dots\dots\dots (Maxwell, et al., 2021)$$

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilaksanakan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Laboratorium Rekayasa Perangkat Lunak (Lab RPL) FMIPA dan Ruang Laboratorium 204 UPT TIK Universitas Lampung yang berada di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No.1, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung. Penelitian dilakukan dari Desember tahun ajaran 2020/2021 sampai dengan Juni tahun ajaran 2020/2021.

3.2 Bahan dan Alat

3.2.1 Bahan penelitian

Penelitian ini menggunakan *dataset* berisi gambar kupu-kupu sebagai bahannya. Gambar kupu-kupu yang digunakan sebanyak 800 gambar (sebelum *augmentasi*) dengan kelas sebanyak 8 dan masing-masing kelas tersebut memiliki 100 gambar. *Dataset* didapatkan dari tempat penangkaran kupu-kupu Taman Kupu-Kupu Gita Persada. Kelas kupu-kupu yang digunakan diantaranya Kupu Jarak (*Ariadne ariadne*), Kupu Batik Cap (*Cethosia penthesilea*), Kupu Hijau Biru (*Papilio peranthus*), Kupu Pantat Merah (*Pacliopta aristolochiae*), Kupu Raja Limau (*Papilio memnon*), Kupu Jojo (*Papilio nephelus*), Kupu Harimau Hijau Kuning (*Parantica aspiasa*), dan Kupu Raja Helena (*Troides Helena*). Bahan penelitian yang digunakan untuk percobaan setelah ditambahkan *augmentasi flipping (vertical dan horizontal)* sebanyak 2400 gambar kupu-kupu. Nama kelas dan jumlah dataset yang digunakan sebelum diterapkan *augmentasi* dapat dilihat pada Tabel 2.

Tabel 2. Nama kelas dan jumlah *dataset* awal

Nama Kelas	Jumlah <i>Dataset</i>
Kupu Jarak	100
Kupu Batik Cap	100
Kupu Hijau Biru	100
Kupu Pantat Merah	100
Kupu Raja Limau	100
Kupu Jojo	100
Kupu Harimau Hijau Kuning	100
Kupu Raja Helena	100
Total	800

3.2.2 Alat

Alat yang digunakan dalam penelitian ini ialah sebagai berikut :

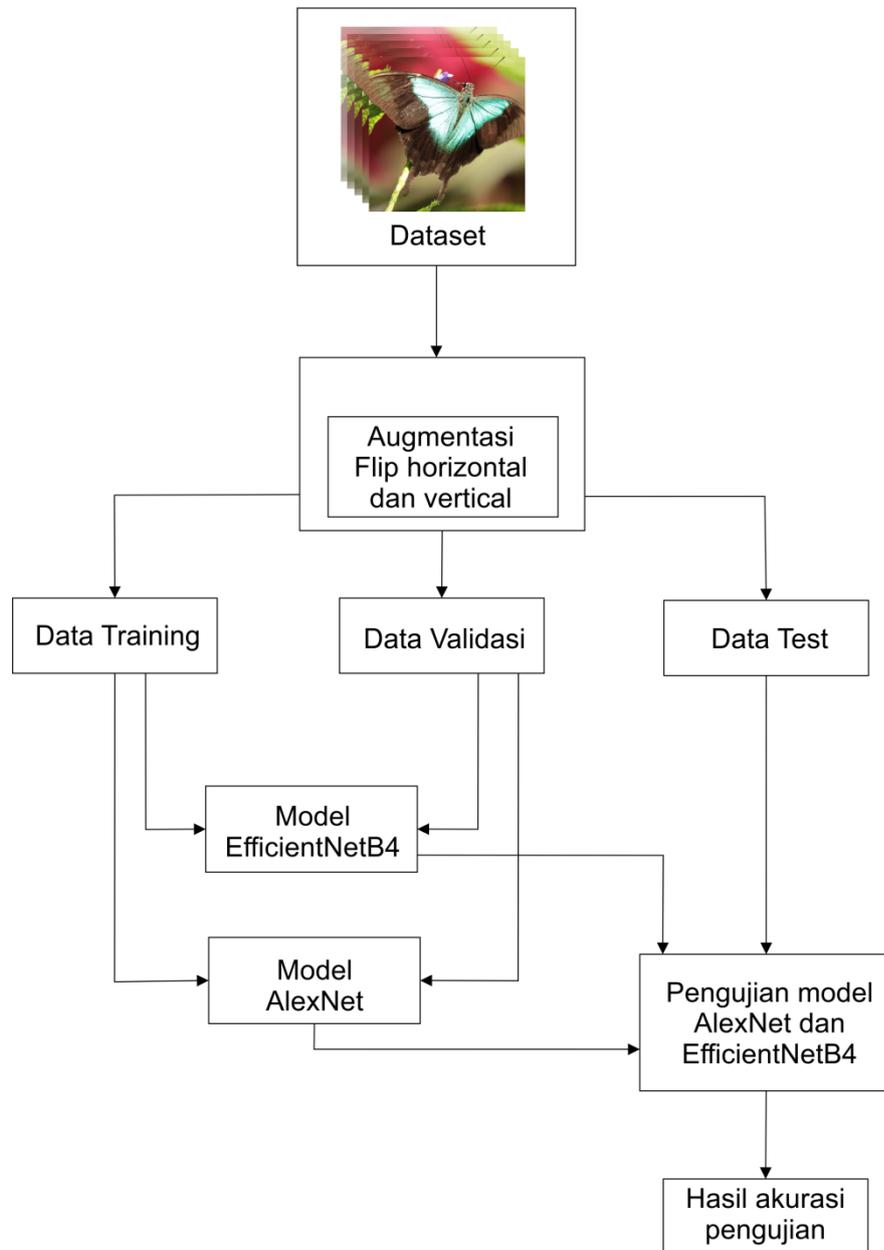
3.2.2.1 Perangkat Lunak

Sistem operasi *Windows* 10 64-bit, *Google Colab*, *Jupyter Notebook*, dan *Google Drive* digunakan sebagai pendukung perangkat lunak dalam proses identifikasi kupu-kupu

3.2.2.2 Perangkat Keras

Komputer GPU NVIDIA Tesla K20 dan Laptop Asus VivoBook X505Z dengan spesifikasi RAM 4.00 GB, *Harddisk* 1 TB, SSD 128 GB, dan *processor* AMD Ryzen 3 2200U dengan Radeon Vega Mobile Gfx 2.50 GHz sebagai alat penelitian untuk proses identifikasi kupu-kupu.

3.3 Metode



Gambar 9. Metode penelitian.

Tahapan yang dilalui dalam penelitian ini dapat dilihat pada Gambar 9 dan dijelaskan sebagai berikut.

3.3.1 Pengumpulan *dataset*

Tahapan pertama yang dilakukan ialah mengumpulkan *dataset* sebagai bahan utama penelitian. *Dataset* yang dikumpulkan sebanyak 800 gambar kupu-kupu dengan 8 kelas. Gambar disimpan dalam *google drive* untuk memudahkan pemanggilan dan penggunaan data ke *google colab*. Gambar kupu-kupu juga disimpan dalam komputer Tesla. Setiap gambar kupu-kupu disimpan ke dalam folder yang berbeda sesuai label masing-masing kelasnya, yang tertera pada nama folder. Mesin akan membaca *dataset* di awal prosesnya, baik gambar kupu-kupu maupun nama kelas yang tertera pada tiap foldernya.

3.3.2 *Augmentasi data*

Penelitian ini menggunakan *augmentasi data* untuk menghindari *overfitting* dan menambah bahan belajar untuk model nantinya. *Augmentasi* juga dilakukan untuk perbandingan akurasi antara model sebelum *augmentasi* dan sesudah ditambah *augmentasi*. Terdapat beberapa macam *augmentasi data*, namun yang digunakan adalah *flipping*. *Flipping* termasuk *augmentasi data* yang mudah diimplementasikan serta terbukti berhasil untuk meningkatkan performa dan akurasi pengujian. *Dataset* ditambah melalui *flipping horizontal* dan *vertical*, masing-masing *flipping* menghasilkan 800 gambar tambahan, sehingga total gambar *augmentasi* sebanyak 1600 gambar. Proses *augmentasi* sekaligus dilakukan *resize* terhadap ukuran data menjadi 224 x 224. Gambar hasil *augmentasi* kemudian ditambahkan dengan *dataset* awal, sehingga total keseluruhan bahan untuk melatih dan menguji data sebanyak 2400 gambar. Nama kelas dan jumlah dataset setelah diterapkan *augmentasi flip horizontal* dan *flip vertical* dapat dilihat pada Tabel 3 dan Tabel 4 .

Tabel 3. Nama kelas dan jumlah *dataset flip horizontal*

Nama Kelas	<i>Dataset flip horizontal</i>
Kupu Jarak	200

Tabel 3 (lanjutan)

Nama Kelas	<i>Dataset flip horizontal</i>
Kupu Raja Limau	200
Kupu Jojo	200
Kupu Harimau Hijau Kuning	200
Kupu Raja Helena	200
Kupu Batik Cap	200
Kupu Hijau Biru	200
Kupu Pantat Merah	200
Total	1600

Tabel 4. Nama kelas dan jumlah *dataset flip vertical*

Nama Kelas	<i>Dataset flip vertical</i>
Kupu Jarak	200
Kupu Batik Cap	200
Kupu Hijau Biru	200
Kupu Pantat Merah	200
Kupu Raja Limau	200
Kupu Jojo	200
Kupu Harimau Hijau Kuning	200
Kupu Raja Helena	200
Total	1600

3.3.3 Pembagian data

Pembagian data yang dilakukan membagi data menjadi 3 bagian utama, yaitu *data training*, *data validasi*, dan *data test*.

3.3.3.1 Data Training

Data training dibutuhkan sebagai bahan utama untuk melatih data. Bagian *data training* yaitu sebesar 70% dari keseluruhan gambar, baik sebelum *augmentasi* maupun setelah *augmentasi*.

3.3.3.2 Data Validasi

Data validasi diperlukan untuk membantu tahapan pelatihan data menggunakan *data training*. Proses pelatihan menggunakan *data training* juga memerlukan validasi untuk membuktikan keaslian dan kesamaan data yang dibaca oleh model. *Data validasi* akan digunakan setiap model menggunakan *data training*, dan begitu seterusnya proses akan berulang sampai didapatkan hasil akurasi latih maupun uji. Bagian *data validasi* yang digunakan sebanyak 10% dari keseluruhan gambar.

3.3.3.3 Data Test

Data test digunakan untuk menguji performa dan keberhasilan model. Data yang digunakan haruslah belum pernah digunakan dan dilihat sebelumnya oleh *data validasi* maupun *data training*. Bagian *data testing* biasanya sama dengan *data validasi* ataupun lebih kecil. Penelitian ini menggunakan *data test* sebesar 20% dari keseluruhan gambar.

3.3.4 Proses latih data

Tahapan melatih data menggunakan dua model utama yaitu *AlexNet* dan *EfficientNet*. Dua model tersebut digunakan untuk melakukan perbandingan akurasi yang dihasilkan keduanya. Pelatihan data atau proses model belajar dari data juga menggunakan beberapa *hyperparameter* seperti *epoch*, *batch-size*, *optimizer*, dan *learning-rate* sebagai penunjang keberhasilan model melakukan pelatihan maupun pengujian nantinya. Kedua model akan menggunakan *hyper-parameter* yang sama sehingga

hasil perbandingannya bisa terlihat. Nilai *hyperparameter* yang digunakan dapat dilihat pada Tabel 5.

Tabel 5. *Hyperparameter* model

<i>Hyperparameter</i>	Jenis/nilai
<i>Epoch</i>	10
<i>Batch-size</i>	8
<i>Optimizer</i>	Adam
<i>Learning-rate</i>	0.0001

3.3.5 Pengujian model

Pengujian model dilakukan setelah proses pelatihan model terhadap *data training* dan validasi selesai dilakukan. *Data test* akan digunakan pada tahapan ini, untuk menunjukkan keberhasilan model yang telah dilatih. Hasil akurasi pengujian juga dapat ditampilkan setelah proses *training* selesai, sehingga dapat dilakukan perbandingan terhadap kedua model yang digunakan.

3.3.6 Klasifikasi objek

Hasil klasifikasi objek dapat dilihat sebagai hasil terakhir dari tahapan-tahapan model. Klasifikasi akan menunjukkan kecenderungan objek terhadap sebuah kelas. Kecenderungan tersebut ditampilkan melalui nilai probabilitas yang ditampilkan masing-masing kelas objek. Tingkat keberhasilan dan performa model dalam klasifikasi terhadap sebuah objek, juga dapat dilihat dan dianalisis melalui nilai-nilai dalam confusion matrix.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Simpulan yang didapatkan dari hasil penelitian adalah sebagai berikut.

1. Identifikasi kupu-kupu menggunakan arsitektur *AlexNet* dan *EfficientNetB4* berhasil dilakukan dengan tambahan *augmentasi data flip horizontal* dan *flip vertical*.
2. Hasil akurasi terbesar dengan mesin colab didapatkan oleh *EfficientNetB4* menggunakan *augmentasi flip vertical* dengan akurasi sebesar 99.69%. Hasil akurasi terbesar dengan mesin tesla didapatkan oleh *EfficientNetB4* dan *AlexNet* menggunakan *augmentasi flip horizontal* dengan akurasi sebesar 99.37%.
3. Percobaan berhasil dilakukan pada dua mesin, yaitu menggunakan *Google Colab (online)* dan komputer Tesla K20 (*offline*).
4. Nilai ketepatan model dalam memprediksi kelas kupu-kupu (*precision*) dan nilai efektivitas pencarian data uji dalam menemukan kelas kupu-kupu (*recall*), dihasilkan terbanyak oleh arsitektur *EfficientNetB4* menggunakan *augmentasi flip vertical*, hampir semua kelas kupu-kupunya menghasilkan *precision* dan *recall* sebesar 100%.
5. Terdeteksi kesalahan prediksi pada 3 pasang kelas kupu-kupu yang memiliki kemiripan, corak, warna, maupun bentuk fisik pada gambar kupu-kupu, yaitu Kupu Batik Cap, Kupu Jarak, Kupu Harimau Kuning Hijau, Kupu Hijau Biru, Kupu Raja Limau, dan Kupu Pantat Merah.
6. Jumlah *parameter* belajar *AlexNet* lebih banyak dibandingkan *EfficientNetB4*, tetapi tingkat akurasi terbesar dapat dihasilkan oleh *EfficientNetB4* menggunakan *augmentasi flip vertical*, hal ini membuktikan efisiensi kinerja dari *EfficientNetB4* dengan *parameter* yang lebih sedikit.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut.

1. Menambahkan jumlah kelas kupu-kupu lainnya.
2. Menambahkan jumlah variasi gambar masing-masing kelas kupu-kupu.
3. Menggunakan ragam arsitektur CNN lainnya seperti VGG-16, ResNet-50, Inception, YOLOV5, dan lain-lain.
4. Melakukan percobaan dengan *hyperparameter* yang lebih beragam dari penelitian ini.
5. Mengembangkan penelitian ini menjadi bentuk aplikasi seperti *website* maupun aplikasi *mobile*.

DAFTAR PUSTAKA

- Afaq, S., & Rao, S. (2020). Significance Of Epochs On Training A Neural Network. *International Journal Of Scientific & Technology Research*, 485-488.
- Ali, K., Shaikh, Z. A., Khan, A. A., & Laghari, A. A. (2021). Multiclass Skin Cancer Classification Using EfficientNets - a first step towards preventing skin cancer. *Neuroscience Informatics*, 1-10.
- Almryad, A. S., & Kutucu, H. (2020). Automatic Identification for Field Butterflies by Convolutional Neural Networks. *Engineering Science and Technology, an International Journal*, 189-195.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of Big Data*, 1-74.
- Andrian, R., Anwar, S., Muhammad, M. A., & Junaidi, A. (2019). Identifikasi Kupu-Kupu Menggunakan Ekstraksi Fitur Deteksi Tepi (Edge Detection) dan Klasifikasi K-Nearest Neighbor (KNN). *Jurnal Teknik Informatika dan Sistem Informasi*, 234-243.
- Andrian, R., Maharani, D., Muhammad, M. A., & Junaidi, A. (2020). Butterfly Identification Using Gray Level Co-Occurrence Matrix (GLCM) Extraction Feature and K-Nearest Neighbor (KNN) Classification. *Jurnal Ilmiah Teknologi Sistem Informasi*, 11-21.
- Bakri, B. A., Ahmad, Z., & Hatim, S. M. (2019). Butterfly Family Detection and Identification Using Convolutional Neural Network for LepidopteroLOGY. *International Journal of Recent Technology and Engineering*, 635-640.

- Borugadda, P., Lakshmi, R., & Govindu, S. (2021). Classification of Cotton Leaf Diseases Using AlexNet and Machine Learning Models. *Current Journal of Applied Science and Technology*, 29-37.
- Borza, D., Itu, R., & Danescu, R. (2018). Micro Expression Detection and Recognition From High Speed Cameras Using Convolutional Neural Networks.
- Butterfly Park, G. (n.d.). Retrieved Maret 23, 2022, from Gita Persada Butterfly Park: <http://gitapersada.weebly.com/home.html>
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *Jurnal Sistem & Teknologi Informasi Indonesia*, 49-56.
- Irfansyah, D., Mustikasari, M., & Suroso, A. (2021). Arsitektur Convolutional Neural Network (CNN) AlexNet untuk Klasifikasi Hama pada Citra Daun Tanaman Kopi. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 87-92.
- Kartini, P. D., Winarno, G. D., Iswandaru, D., & Setiawan, A. (2021). Persepsi Pengunjung Terhadap Kegiatan Wisata di Taman Kupu-Kupu Gita Persada. *Jurnal Hutan Tropis*, 56-64.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, (pp. 1-9).
- Maxwell, A. E., Warner, T. A., & Guillen, L. A. (2021). Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies-Part 1: Literature Review. *Remote Sensing*, 1-27.
- Oktaviati, W., Rifanjani, S., & Ardian, H. (2019). Keanekaragaman Jenis Kupu-Kupu (Ordo Lepidoptera) Pada Ruang Terbuka Hijau Kota Pontianak. *Jurnal Hutan Lestari*, 7(1), 79-85.
- Rochmawati, N., Hidayati, H. B., & Yamasari, Y. (2021). Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal Information Engineering and Educational Technology*, 44-48.

- Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 1-48.
- Syahfitri, D. R. 2019. *Keanekaragaman Jenis Kupu-Kupu (Lepidoptera : Rhopalocera) Pada Kawasan Hutan Resort Cinta Raja, Taman Nasional Gunung Leuser*. (Skripsi). Universitas Sumatera Utara.
- Tan, M., & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.
- Theivaprakasham, H. (2020). Identification of Indian Butterflies Using Deep Convolutional Neural Network. *Journal of Asia-Pacific Entomology*.