

SKRIPSI

PENERAPAN *CONVOLUTIONAL NEURAL NETWORKS* (CNN) DALAM KLASIFIKASI TINGKAT KEMATANGAN PUCUK DAUN TEH

Oleh:
Muhammad Faisal Raafi
065119132



PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024

SKRIPSI

PENERAPAN *CONVOLUTIONAL NEURAL NETWORKS* (CNN) DALAM KLASIFIKASI TINGKAT KEMATANGAN PUCUK DAUN TEH

Diajukan sebagai salah satu syarat untuk memperoleh
Gelar Sarjana Komputer Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

Oleh:
Muhammad Faisal Raafi
065119132



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2024**

HALAMAN PENGESAHAN

Judul : Penerapan *Convolutional Neural Networks* (CNN) Dalam Klasifikasi
Tingkat Kematangan Pucuk Daun Teh
Nama : Muhamad Faisal Raafi
NPM : 065119132

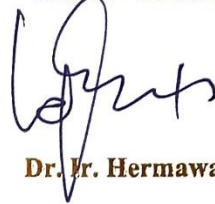
Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA - UNPAK



Mulyati, S.Si, M.Kom.

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA - UNPAK



Dr. Ir. Hermawan

Mengetahui,

Ketua Program Studi Ilmu Komputer
FMIPA – UNPAK



Arie Qur'ania, M.Kom.

Dekan

FMIPA – UNPAK


Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI

Dengan ini saya menyatakan bahwa:

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian di mana sumber informasinya dicantumkan dengan cara yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, 6 November 2024



Muhammad Faisal Raafi
065119132

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Saya yang bertandatangan di bawah ini :

Nama : Muhammad Faisal Raafi
NPM : 065119132
Judul Skripsi : Penerapan *Convolutional Neural Networks*
(CNN) Dalam Klasifikasi Tingkat Kematangan
Pucuk Daun Teh

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, 6 November 2024



Muhammad Faisal Raafi
065119132

RIWAYAT HIDUP



Muhammad Faisal Raafi, lahir di Cisarua, Bogor pada tanggal 26 Februari 2001 dari pasangan Bapak Mukrori dan Ibu Tasmi. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis memulai pendidikan di TK Pertiwi Desa Cibeureum, kemudian melanjutkan di SDN 01 Cibeureum, SMP YPC Cisarua hingga menjadi alumni dari SMA N 1 Cisarua. Pada tahun 2019 penulis meneruskan pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Penulis aktif mengikuti organisasi tingkat Fakultas. Pada tahun 2019 penulis menjadi anggota luar biasa, kemudian menjadi anggota kepengurusan BEM periode 2021 di departemen pendidikan, budaya, dan olahraga. Penulis juga sempat menjadi anggota panitia pada program dan menjadi ketua panitia Piala MIPA 2021. Setelah demisioner dari BEM, penulis melanjutkan organisasinya menjadi kepengurusan BLM 2022 Ketua Komisi 2 bidang keilmuan dan kelembagaan. Pada bulan November 2022 penulis melaksanakan Praktik Lapang di Ponpes Tahfizh Entrepreneur Thusina YBMPLN. Pada bulan November 2023 penulis memulai penelitian tugas akhir skripsi dan selesai pada bulan Juli 2024 dengan judul “Penerapan *Convolutional Neural Networks* (CNN) Dalam Klasifikasi Tingkat Kematangan Pucuk Daun Teh” untuk menuntaskan kewajiban sebagai mahasiswa dan mendapatkan gelar Sarjana Ilmu Komputer. Peneliti juga meraih Juara 2 dalam acara pameran poster hasil penelitian & karya ilmiah ICOTENS 2024 dengan tema “*Science and Technology Research for Future Sustainable Development*” yang diselenggarakan oleh FMIPA Universitas Pakuan pada 1 Agustus 2024.

KATA PENGANTAR

Segala Puji Syukur ke Hadirat Allah SWT, karena rahmat dan hidayah- Nya penulisan dapat menyelesaikan skripsi ini yang berjudul “Penerapan *Convolutional Neural Networks* (CNN) Dalam Klasifikasi Tingkat Kematangan Pucuk Daun Teh”.

Penulisan skripsi ini dibuat dalam rangka menuntaskan salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Ilmu Komputer FMIPA UNPAK Bogor.

Pada kesempatan ini, izinkan penulis menyampaikan hormat dan terima kasih sebesar-besarnya kepada :

1. Dr. Ir. Hermawan, selaku Pembimbing Utama yang telah memberikan dorongan moral dan motivasi kepada penulis.
2. Mulyati, M.Kom, selaku pembimbing Pendamping yang telah memberikan bimbingan, semangat dan motivasi.
3. Arie Qur'ania, M.Kom, selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Pakuan Bogor.
4. Kedua Orang Tua yang selalu memberikan dukungan penuh dan do'a.
5. Saudara kakak yang memberikan semangat selama penelitian.
6. Teman bermain grup Bikini Bottom yang banyak memberikan cerita pengalaman melalui *game*.
7. Kawan Band CORD yang selalu memberikan semangat melalui musik.

Penulis menyadari ini masih terdapat banyak kekurangan. Oleh karena itu, dengan segala kerendahan hati penulis mengharapkan kritik dan saran demi tercapainya sebuah laporan yang bermanfaat serta dapat dipertanggungjawabkan isinya.

Bogor, 6 November 2024

Muhamad Faisal Raafi
065119132

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	i
PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI.....	ii
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA.....	iii
RIWAYAT HIDUP	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Ruang Lingkup.....	2
1.4. Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	3
2.1 Tanaman Teh.....	3
2.1.1 Pemangkasan Tanaman Teh.....	3
2.1.2 Pemetikan Pucuk Daun Teh	3
2.2 <i>Deep Learning</i>	4
2.3 <i>Convolutional Neural Network (CNN)</i>	4
2.3.1 Model CNN.....	4
2.3.2 Arsitektur CNN	7
2.4 TensorFlow	8
2.5 <i>Transformation Data</i>	9
2.6 Keras	9
2.7 <i>Confusion Matrix</i>	9
2.8 Penelitian Terdahulu	10
2.9 Tabel Perbandingan Penelitian.....	12
BAB III METODOLOGI PENELITIAN	13
3.1. Metode Penelitian	13
3.1.1 Data Selection	13
3.1.2 Data Preprocessing.....	13
3.1.3 Data Transformation	14
3.1.4 Data Mining	14
3.1.5 Evaluasi	14
3.2 Waktu dan Tempat Penelitian.....	14
3.2.1 Waktu Penelitian	14
3.2.2 Tempat Penelitian	14
3.3 Alat.....	14
3.4 Bahan	14
BAB IV RANCANGAN DAN IMPLEMENTASI.....	15
4.1 Tahap Data Selection	15
4.2 Tahap Data Preprocessing.....	16
4.3 Tahap Data Transformation	17

4.4	Tahap Data Mining	18
4.4.1	Arsitektur LeafNet	20
4.4.2	Arsitektur VGGNet11	21
4.5	Tahap Evaluasi	21
BAB V HASIL DAN PEMBAHASAN		24
5.1	Hasil	24
5.1.1	<i>Data Selection</i>	24
5.1.2	<i>Data Preprocessing</i>	24
5.1.3	<i>Data Transformation</i>	24
5.1.4	<i>Data Mining</i>	24
5.1.4.1	Arsitektur LeafNet	24
5.1.4.2	Pelatihan Arsitektur VGGNet11	26
5.2	Pembahasan	28
5.2.1	Evaluasi	28
BAB VI KESIMPULAN		31
6.1	Kesimpulan	31
6.2	Saran	31
DAFTAR PUSTAKA		32
LAMPIRAN		34

DAFTAR GAMBAR

	Halaman
Gambar 1. Tanaman Teh.....	3
Gambar 2. (a) Ranting Dengan Kucup Burung, (b) Ranting Peko	4
Gambar 3. Model Dasar CNN.....	5
Gambar 4. <i>Convolution Layer</i>	5
Gambar 5. <i>Pooling Layer</i>	5
Gambar 6. <i>Activation Function: ReLU</i>	6
Gambar 7. <i>Fully Connected Layer</i>	7
Gambar 8. Arsitektur LeafNet.....	8
Gambar 9. Arsitektur VGGNet11	8
Gambar 10. Tahapan KDD.....	13
Gambar 11. Ilustrasi Pengambilan Citra Daun Teh	15
Gambar 12. <i>Flowchart data preprocessing</i>	17
Gambar 13. <i>Resize Citra</i>	17
Gambar 14. Proses Transformasi Data.....	18
Gambar 15. <i>Flowchart Convolutional Neural Network</i>	18
Gambar 16. Perhitungan Lapisan Konvolusi	19
Gambar 17. Perhitungan ReLU	19
Gambar 18. Perhitungan Lapisan <i>Max Pooling</i>	20
Gambar 19. Dataset Label	24
Gambar 20. Data Setelah Proses <i>Data Transformation</i>	24
Gambar 21. Prediksi Daun Teh Belum Matang	29
Gambar 22. <i>Confusion Matrix</i> (a) Arsitektur LeafNet, (b) Arsitektur VGGNet11..	29

DAFTAR TABEL

	Halaman
Tabel 1. Tabel Perbandingan.....	12
Tabel 2. Alat.....	14
Tabel 3. Kelas Tingkat Kematangan Pucuk Daun Teh	16
Tabel 4. Perhitungan Lapisan <i>Softmax</i>	20
Tabel 5. Perhitungan <i>Confusion Matrix</i>	22
Tabel 6. Perhitungan Evaluasi.....	23
Tabel 7. Parameter Arsitektur LeafNet	24
Tabel 8. Perbandingan Grafik Akurasi dan <i>Loss</i> Arsitektur LeafNet	25
Tabel 9. Hasil Pelatihan Model LeafNet berdasarkan Proporsi Dataset	26
Tabel 10. Parameter Arsitektur VGGNet11	26
Tabel 11. Perbandingan Grafik Akurasi dan Loss Arsitektur VGGNet11.....	27
Tabel 12. Hasil Pelatihan Model VGGNet11 Berdasarkan Proporsi Dataset	28
Tabel 13. <i>accuracy, precision, recall</i> , dan <i>F1-Score</i> Pada Arsitektur LeafNet	29
Tabel 14. <i>accuracy, precision, recall</i> , dan <i>F1-Score</i> Pada Arsitektur VGGNet11 ..	30

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Keputusan.....	35
Lampiran 2. Kartu Bimbingan	37
Lampiran 3. Surat Permohonan Pengambilan Data	38
Lampiran 4. Dataset Pucuk Daun Teh	39
Lampiran 5. Kode Python	39
Lampiran 6. Proses <i>Data Prerocessing</i>	41
Lampiran 7. Tabel Prediksi	41
Lampiran 8. <i>Confusion Matrix</i>	48
Lampiran 9. PTPN VIII Gunung Mas.....	51

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teh merupakan bahan minuman yang dibuat dari pucuk matang yang telah mengalami proses pengolahan tertentu. Teh merupakan salah satu komoditas ekspor sejak lama di Indonesia. Indonesia menjadi salah satu negara penghasil dan pengeksport teh utama di dunia dengan pemasaran tersebar ke negara-negara konsumen teh. Kontribusi teh Indonesia sebagian besar berasal dari Jawa Barat dengan produksi sebesar 66,93%. Pengembangan tanaman teh saat ini dan masa datang dengan kondisi tanaman yang semakin tua sehingga perlu dimaksimalkan proses metabolismenya melalui pemeliharaan tanaman teh (Anjarsari et al., 2020).

Pengamatan visual adalah salah satu metode yang digunakan untuk menentukan tingkat kematangan pucuk daun teh selama proses pengolahan. Pada penelitian Ranandita Putra dan Sunardi (2022), pengamatan visual dilakukan dengan memeriksa perubahan warna dan tekstur pucuk daun teh selama proses pengeringan. Kematangan daun teh dapat diidentifikasi dari perubahan warna daun yang cenderung menjadi lebih gelap dan teksturnya yang lebih kering dan rapuh.

Namun, pengamatan visual yang dilakukan oleh mandor pemanenan pucuk daun teh sering kali mengalami kesalahan waktu pemetikan. Ketika musim hujan, pucuk daun teh harus segera dipanen untuk menghindari pucuk daun teh melewati masa panen sehingga mengurangi kualitas tanaman teh. Salah satu pendekatan teknologi yang dapat digunakan untuk memberikan solusi permasalahan tersebut adalah dengan menerapkan algoritma *Convolutional Neural Networks* (CNN). Algoritma CNN memiliki kelebihan utama dalam menangani data visual secara efisien melalui lapisan konvolusi yang mampu mengenali pola dan fitur yang kompleks. LeafNet merupakan salah satu metode ekstraksi fitur untuk klasifikasi daun menggunakan *deep learning*. Kelebihan dari arsitektur LeafNet adalah jumlah parameter yang sedikit, sehingga lebih ringan dan cepat dalam pelatihan. Kelebihan dari arsitektur VGGNet11 adalah lebih banyak lapisan konvolusi yang memungkinkan menangkap fitur yang lebih kompleks dan beragam (Beikmohammadi et al., 2019).

Beberapa penelitian terdahulu yang membahas tentang metode CNN diantaranya, yaitu Pujiati & Rochmawati, (2022) dengan judul “*Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode Convolutional Neural Network (CNN)*”. Pada penelitian tersebut menggunakan metode arsitektur CNN untuk mengidentifikasi daun herbal dengan melalui tahap *preprocessing*. Dataset yang digunakan sebanyak 21.450 citra tumbuhan dengan 33 kelas tumbuhan herbal, dimana dataset dibagi menjadi 3 bagian, yaitu 16.500 data *training*, 3.300 data validasi, dan 1.650 data *testing*. Penelitian tersebut menghasilkan akurasi tertinggi sebesar 94% dengan *loss* terendah 0,28 dari proses *training* dan validasi sebanyak 150 *epoch*.

Penelitian lainnya Ibrahim et al., (2022) dengan judul “*Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network*”. Pada penelitian tersebut mengklasifikasi daun teh dari jenis sampel varietas Assamica Klon (GMB 7) yang berada pada pusat penelitian Teh dan KinaGambung. Metode yang diterapkan pada penelitian ini adalah metode *Convolutional Neural Network* (CNN) menggunakan arsitektur VGGNet19 dan ResNet50. Hasil pengujian sistem memperoleh nilai akurasi terbaik sebesar 97.5% menggunakan arsitektur VGGNet19 dengan *Optimizer* RMSprop, *learning rate* 0.01, *batch size* 32 dan *epoch* 100.

Penelitian lainnya Usman et al., (2021) dengan judul “Evaluasi *Optimizer* pada *Residual Network* untuk Klasifikasi Klon Teh Seri GMB Berbasis Citra Daun”. Pada penelitian tersebut mengidentifikasi terhadap 11 klon teh seri GMB (GMB-1 hingga GMB-11). Metode yang diterapkan pada penelitian ini adalah CNN dengan arsitektur ResNet101. Hasil pengujian dari evaluasi sistem dengan membandingkan tujuh algoritma *optimizer* yang berbeda, yaitu Adam, SGD, RMSprop, AdaGrad, AdaMax, AdaDelta dan rata-rata presisi, *recall* dan *F1-Score* terbaik sebesar 96% dan akurasi terbaik sebesar 97%.

Berdasarkan permasalahan yang ada pada latar belakang dan penelitian terdahulu, maka dilakukanlah penelitian yang menerapkan *Convolutional Neural Networks* (CNN) dalam klasifikasi tingkat kematangan pucuk daun teh. Diharapkan pada penelitian ini dapat membantu mengklasifikasi tingkat kematangan pucuk daun teh dalam jumlah yang besar dan mencapai nilai akurasi tinggi.

1.2. Tujuan

Penelitian ini bertujuan untuk menerapkan metode *Convolutional Neural Network* (CNN) dalam klasifikasi tingkat kematangan pucuk daun teh.

1.3. Ruang Lingkup

Ruang Lingkup dalam penelitian ini meliputi:

1. Lokasi penelitian di PTPN VIII Gunung Mas, Desa Tugu Selatan, Kecamatan Cisarua, Kabupaten Bogor.
2. Data berupa daun kedua dan ketiga dari pucuk pek dan kuncup burung daun teh. Jenis daun teh yang diteliti adalah TRI.
3. Dataset sebanyak 1000 data citra yang diambil menggunakan kamera *smartphone* secara langsung, kemudian pembagian dataset dibagi menjadi 3 proporsi yang berbeda, yaitu 70:15:15, 80:10:10, dan 90:5:5.
4. Implementasi sistem menggunakan bahasa pemrograman *Python*.
5. Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur LeafNet dan VGGNet11.

1.4. Manfaat Penelitian

Adapun manfaat pada penelitian, diantaranya:

1. Dapat mengklasifikasi tingkat kematangan pucuk daun teh dengan menggunakan metode CNN dengan arsitektur LeafNet dan VGGNet11
2. Dapat mengukur kemampuan model CNN dari klasifikasi tingkat kematangan.
3. Dapat menjadi sumber referensi untuk penelitian selanjutnya yang berkaitan tentang metode CNN pada klasifikasi daun.

BAB II

TINJAUAN PUSTAKA

2.1 Tanaman Teh

Teh (*Camellia sinensis*) merupakan jenis tanaman dari keluarga *Theaceae* yang memiliki banyak manfaat kesehatan. Teh merupakan salah satu minuman yang banyak dikonsumsi oleh masyarakat lokal maupun luar negeri. Hal ini dikarenakan teh memiliki rasa dan aroma yang khas. Syarat untuk tumbuh optimal tanaman teh adalah pada suhu 13-15°C dengan kelembaban relatif > 70%. Ketinggian tempat yang baik untuk pertumbuhan tanaman teh adalah 400 – 2.000 MPDL (Mada et al., 2018).



Gambar 1. Tanaman Teh

Berdasarkan dari Gambar 1 tanaman teh yang siap dipangkas dengan kriteria seperti tinggi bidang petik > 110cm, produktivitas menurun, dan pucuk burung > 70%. Pemangkasan tanaman teh dibutuhkan pengelolaan yang baik untuk menjaga tanaman dari kerusakan dari pemangkasan. Pemangkasan dapat dinyatakan berhasil yang dipengaruhi oleh jenis, cara pemangkasan, waktu pemangkasan, daur atau gilir pangkas, dan kemahiran pemangkas (Ranandita Putra & Sunardi, 2022).

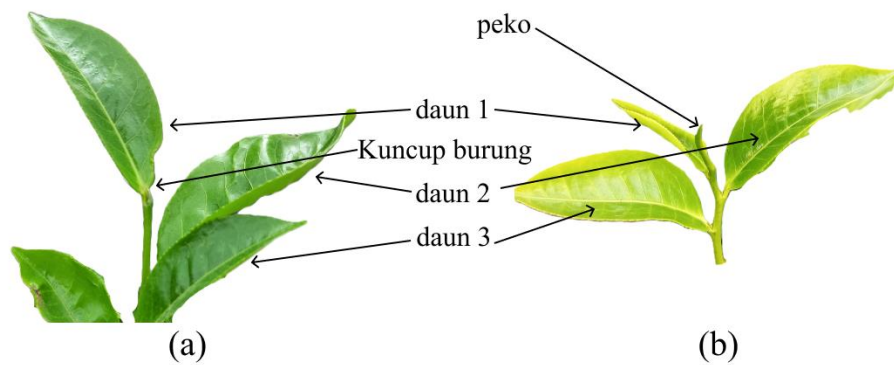
2.1.1 Pemangkasan Tanaman Teh

Pemangkasan daun teh dilakukan untuk meningkatkan produktivitas tanaman teh dan kualitas teh yang dihasilkan. Tujuan pemangkasan adalah membentuk bidang petik tetap rendah untuk memudahkan pemetikan. Sistem pemangkasan merupakan aturan ketinggian pangkasan yang diterapkan dalam satu siklus pangkas dibandingkan dengan siklus pangkas sebelumnya. Sistem ini setiap kali melakukan pemangkasan selalu menaikkan bidang pangkasan (3-5 cm) lebih tinggi dari bidang pangkasan sebelumnya sampai batas maksimal pada ketinggian 65 cm – 70 cm, kemudian turun kembali pada ketinggian 50-55 cm (Wiratno et al., 2022).

2.1.2 Pemetikan Pucuk Daun Teh

Pemetikan pucuk daun teh adalah proses pemetikan bagian kuncup tanaman teh yang terdiri dari dua hingga tiga daun muda untuk diolah menjadi berbagai jenis teh, seperti teh putih, teh hijau, teh oolong, dan teh hitam. Umumnya, pemetikan produksi secara berkelanjutan sesuai dengan jenis petikan yang digunakan hingga siklus pemangkasan berikutnya. Pemetikan pucuk daun teh bertujuan untuk memenuhi syarat-syarat pengolahan dimana tanaman mampu membentuk suatu kondisi yang memproduksi secara berkesinambungan (Agrikan, 2020).

Menurut buku budidaya dan pasca panen teh Wiratno et al., (2022), penamaan daun teh dapat disimak pada Gambar 2 agar aspek-aspek pemetikan mudah dimengerti.



Gambar 2. (a) Ranting Dengan Kucup Burung, (b) Ranting Peko

Beberapa faktor yang menentukan pertumbuhan pucuk daun teh, antara lain:

1. Umur pangkas yang melambat berakibat pada daur petik yang semakin panjang.
2. Bertambah tingginya letak kebun tanaman teh, maka pertumbuhan semakin lambat sehingga daur petik jadi panjang.
3. Daur petik lebih panjang pada saat musim kemarau dibandingkan musim hujan.
4. Tanaman teh yang sehat akan membuat daur petik lebih cepat dibandingkan dengan tanaman teh yang kurang sehat.

2.2 *Deep Learning*

Deep Learning adalah salah satu bagian dari *Machine Learning*, melibatkan penggunaan algoritma yang memodelkan abstraksi tingkat tinggi dari data. *Deep Learning* menggunakan serangkaian fungsi transformasi *non-linear* yang tersusun dalam lapisan-lapisan yang dalam dan kompleks. *Deep Learning* sangat baik untuk diterapkan pada *supervised learning*, *unsupervised learning* dan *semi-supervised learning* maupun *reinforcement learning* dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks dan sebagainya. Algoritma yang termasuk pada *deep learning* adalah algoritma *Convolutional Neural Network (CNN)*, *Recurrent Neural Network (RNN)* dan *Generative Adversarial Network (GAN)* (Cholissodin et al., 2020).

2.3 *Convolutional Neural Network (CNN)*

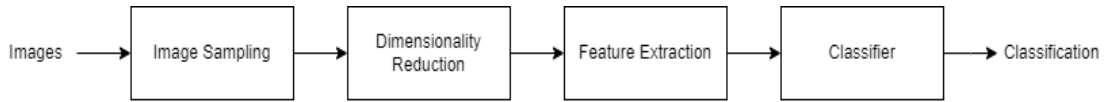
CNN atau *Convolutional Neural Network* merupakan salah satu model dari *Deep Learning* yang menyerupai dengan jaringan syaraf tiruan yang dapat divisualisasikan sebagai kumpulan neuron atau node yang disusun sebagai graf asiklik (graf yang tanpa adanya *loop* di dalamnya). CNN memiliki ciri khas yaitu terdapat lapisan tersembunyi (*hidden layer*) yang hanya terhubung ke subset neuron di lapisan sebelumnya. Karena konektivitas tersebut, CNN dapat mempelajari fitur hirarki yaitu filter yang dilatih untuk tujuan spesifik (Cholissodin et al., 2020).

2.3.1 Model CNN

Model umum dari *Artificial Neural network (ANN)* memiliki satu *layer input* dan *layer output*, serta beberapa *hidden layer*. Sebuah neuron tertentu menerima vektor *input X* dan menghasilkan *output Y* dengan melakukan fungsi *F* tertentu pada *input* tersebut, kemudian direpresentasikan pada persamaan umum (1).

$$F(X, W) = Y \quad (1)$$

Ada persamaan umum (1) tersebut W melambangkan vektor bobot yang mewakili kekuatan interkoneksi antara neuron-neuron dari dua *layer* yang saling berdekatan. Vektor bobot ini dapat digunakan untuk melakukan klasifikasi gambar (Agarwal et al., 2020).

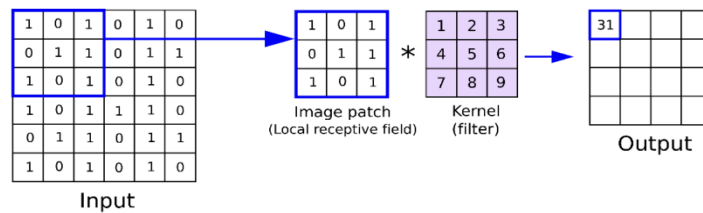


Gambar 3. Model Dasar CNN

Berdasarkan Gambar 3 model CNN terdiri dari 5 komponen utama, yaitu *convolution layer*, *pooling layer*, *activation function*, *fully connected layer* dan *softmax layer*. 5 lapisan dijelaskan sebagai berikut.

1. *Convolution Layer*

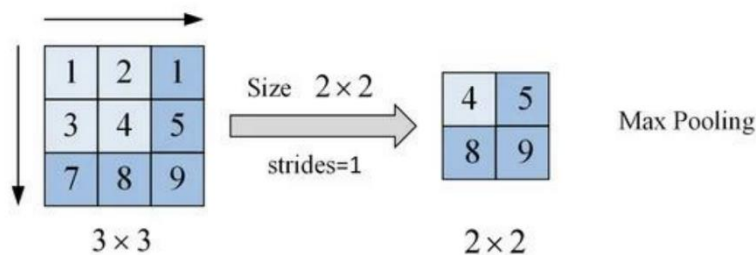
Lapisan konvolusi ini adalah lapisan pertama dalam CNN. Lapisan ini terdiri dari sejumlah filter konvolusi yang diterapkan pada input gambar. Setiap filter melakukan operasi konvolusi pada bagian input yang sesuai dan menghasilkan fitur-fitur lokal berdasarkan pola yang ditemukan. Filter ini bergerak secara bertahap di seluruh gambar untuk menghasilkan peta fitur (feature map) yang menggambarkan fitur-fitur yang diambil dari gambar (Chen et al., 2019). Contoh ilustrasi proses Convolution Layer dengan citra input 6x6 dan ukuran kernel 3x3 dapat dilihat pada Gambar 4.



Gambar 4. Convolution Layer

2. *Pooling Layer*

Lapisan penggabungan (*pooling*) biasanya ditempatkan setelah lapisan konvolusi. Tujuannya adalah untuk mengurangi dimensi spasial dari peta fitur yang dihasilkan oleh lapisan konvolusi. *Pooling* layer dibagi menjadi 2 macam, yaitu *Average Pooling* dan *Max Pooling*. Namun, salah satu metode *pooling* yang umum adalah *Max Pooling*, dimana nilai maksimum diambil dari sekelompok piksel dalam peta fitur. *Pooling* membantu mengurangi kompleksitas komputasi dan memperkenalkan invariansi terhadap translasi pada fitur-fitur yang ditemukan (Agarwal et al., 2020). Contoh *Max Pooling Layer* terhadap citra 3x3 menjadi 2x2 dengan jumlah langkah (*stride*) 1 yang dapat dilihat pada Gambar 5.



Gambar 5. Pooling Layer

3. *Activation Function*

Setelah lapisan konvolusi atau lapisan *pooling*, fungsi aktivasi diterapkan untuk memberikan sifat *non-linear* pada *output*. Fungsi aktivasi seperti ReLU (*Rectified Linier Unit*) atau Sigmoid digunakan untuk memperkenalkan *non-linearitas* dalam jaringan. Hal ini memungkinkan jaringan untuk mempelajari hubungan yang kompleks antara fitur-fitur yang diekstraksi (Chen et al., 2019). Fungsi persamaan ReLU dapat dilihat pada persamaan (2) dan grafik ReLU dapat dilihat pada Gambar 6.

$$f(x) = \max(0, x) \tag{2}$$

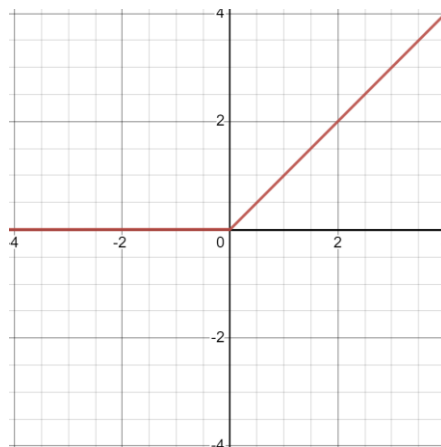
Kemudian fungsi ReLU tersebut dapat dijabarkan pada persamaan (3).

$$f(x) = \begin{cases} x, & 0 > x \\ 0, & x \leq 0 \end{cases} \dots \tag{3}$$

Keterangan:

$f(x)$ = fungsi ReLU

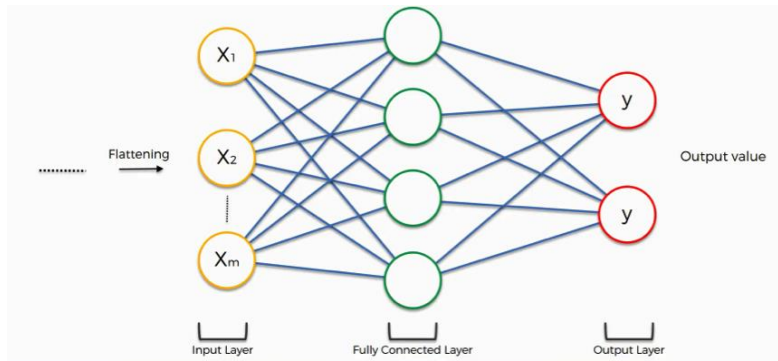
- Jika nilai x lebih besar 0, maka nilai x akan tetap
- Jika nilai x lebih kecil atau sama dengan 0, maka nilai x akan dibulatkan menjadi 0.



Gambar 6. *Activation Function: ReLU*

4. *Fully Connected Layer*

Lapisan terhubung penuh terletak di bagian akhir dari arsitektur CNN. Lapisan ini menghubungkan setiap neuron dari lapisan sebelumnya ke setiap neuron dalam lapisan ini. Lapisan ini bertindak sebagai klasifikasi akhir dan menghasilkan *output* berupa probabilitas atau label kelas dari gambar *input* (Ibrahim et al., 2022). Proses klasifikasi pada *fully connected layer* dapat dilihat pada Gambar 7.



Gambar 7. *Fully Connected Layer*

5. *Softmax Layer*

Lapisan *softmax* merupakan lapisan yang digunakan untuk mengubah output dari CNN menjadi probabilitas kelas output. Lapisan ini bekerja dengan mengambil nilai *output* dari CNN yang berupa vektor angka dan mengubahnya menjadi vektor probabilitas. Setiap probabilitas mewakili kemungkinan bahwa data masuk ke dalam kelas tertentu Fungsi *softmax* adalah fungsi yang digunakan untuk mengubah nilai *input* menjadi probabilitas. Probabilitas tersebut mewakili kemungkinan suatu nilai tertentu terjadi (Victor Zhou, 2019). Adapun rumus *softmax* ini pada persamaan (4).

$$Probabilitas(x) = \frac{e^x}{\sum e^x} \quad (4)$$

Keterangan:

x = nilai output dari *fully connected layer*

2.3.2 Arsitektur CNN

Terdapat ragam variasi arsitektur dalam penerapan dan pengembangan CNN. Arsitektur dijelaskan di bawah berikut.

1. Arsitektur LeafNet

Pada arsitektur LeafNet terdapat lima lapisan konvolusi, dua *fully connected layer*, dan lapisan terakhir sebagai *classification layer*. Selain itu, jumlah filter dalam lapisan konvolusi pertama, kedua dan kelima dirancang agar setengah dari jumlah yang digunakan dalam filter AlexNet. Selanjutnya, jumlah neuron dalam *fully connected layer* adalah 500, 100, dan 7, masing-masing berbeda dari jumlah yang digunakan dalam arsitektur standar AlexNet (Chen et al., 2019). Arsitektur ini dapat dilihat pada Gambar 8.

2.5 Transformation Data

Transformasi data adalah upaya untuk mengubah skala pengukuran data asli menjadi bentuk yang lain. Tujuan transformasi data adalah untuk memudahkan analisis statistik dan mengurangi kompleksitas data (Anwar hidayat, 2017).

Z-Score adalah metode statistik yang digunakan untuk menghitung seberapa jauh suatu nilai dari rata-rata data lainnya dengan dibagi standar deviasi data. Rumus Z-score dapat dilihat pada persamaan (5) berikut (Whendasmoro & Joseph, 2022).

$$Z - Score = \frac{(x - \mu)}{\sigma} \quad (5)$$

2.6 Keras

Keras adalah API untuk mengembangkan jaringan syaraf tiruan (*neural network*) yang dapat membuat sebuah *multi layer perceptron* (MLP) dan *convolutional neural network* (CNN). Pengaplikasian Keras sangat luas di mana dapat membangun jaringan syaraf tiruan seperti klasifikasi gambar, pemrosesan bahasa alami, pengenalan suara, dan prediksi *time series* (Dadan Dahman W., 2021).

2.7 Confusion Matrix

Confusion matrix adalah alat yang digunakan dalam evaluasi kinerja model klasifikasi dan menyatakan ketidakbenaran data untuk setiap kelas. Evaluasi dari *confusion matrix* dengan mencari nilai *accuracy*, *recall*, *precision*, dan *F1-Score* yang dapat dilihat berikut ini (Krstinić et al., 2020).

- a. *Accuracy*: Rasio antara jumlah prediksi yang benar dengan jumlah total data yang dievaluasi. Metrik ini memberikan gambaran umum tentang seberapa baik model dapat melakukan prediksi yang benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

- b. *Recall*: *Recall* dikenal juga sebagai *sensitivity* atau *true positive rate*, untuk mengukur seberapa baik model dapat mengidentifikasi semua *instance* yang relevan dari suatu kelas.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

- c. *Precision*: *Precision* dikenal juga sebagai *positive predictive value*, *Precision* merupakan presentase data yang diklasifikasi sebagai positif yang benar-benar positif.

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

- d. *F1-Score*: Merupakan ukuran akurasi yang menggabungkan *recall* dan *precision* untuk memberikan keseimbangan.

$$F1score = \frac{2 * recall * precision}{recall + precision} \quad (9)$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *False Negative*

2.8 Penelitian Terdahulu

Penelitian terdahulu terkait dengan penelitian yang akan dilakukan adalah sebagai berikut.

1. Nama : Rosida Pujiati dan Naim Rochmawati
Judul : Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode *Convolutional Neural Network* (CNN)
Tahun : 2022
Isi : Pada penelitian tersebut menggunakan metode arsitektur CNN untuk mengidentifikasi daun herbal dengan melalui tahap *preprocessing*. Dataset yang digunakan sebanyak 21.450 citra tumbuhan dengan 33 kelas tumbuhan herbal, dimana dataset dibagi menjadi 3 bagian, yaitu 16.500 data latih, 2.200 data validasi, dan 1.650 data *testing*. Dari penelitian ini menghasilkan akurasi tertinggi sebesar 94% dengan *loss* terendah 0,28 dari proses latih dan validasi sebanyak 150 *epoch*.
2. Nama : Nur Ibrahim, Gita Ayu Lestary, Faniesa Saufana Hanafi, haerudin Saleh, Nor Kumalasari Caecar Pratiwi, Muthia Syafika Haq, Adhi Irianto Mastur
Judul : Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode *Convolutional Neural Network*
Tahun : 2022
Isi : Pada penelitian tersebut mengklasifikasi daun teh dari jenis sampel *varietas assamica* Klon (GMB 7) yang berada pada pusat penelitian Teh dan KinaGambung. Metode yang diterapkan pada penelitian ini adalah metode *Convolutional Neural Network* (CNN) menggunakan arsitektur VGGNet19 dan ResNet50. Hasil pengujian sistem memperoleh nilai akurasi terbaik sebesar 97.5% menggunakan arsitektur VGGNet19 dengan *Optimizer* RMSprop, *learning rate* 0.01, *batch size* 32 dan *epoch* 100.
3. Nama : Koredianto Usman, Nor Kumalasari Caecar Pratiwi, Nur Ibrahim, Heri Syahrian, Vitria Puspitasari Rahadi
Judul : Evaluasi *Optimizer* pada *Residual Network* untuk Klasifikasi Klon Teh Seri GMB Berbasis Citra Daun
Tahun : 2021
Isi : Pada penelitian tersebut mengidentifikasi terhadap 11 klon teh seri GMB (GMB-1 hingga GMB-11). Metode yang diterapkan pada penelitian ini adalah CNN dengan arsitektur ResNet101. Hasil pengujian dari evaluasi sistem dengan membandingkan tujuh algoritma *optimizer* yang berbeda, yaitu Adam, SGD, RMSprop, AdaGrad, AdaMax, AdaDelta dan rata-rata presisi, *recall* dan *F1-Score* terbaik sebesar 96% dan akurasi terbaik sebesar 97%.

4. Nama : Naoya Takahashi, Yuki Mitsufuji
Judul : D3NET: *Densely Connected Multidilated Densenet For Music Source Separation*
Tahun : 2021
Isi : Penelitian ini membahas pemisahan sumber musik dalam sinyal audio dengan metode *Convolutional Neural Network* (CNN). Pendekatan sebelumnya menggunakan pemetaan fitur yang diperkecil dan diperbesar secara berurutan atau konvolusi yang membesar secara bertahap untuk dimasak model bidang yang benar. Metode yang diterapkan pada penelitian ini adalah CNN dengan arsitektur *Densely Connected Dilated DenseNet* (D3Net), yang menggabungkan konvolusi dilatasi multi-resolusi dengan DenseNet untuk menghindari masalah *aliasing*. Hasilnya menunjukkan bahwa D3Net mencapai kinerja terbaik dengan rasio sinyal terhadap distorsi (SDR) rata-rata sebesar 6,01 dB pada dataset MUSDB18.
5. Nama : Jing Chen, Qi Liu, Lingwang Gao
Judul : *Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model*
Tahun : 2019
Isi : Penelitian ini membahas klasifikasi jenis penyakit tanaman teh dari gambar daun. Metode yang digunakan CNN dengan arsitektur LeafNet dan untuk klasifikasi penyakit melalui algoritma *Support Vector Machine* (SVM) dan *multi-layer perceptron* (MLP). Hasilnya menunjukkan bahwa LeafNet memiliki nilai akurasi klasifikasi rata-rata sebesar 90,16%, sedangkan algoritma SVM sebesar 60,62% dan algoritma MLP sebesar 70,77%. Dengan nilai dari ketiga algoritma tersebut, LeafNet lebih unggul dibandingkan dengan algoritma SVM dan MLP.

2.9 Tabel Perbandingan Penelitian

Indikator yang akan dibandingkan berupa Arsitektur CNN dan *output* yang dapat dilihat pada Tabel 1.

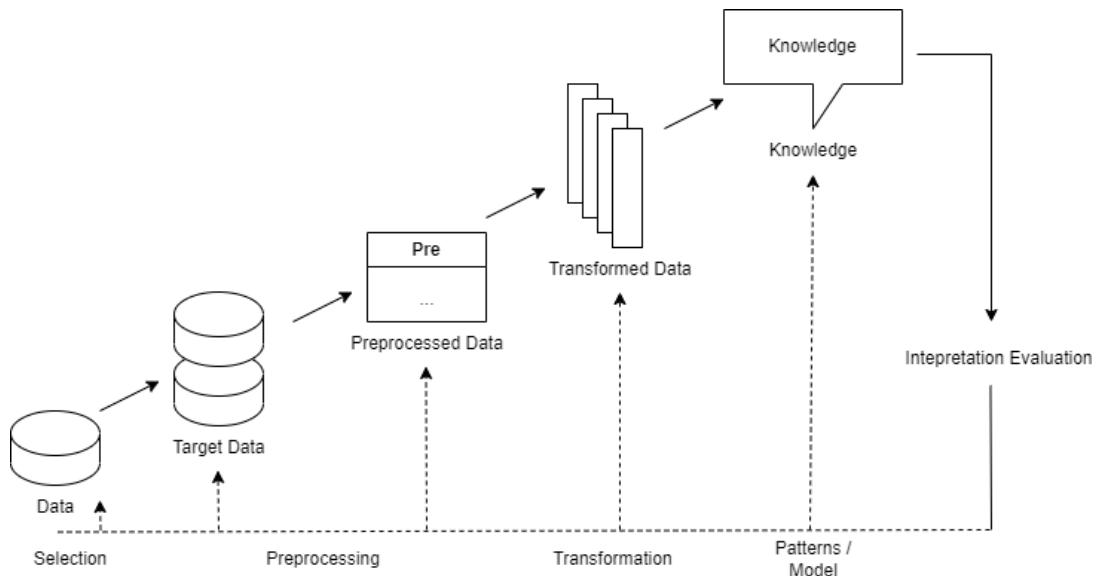
Tabel 1. Tabel Perbandingan

No.	Nama dan Tahun	Judul	Arsitektur CNN					Output	
			CNN	ResNet	D3Net	LeafNet	VGGNet	Aplikasi	Analisis
1.	Rosida Pujiati dan Naim Rochmawati (2022)	Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode <i>Convolutional Neural Network</i> (CNN)	✓					✓	✓
2.	Nur Ibrahim, Gita Ayu Lestary, Faniesa Saufana Hanafi, haerudin Saleh, Nor Kumalasari Caecar Pratiwi, Muthia Syafika Haq, Adhi Irianto Mastur (2022)	Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode <i>Convolutional Neural Network</i>					✓	✓	
3.	Koredianto Usman, Nor Kumalasari Caecar Pratiwi, Nur Ibrahim, Heri Syahrin, Vitria Puspitasari Rahadi (2021)	Evaluasi <i>Optimizer</i> pada <i>Residual Network</i> untuk Klasifikasi Klon Teh Seri GMB Berbasis Citra Daun		✓					✓
4.	Naoya Takahashi, Yuki Mitsufuji (2021)	D3NET: <i>Densely Connected Multidilated Densenet For Music Source Separation</i>			✓			✓	✓
5.	Jing Chen, Qi Liu, Lingwang Gao (2019)	<i>Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model</i>				✓			✓
6.	Muhamad Faisal Raafi (2023)	Penerapan <i>Convolutional Neural Networks</i> (CNN) Dalam Klasifikasi Tingkat Kematangan Pucuk Daun Teh				✓	✓		✓

BAB III METODOLOGI PENELITIAN

3.1. Metode Penelitian

Metode penelitian yang diterapkan adalah metode KDD (*Knowledge Discovery and Data Mining*). KDD ini merupakan metode untuk melakukan pengumpulan, pemakaian data, pemilihan data set yang memiliki ukuran besar (Suraya et al., 2023). Proses KDD ini dapat dilihat pada Gambar 10.



Gambar 10. Tahapan KDD

Adapun penjelasan setiap tahapan dari KDD adalah:

3.1.1 Data Selection

Tahapan data *selection* adalah tahapan untuk memilih data yang relevan untuk dianalisis. Ada beberapa aspek *selection* data pada penelitian ini:

- Kualitas Data: Data citra daun teh harus berkualitas baik, bebas dari *noise*.
- Representatifitas: Data citra daun teh mewakili berbagai tingkat kematangan daun teh.
- Jumlah Data: Data citra daun teh mencukupi untuk pelatihan pada model CNN.

Pada penelitian ini data dikumpulkan dengan Pengambilan citra secara langsung menggunakan kamera *smartphone* dengan pengaturan kamera yang disesuaikan untuk foto pucuk daun teh. Daun teh yang diambil merupakan daun pucuk dari tanaman teh. Proses pengambilan foto daun teh ketika kondisi cahaya yang konsisten agar citra yang dihasilkan merupakan citra yang baik.

3.1.2 Data Preprocessing

Pada *data preprocessing* data akan dibersihkan dari *noise* berupa piksel yang rusak, gangguan objek selain daun teh, dan gangguan lainnya. Cara pembersihan dilakukan dengan *resize* resolusi citra. Resolusi citra ini guna untuk pengurangan ukuran data yang lebih efisien sehingga dapat mempercepat pemrosesan dalam pelatihan model atau pengujian dan mengurangi kompleksitas penggunaan model yang lebih sederhana dan ringan.

3.1.3 Data Transformation

Tahap ini melibatkan normalisasi data, pembersihan data, penghapusan data yang tidak relevan, dan pengisian nilai yang hilang. Tujuannya adalah untuk memastikan bahwa data yang digunakan untuk pelatihan model *deep learning* adalah data yang berkualitas dan menghasilkan model yang lebih akurat. Z-Score merupakan salah satu metode normalisasi data. Metode ini mengubah nilai input pada citra menjadi rata-rata 0 dan deviasi standar 1 (Indolia et al., 2018).

3.1.4 Data Mining

Pada tahapan ini merupakan proses implementasi algoritma untuk ekstraksi pola dari data yang ada. Pada penelitian ini, algoritma yang diimplementasikan adalah CNN dengan arsitektur LeafNet dan VGGNet11. Tahapan *data mining* melalui lapisan-lapisan CNN, seperti *convolution layer*, *fully connected layer*, dan *classification layer*.

3.1.5 Evaluasi

Dalam evaluasi kemampuan model CNN dalam mengklasifikasikan tingkat kematangan pucuk daun teh diperlukan pengujian dengan data uji yang belum pernah dilihat sebelumnya. Evaluasi klasifikasi dapat dilakukan dengan metrik *accuracy*, *recall*, *precision* dan *F1-Score*.

3.2 Waktu dan Tempat Penelitian

3.2.1 Waktu Penelitian

Penelitian ini dimulai pada November 2023 sampai Juli 2024.

3.2.2 Tempat Penelitian

Penelitian ini dilakukan di PTPN VIII Gunung Mas, Perkebunan Teh, Jl. Raya Puncak KM 27, Kecamatan Cisarua, Kabupaten Bogor, Jawa Barat. Dokumentasi tempat penelitian dapat dilihat pada Lampiran 9.

3.3 Alat

Adapun perangkat lunak (*software*) dan perangkat keras (*hardware*) yang digunakan pada penelitian ini, diantaranya pada Tabel 2.

Tabel 2. Alat

No	Perangkat Lunak	Perangkat Keras
1.	Google Colaboratory	Laptop Asus Vivobook
2.	Google Drive	Smartphone Samsung Galaxy A32
3.	ChatGPT	Tensor Processing Unit (TPU)
4.	Benzin	

3.4 Bahan

Adapun bahan yang digunakan pada penelitian ini, diantaranya:

1. Dataset pucuk daun teh dengan jumlah 1000 foto.
2. *Library* TensorFlow
3. *Library* Keras
4. *Library* numpy
5. *Library* pandas
6. *Library* matplotlib

BAB IV RANCANGAN DAN IMPLEMENTASI

4.1 Tahap *Data Selection*

Pada tahap ini dilakukan proses pengumpulan data yang akan dilatih pada model CNN. Data diambil secara acak dari blok tanaman teh yang berbeda. Data berupa gambar pucuk daun teh yang diambil secara langsung menggunakan *smartphone*. Kemudian gambar dirotasikan horizontal dan dihilangkan *background* untuk menjaga konsistensi gambar daun dan disimpan di Google Drive.



Gambar 11. Ilustrasi Pengambilan Citra Daun Teh

Pada ilustrasi Gambar 11 dijelaskan ilustrasi pada saat pengambilan citra daun teh dengan menggunakan *smartphone* Samsung Galaxy A32. Beberapa pengaturan saat pengambilan foto, yaitu:

- Resolusi 1932x1576 piksel
- 5MP foto
- Mode macro*
- Menggunakan *flash*





Adapun ketentuan dalam pengambilan foto, yaitu:

- Jarak kamera dengan objek daun berkisar 3-5cm
- Skala foto disesuaikan dengan bentuk daun yang memanjang
- Foto diambil pada waktu pukul 09.00 WIB sampai dengan pukul 16.00 WIB.
- Daun teh yang difoto secara acak pada blok tanaman teh yang berbeda.

Dataset ini terdiri dari 1000 gambar yang nantinya akan dibagi dalam beberapa proporsi untuk tujuan pelatihan, validasi, dan pengujian. Proporsi pertama, 70% data latih berjumlah 700 gambar, 15% data validasi berjumlah 152 gambar, dan 15% data uji berjumlah 148 gambar. Proporsi kedua, 80% data latih berjumlah 800 gambar, 10% data validasi berjumlah 100 gambar, dan 10% data uji berjumlah 100 gambar. Proporsi

ketiga, 90% data latih berjumlah 900 gambar, 5% data validasi berjumlah 52 gambar, dan 5% data uji berjumlah 48 gambar. Setiap proporsi dan kelas dari dataset ini sudah dipisahkan untuk memudahkan dalam memuat data pada Google Colaboratory. Label dan kelas dataset daun teh diklasifikasikan berdasarkan tingkat kematangan daun teh yang memiliki ciri warna dan ukuran seperti pada Tabel 3.

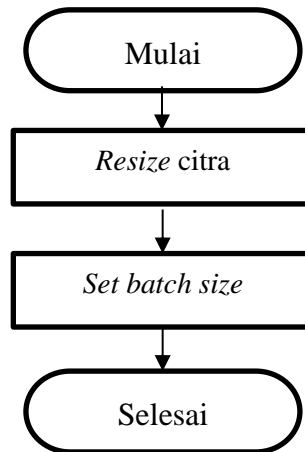
Tabel 3. Kelas Tingkat Kematangan Pucuk Daun Teh

Gambar	Tingkat Kematangan	Label
	Belum Matang	0
	Setengah Matang	1
	Matang	2
	Tua	3

Penelitian ini dilaksanakan selama rentang 8 bulan yang dimulai pada bulan November 2023 hingga bulan Juni 2024. Penelitian ini memakan waktu lama pada saat pengumpulan gambar pucuk daun teh yang sebanyak 1000 citra.

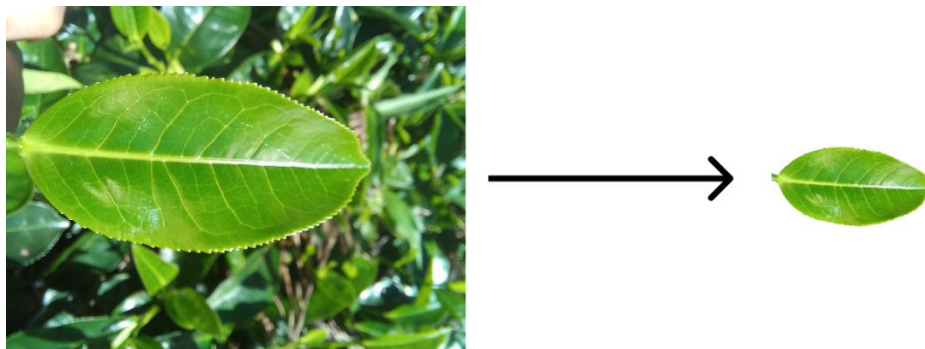
4.2 Tahap *Data Preprocessing*

Tahap dari *data preprocessing* ditampilkan menggunakan *flowchart* yang dapat dilihat pada Gambar 12.



Gambar 12. *Flowchart data preprocessing*

Gambar yang telah diambil dari dataset akan diubah ukurannya dengan *resize* citra agar memiliki dimensi yang konsisten. Hal ini penting karena konsistensi data mempermudah proses pelatihan model CNN dan meningkatkan efisiensi dalam komputasi. *Resize* citra dari resolusi 1932x2576 piksel menjadi resolusi 168x244 piksel. Berikut contoh *resize* citra pada Gambar 13.

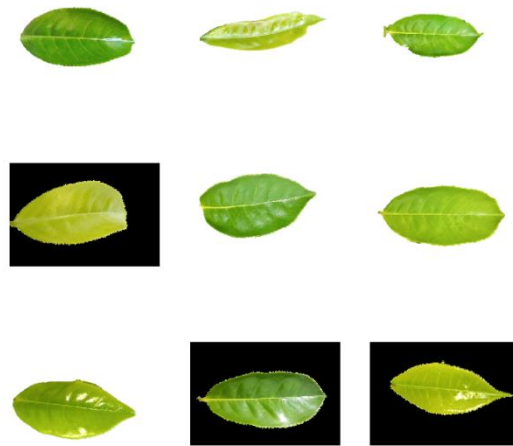


Gambar 13. *Resize Citra*

Selanjutnya, dilakukan pengaturan *batch size*, dimana ukuran *batch size* sebesar 30 untuk data latih dan data validasi agar selama proses pelatihan stabil, sedangkan ukuran *batch size* pada data uji disesuaikan dengan jumlah data uji dengan proporsi dataset yang berbeda. Proses *data preprocessing* dapat dilihat pada Lampiran 6.

4.3 Tahap Data Transformation

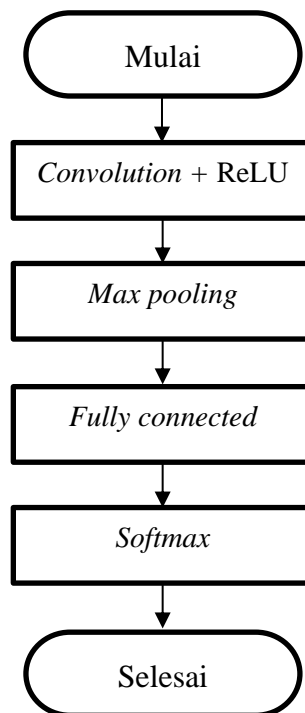
Pada tahap ini data akan dilakukan proses normalisasi citra dengan metode Z-Score dan augmentasi citra. Transformasi data ini bertujuan untuk meningkatkan performa dan akurasi model dalam mengklasifikasikan pucuk daun teh. Data setelah proses transformasi data dapat dilihat pada Gambar 14.



Gambar 14. Proses Transformasi Data

4.4 Tahap Data Mining

Pada tahap ini dilakukan proses pembuatan model CNN dengan arsitektur LeafNet maupun VGGNet11 yang ditampilkan menggunakan flowchart yang dapat dilihat pada Gambar 15.



Gambar 15. Flowchart Convolutional Neural Network

Algoritma CNN memiliki 5 lapisan utama dalam pelatihan model yang diantaranya *convolution layer*, *pooling layer*, *activation function*, *fully connected layer*, dan *softmax layer*. Perhitungan setiap lapisan sebagai berikut.

a. *Convolution Layer*

Lapisan ini menggunakan 32 filter ukuran 3x3 untuk mengekstrak fitur dari gambar daun teh dan berjalan ke seluruh gambar daun teh dan menghasilkan peta fitur.

10	22	43	20	15
17	33	32	29	30
16	24	33	33	42
10	8	33	33	32
10	11	15	15	23

1	0	1
0	1	0
1	0	1

 \times

102		

 $=$

...

10	22	43	20	15
17	33	32	29	30
16	24	33	33	42
10	8	33	33	32
10	11	15	15	23

1	0	1
0	1	0
1	0	1

 \times

102	99	133
92	103	127
74	83	113

 $=$

Gambar 16. Perhitungan Lapisan Konvolusi

Dari perhitungan konvolusi pada Gambar 16, dimana perhitungan proses konvolusi di piksel pertama adalah $= (10 * 1) + (22 * 0) + (43 * 1) + \dots = 102$ dan seterusnya hingga menghasilkan peta fitur baru. Fungsi Aktivasi ReLU digunakan untuk merubah nilai-nilai negatif. Hal ini dilakukan untuk meningkatkan efisiensi komputasi. Misalkan dari lapisan konvolusi terdapat nilai negatif yang kemudian dirubah nilainya dengan ReLU seperti pada di Gambar 17.

-32	20	20
2	-11	10
23	-41	23

 $=$

0	20	20
2	0	10
23	0	23

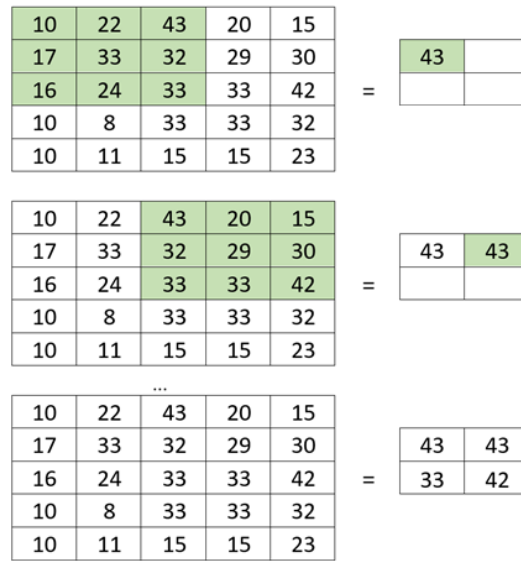
Gambar 17. Perhitungan ReLU

Maka, berdasarkan persamaan (8) perhitungan tersebut dapat dijabarkan berikut ini.

$$\begin{aligned}
 f(-32) &= -32 < 0 = 0 \\
 f(20) &= 0 > 20 = 20 \\
 f(-11) &= -11 < 0 = 0 \\
 &\dots \\
 f(23) &= 0 > 23 = 23
 \end{aligned}$$

b. *Max Pooling Layer*

Max pooling layer digunakan untuk mengurangi ukuran peta fitur. Pengurangan ukuran peta fitur dilakukan untuk mengurangi jumlah data yang perlu diproses oleh lapisan-lapian selanjutnya yang dapat meningkatkan efisiensi komputasi dan mengurangi overfitting. Misalkan, dari lapisan sebelumnya memiliki peta fitur 5x5 yang akan dikurangi jumlah datanya dengan 2 langkah seperti pada Gambar 18.



Gambar 18. Perhitungan Lapisan *Max Pooling*

c. *Softmax Layer*

Lapisan softmax mengubah nilai output dari fully connected layer menjadi probabilitas. Misalkan, output dari fully connected layer adalah 1.2, 0.9, 2.1, 0.4, maka dengan menggunakan persamaan 9 perhitungan softmax dapat dilihat pada Tabel 4.

Tabel 4. Perhitungan Lapisan Softmax

Tingkat Kematangan	x	e^x	$Probabilitas = \frac{e^x}{15,438}$
0_blm_matang	1,2	3,320	0,215
1_set_matang	0,9	2,460	0,159
2_matang	2,1	8,166	0,529
3_tua	0,4	1,492	0,097

Implementasi algoritma CNN pada penelitian ini menggunakan dua arsitektur CNN yang berbeda, yaitu LeafNet dan VGGNet. Dari kedua arsitektur ini akan dibandingkan untuk menentukan hasil yang optimal. Untuk mempercepat konvergensi dan meningkatkan stabilitas pelatihan digunakan *Optimizer Adam (Adaptive Moment Estimation)*. Model dilatih selama 50 *epoch* dan diterapkan teknik *EarlyStopping* untuk memastikan model tidak mengalami *overfitting* atau *underfitting*. *EarlyStopping* adalah teknik regulasi dalam pembelajaran mesin yang digunakan untuk mencegah *overfitting* dengan cara menghentikan proses pelatihan lebih awal (Muttineni Sai Rohith, 2022).

4.4.1 Arsitektur LeafNet

Lapisan-lapisan pada arsitektur LeafNet meliputi 5 lapisan konvolusi, 3 lapisan *max pooling*, lapisan *flatten*, 3 lapisan *fully connected* yang diikuti dengan *dropout* yang dijelaskan berikut ini:

1. Pada lapisan pertama, gambar input dengan ukuran 168x244 piksel dan 3 saluran warna (RGB) diproses oleh lapisan *convolutional* dengan 32 filter berukuran kernel 3x3 dan fungsi aktivasi ReLU. Setelah itu, dilakukan *max pooling* dengan ukuran 2x2 untuk mengurangi dimensi fitur.
2. Lapisan *convolutional* kedua menggunakan 64 filter dengan ukuran kernel 3x3

- dan fungsi aktivasi ReLU, diikuti dengan *max pooling* berukuran 2x2.
3. Lapisan *convolutional* ketiga menggunakan 128 filter dengan ukuran kernel 3x3 dan fungsi aktivasi ReLU.
 4. Lapisan *convolutional* keempat menggunakan 128 filter dengan ukuran kernel 3x3 dan fungsi aktivasi ReLU.
 5. Lapisan *convolutional* kelima menggunakan 128 filter dengan ukuran kernel 3x3 dan fungsi aktivasi ReLU, diikuti dengan *max pooling* berukuran 2x2.
 6. Setelah melalui lapisan-lapisan *convolutional* dan *pooling*, fitur-fitur yang telah diekstraksi dipadatkan menggunakan lapisan *flatten*. Kemudian, data diproses melalui lapisan *fully connected* pertama yang memiliki 512 neuron dengan fungsi aktivasi ReLU dan *dropout* sebesar 0.5 untuk mengurangi *overfitting*.
 7. Lapisan *fully connected* kedua memiliki 256 neuron dengan fungsi aktivasi ReLU dan *dropout* sebesar 0.5.
 8. Lapisan *fully connected* ketiga memiliki 128 neuron dengan fungsi aktivasi ReLU dan *dropout* sebesar 0.5.

4.4.2 Arsitektur VGGNet11

Lapisan-lapisan pada arsitektur VGGNet11 terdiri dari 6 lapisan konvolusi, 4 lapisan *max pooling*, lapisan *flatten*, 3 lapisan *fully connected* yang diikuti dengan *dropout* yang dijelaskan berikut ini:

1. Pada lapisan pertama, gambar input dengan ukuran 168x244 piksel dan 3 saluran warna (RGB) diproses oleh lapisan *convolutional* dengan 64 filter berukuran kernel 3x3, padding '*same*', dan fungsi aktivasi ReLU. Setelah itu, dilakukan *max pooling* dengan ukuran 2x2 dan *stride* 2x2 untuk mengurangi dimensi fitur.
2. Lapisan *convolutional* kedua menggunakan 128 filter dengan ukuran kernel 3x3, padding '*same*', dan fungsi aktivasi ReLU, diikuti dengan *max pooling* berukuran 2x2 dan *stride* 2x2.
3. Dua lapisan *convolutional* dengan 256 filter, masing-masing berukuran kernel 3x3, padding '*same*', dan fungsi aktivasi ReLU. Setelah itu, dilakukan *max pooling* dengan ukuran 2x2 dan *stride* 2x2.
4. Dua lapisan *convolutional* dengan 512 filter, masing-masing berukuran kernel 3x3, padding '*same*', dan fungsi aktivasi ReLU. Setelah itu, dilakukan *max pooling* dengan ukuran 2x2 dan *stride* 2x2.
5. Setelah melalui lapisan-lapisan *convolutional* dan *pooling*, fitur-fitur yang telah diekstraksi dipadatkan menggunakan lapisan *flatten*. Data kemudian diproses melalui lapisan *fully connected* pertama yang memiliki 512 neuron dengan fungsi aktivasi ReLU dan *dropout* sebesar 0.5 untuk mengurangi *overfitting*.
6. Lapisan *fully connected* kedua memiliki 256 neuron dengan fungsi aktivasi ReLU dan *dropout* sebesar 0.5.

4.5 Tahap Evaluasi

Dalam menguji kemampuan model CNN akan dievaluasi setiap hasil pelatihan model. Evaluasi data yang telah melalui pelatihan akan diuji menggunakan data uji. Selanjutnya akan dibuatkan prediksi untuk dijadikan *confusion matrix* yang bertujuan mengukur kemampuan klasifikasi dengan metrik *accuracy*, *recall*, *precision* dan *F1-Score*. Misalkan pada evaluasi dari *confusion matrix* pada Tabel 5.

Tabel 5. Perhitungan *Confusion Matrix*

		Prediksi			
		0_blm_matang	1_set_matang	2_matang	3_tua
Aktual	0_blm_matang	20	2	2	1
	1_set_matang	3	18	3	1
	2_matang	1	2	20	1
	3_tua	1	3	3	18

Maka, berdasarkan persamaan (6), (7), (8) dan (9) hasil dari perhitungan dilakukan pada berikut ini.

a. *Accuracy*

$$Accuracy = \frac{20 + 18 + 20 + 18}{100} = 0,76$$

b. *Precision*

$$Precision_0 = \frac{20}{20 + 3 + 1 + 1} = 0,8$$

$$Precision_1 = \frac{18}{18 + 2 + 2 + 3} = 0,72$$

$$Precision_2 = \frac{20}{20 + 2 + 3 + 3} = 0,71$$

$$Precision_3 = \frac{18}{18 + 1 + 1 + 1} = 0,85$$

c. *Recall*

$$Recall_0 = \frac{20}{20 + 2 + 2 + 1} = 0,8$$

$$Recall_1 = \frac{18}{18 + 3 + 3 + 1} = 0,72$$

$$Recall_2 = \frac{20}{20 + 1 + 2 + 1} = 0,83$$

$$Recall_3 = \frac{18}{18 + 1 + 3 + 3} = 0,72$$

d. *F1 Score*

$$F1_0 = \frac{2 * 0,8 * 0,8}{0,8 + 0,8} = 0,8$$

$$F1_1 = \frac{2 * 0,72 * 0,72}{0,72 + 0,72} = 0,72$$

$$F1_2 = \frac{2 * 0,71 * 0,83}{0,71 + 0,83} = 0,76$$

$$F1_3 = \frac{2 * 0,85 * 0,72}{0,85 + 0,72} = 0,78$$

Setelah didapatkan nilai-nilai tersebut, dibuatkan tabel performa model pada Tabel 6.

Tabel 6. Perhitungan Evaluasi

Kelas data	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Support</i>
0_blm_matang	0,80	0,80	0,8	25
1_set_matang	0,72	0,72	0,772	25
2_matang	0,72	0,83	0,76	25
3_tua	0,85	0,72	0,78	25
<i>accuracy</i>		76		100

BAB V HASIL DAN PEMBAHASAN

5.1 Hasil

5.1.1 Data Selection

Pada tahap ini dilakukan pemilihan data dari 1000 data ke dalam 4 kelas tingkat kematangan pucuk daun teh. Dataset yang tersimpan pada Google Drive dapat diakses pada Lampiran 4. Dari ke empat kelas tersebut diberikan label seperti pada Gambar 19.

```
{'0_belum_matang': 0, '1_set_matang': 1, '2_matang': 2, '3_tua': 3}  
{'0_belum_matang': 0, '1_set_matang': 1, '2_matang': 2, '3_tua': 3}  
{'0_belum_matang': 0, '1_set_matang': 1, '2_matang': 2, '3_tua': 3}
```

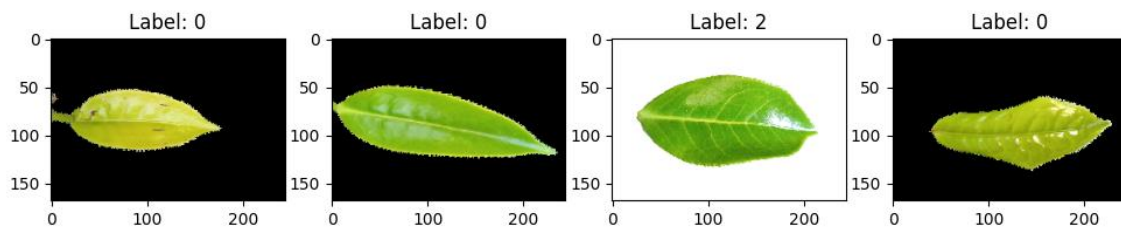
Gambar 19. Dataset Label

5.1.2 Data Preprocessing

Pada tahap ini dilakukan menghilangkan *background* citra menggunakan *website* benzin. Kemudian, ukuran resolusi citra *resize* menggunakan fungsi *resize* yang sudah disediakan oleh *library* TensorFlow.

5.1.3 Data Transformation

Metode *data transformation* yang diterapkan yaitu Z-Score pada setiap kelas yang telah diberi label. Data yang telah melalui proses transformasi dapat dilihat pada Gambar 20.



Gambar 20. Data Setelah Proses *Data Transformation*

5.1.4 Data Mining

Model CNN dibangun menggunakan bahasa *Python* dengan *library* TensorFlow. Kode *Python* ini dapat diakses pada Lampiran 5. Kode ditulis pada Google Colaboratory yang disediakan oleh Google. Kode dijalankan dengan memanfaatkan *hardware accelerator* TPU (*Tensor Processing Unit*) yang dikhususkan untuk model *deep learning* dengan tingkat komputasi yang tinggi.

5.1.4.1 Arsitektur LeafNet

Pembuatan model arsitektur LeafNet dengan memiliki lapisan-lapisan dan parameternya yang dapat dilihat pada Tabel 7.

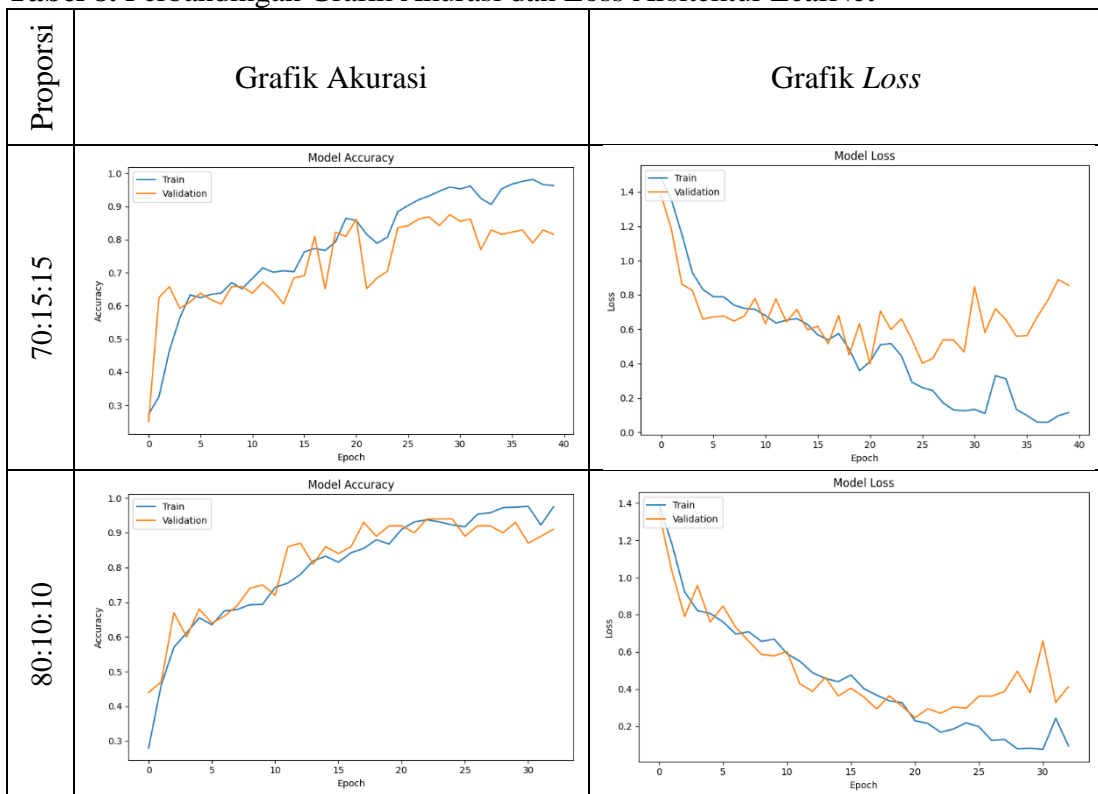
Tabel 7. Parameter Arsitektur LeafNet

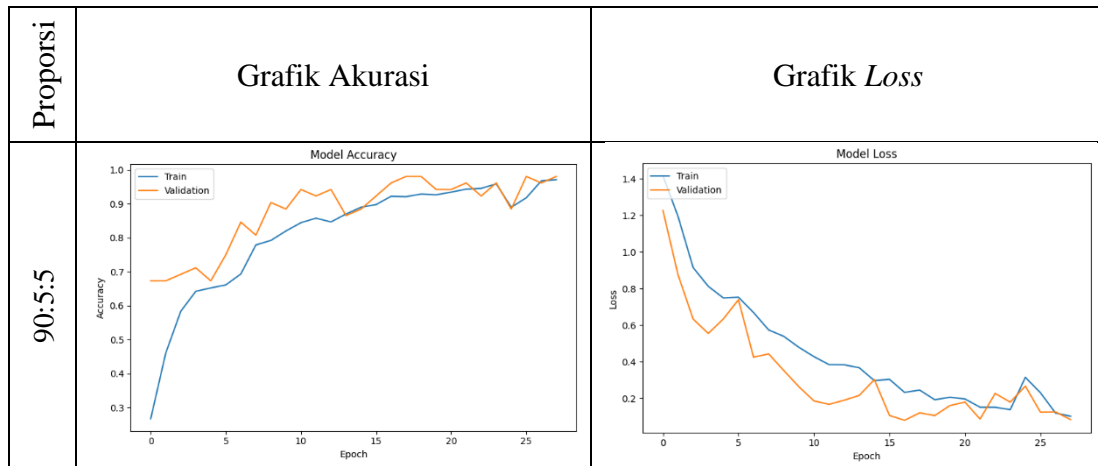
<i>Layer (type)</i>	<i>Output Shape</i>	<i>Param #</i>
conv2d	(None, 166, 242, 32)	896
max_pooling2d	(None, 83, 121, 32)	0
conv2d_1	(None, 81, 119, 64)	18496
max_pooling2d_1	(None, 40, 59, 64)	0

<i>Layer (type)</i>	<i>Output Shape</i>	<i>Param #</i>
conv2d_2	(None, 38, 57, 128)	73856
conv2d_3	(None, 36, 55, 128)	147584
conv2d_4	(None, 34, 53, 128)	147584
max_pooling2d_2	(None, 17, 26, 128)	0
flatten	(None, 56576)	0
dense	(None, 512)	28967424
dropout	(None, 512)	0
dense_1	(None, 256)	131328
dropout_1	(None, 256)	0
dense_2	(None, 128)	32896
dropout_2	(None, 128)	0
dense_3	(None, 4)	516
Total params		29520580 (112.61 MB)
Trainable params		29520580 (112.61 MB)
Non_trainable params		0 (0.00 Byte)

Perbandingan grafik akurasi dan *loss* arsitektur LeafNet pada setiap proporsi yang dapat disimak pada Tabel 8.

Tabel 8. Perbandingan Grafik Akurasi dan Loss Arsitektur LeafNet





Pada Tabel 8 menunjukkan arsitektur LeafNet berjalan dengan sangat baik seiring bertambahnya *epoch* dan penurunan *loss* yang rendah. Grafik pelatihan dari tiga proporsi data menunjukkan variasi performa model pada dataset dengan perbandingan (70:15:15), (80:10:10), dan (90:5:5). Pada proporsi pertama (70:15:15), terlihat bahwa akurasi *training* meningkat dengan baik, namun akurasi validasi tidak mengikuti tren yang sama dan bahkan menunjukkan fluktuasi yang signifikan. Hal ini diindikasikan oleh nilai *loss* validasi yang tetap tinggi dibandingkan dengan *loss* pada data *training* yang terus menurun. Hal ini menunjukkan bahwa model mengalami *overfitting*, yaitu model belajar terlalu baik pada data *training* hingga kehilangan kemampuan generalisasi pada data validasi. Sebaliknya, proporsi (80:10:10) menunjukkan keseimbangan yang lebih baik antara akurasi dan *loss* pada data *training* dan validasi, mengindikasikan performa yang lebih stabil dengan akurasi validasi mencapai 95.45% dan *loss* validasi 0.414. Proporsi (90:5:5) memperlihatkan hasil terbaik dengan akurasi validasi 98.08% dan *loss* validasi 0.084, menunjukkan bahwa model mampu mengenali pola dengan baik tanpa kehilangan kemampuan generalisasi, sehingga menjadi proporsi yang paling optimal di antara ketiganya. Angka presentasi dari ketiga proporsi tersebut dibuatkan tabel pada Tabel 9.

Tabel 9. Hasil Pelatihan Model LeafNet berdasarkan Proporsi Dataset

Proporsi	<i>Data train</i>		<i>Data validation</i>	
	<i>accuracy</i>	<i>loss</i>	<i>val_accuracy</i>	<i>val_loss</i>
70:15:15	96,29%	0,115	81,58%	0,855
80:10:10	97,50%	0,093	95,45%	0,414
90:5:5	97,11%	0,102	98,08%	0,084

5.1.4.2 Pelatihan Arsitektur VGGNet11

Pada arsitektur VGGNet11 memiliki lapisan-lapisan serta parameternya yang dapat dilihat pada Tabel 10.

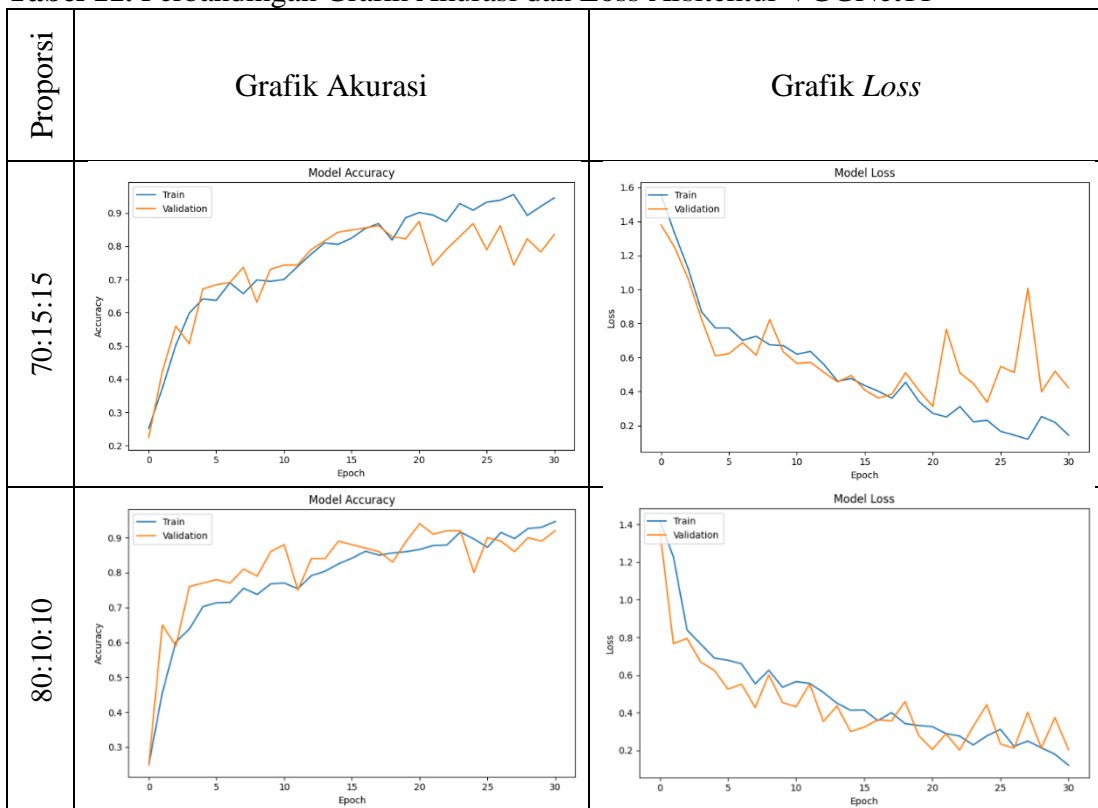
Tabel 10. Parameter Arsitektur VGGNet11

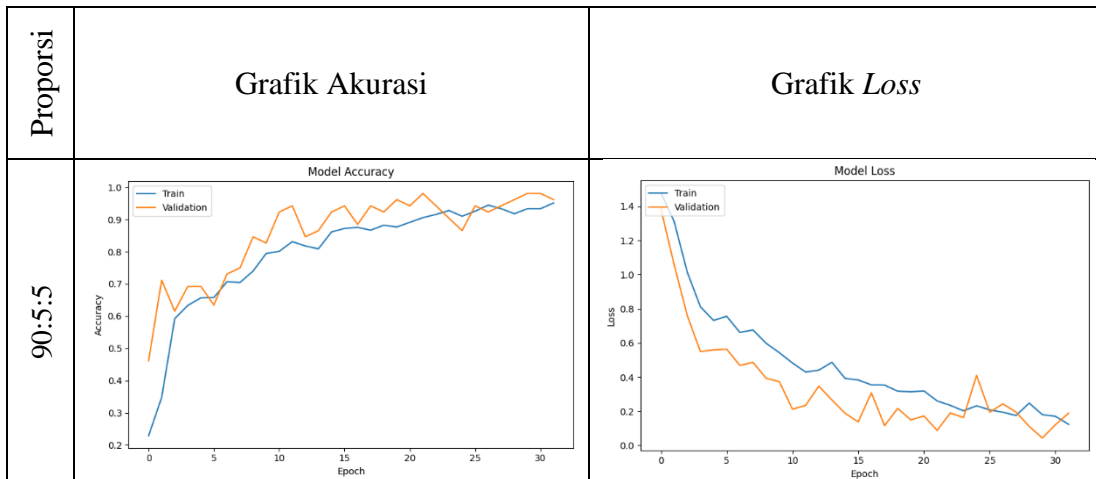
<i>Layer (type)</i>	<i>Output Shape</i>	<i>Param #</i>
conv2d	(None, 168, 244, 64)	1792
max_pooling2d	(None, 84, 122, 64)	0
conv2d_1	(None, 84, 122, 128)	73856
max_pooling2d_1	(None, 42, 61, 128)	0

<i>Layer (type)</i>	<i>Output Shape</i>	<i>Param #</i>
conv2d_2	(None, 42, 61, 256)	295168
conv2d_3	(None, 42, 61, 256)	590080
max_pooling2d_2	(None, 21, 30, 256)	0
conv2d_4	(None, 21, 30, 512)	1180160
conv2d_5	(None, 21, 30, 512)	2359808
max_pooling2d_3	(None, 10, 15, 512)	0
flatten	(None, 76800)	0
dense	(None, 512)	39322112
dropout	(None, 512)	0
dense_1	(None, 256)	131328
dropout_1	(None, 256)	0
dense_2	(None, 4)	1028
Total params		43955332(167.68 MB)
Trainable params		43955332(167.68 MB)
Non_trainable params		0 (0.00 Byte)

Perbandingan grafik akurasi dan *loss* arsitektur VGGNet11 pada setiap proporsi yang dapat disimak pada Tabel 11.

Tabel 11. Perbandingan Grafik Akurasi dan Loss Arsitektur VGGNet11





Pada Tabel 11 menunjukkan arsitektur VGGNet11 berjalan dengan sangat baik seiring bertambahnya *epoch* dan penurunan *loss* yang rendah. Pada proporsi pertama (70:15:15) menunjukkan kinerja yang kurang optimal. Meskipun mencapai akurasi pelatihan yang cukup tinggi, akurasi validasinya jauh lebih rendah, mengindikasikan adanya *overfitting*. Model ini cenderung menghafal pola-pola spesifik dalam data pelatihan, sehingga kurang mampu menggeneralisasi ke data baru. Selanjutnya, proporsi (80:10:10) menunjukkan peningkatan kinerja yang signifikan, terutama pada data validasi. Hal ini menunjukkan bahwa dengan meningkatkan jumlah data pelatihan, model menjadi lebih mampu menangkap fitur-fitur umum dalam data. Pada proporsi (90:5:5) ini mencapai akurasi tertinggi baik pada data pelatihan maupun validasi, serta nilai *loss* terendah. Ini mengindikasikan bahwa model telah berhasil belajar fitur-fitur yang relevan dan mampu menggeneralisasi dengan baik. Dari hasil eksperimen ini, dapat disimpulkan bahwa peningkatan proporsi data pelatihan secara signifikan meningkatkan kinerja model VGGNet11 pada tugas klasifikasi ini. Angka presentasi dari ketiga proporsi tersebut dibuatkan tabel pada Tabel 12.

Tabel 12. Hasil Pelatihan Model VGGNet11 Berdasarkan Proporsi Dataset

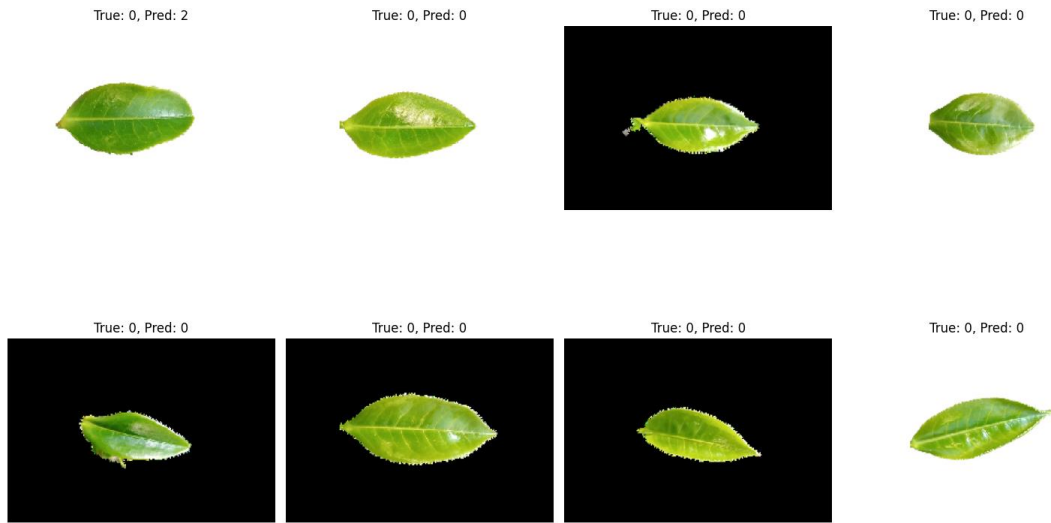
Proporsi	Data train		Data validation	
	accuracy	loss	val_accuracy	val_loss
70:15:15	94,57%	0,142	83,55%	0,421
80:10:10	94,63%	0,120	92,00%	0,203
90:5:5	95,11%	0,122	96,15%	0,188

5.2 Pembahasan

5.2.1 Evaluasi

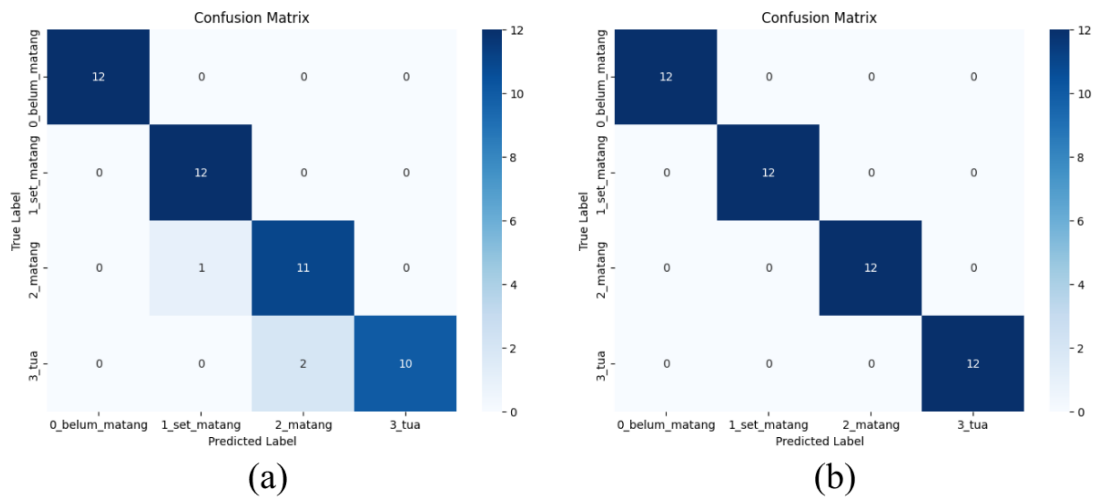
Tahap ini menguji kemampuan model CNN pada arsitektur LeafNet dan VGGNet11 dengan data uji. Dibuat prediksi yang selanjutnya dihitung dengan *confusion matrix* untuk dijadikan bahan klasifikasi dengan metrik *precision*, *recall*, dan *F1-Score*. Pada proporsi 90:5:5 dari kedua model mendapatkan akurasi yang paling tinggi dibandingkan proporsi lainnya. Kemudian, kedua model dilakukan perbandingan performa dengan melihat dari hasil evaluasi.

Prediksi klasifikasi dengan *output* berupa *true label* dan *predicted label*. Ketidakbenaran prediksi pada kelas kematangan daun teh dapat dilihat pada Gambar 21. Tabel prediksi yang lebih lengkapnya dapat dilihat pada Lampiran 7.



Gambar 21. Prediksi Daun Teh Belum Matang

Dari prediksi ini diukur lebih luas lagi dengan menggunakan perhitungan *confusion matrix*. *Confusion matrix* dari arsitektur LeafNet dan VGGNet11 dapat dilihat pada Gambar 22 dan lebih lengkapnya pada Lampiran 8.



Gambar 22. *Confusion Matrix* (a) Arsitektur LeafNet, (b) Arsitektur VGGNet11

Kemudian, laporan hasil performa dari kedua arsitektur CNN dapat dilihat pada Tabel 13 dan Tabel 14.

Tabel 13. *accuracy, precision, recall, dan F1-Score* Pada Arsitektur LeafNet

Kelas	<i>accuracy</i>		<i>precision</i>	<i>recall</i>	<i>F1-Score</i>	<i>support</i>
	<i>train</i>	<i>valid</i>				
belum_matang	97,11%	98,08%	1,00	1,00	1,00	12
set_matang			0,92	1,00	0,96	12
matang			0,85	0,92	0,88	12
tua			1,00	0,83	0,91	12
avg			0,94	0,94	0,94	48

Tabel 13 menunjukkan hasil evaluasi pada arsitektur LeafNet dengan akurasi data latih sebesar 97,11% dan data validasi sebesar 98,08%. Hasil ini menunjukkan bahwa arsitektur LeafNet mampu belajar secara efektif dalam menggeneralisasikan pada data validasi. Pada metrik *precision* dan *recall* terdapat variasi antar kelas, yaitu kelas matang memiliki *precision* lebih rendah sebesar 85% dan *recall* 92%. Hal tersebut menunjukkan bahwa model sedikit kesulitan dalam mengidentifikasi sampel dari kelas ini secara tepat tanpa memasukkan sampel dari kelas lain. Sebaliknya, kelas belum matang dan tua memiliki nilai *precision* dan *recall* yang tinggi. Hal ini menunjukkan keakuratan tinggi dalam pengklasifikasian dan deteksi sampel dari kelas tersebut.

Tabel 14. *accuracy, precision, recall, dan F1-Score* Pada Arsitektur VGGNet11

Kelas	<i>accuracy</i>		<i>precision</i>	<i>recall</i>	<i>F1-Score</i>	<i>support</i>
	<i>train</i>	<i>valid</i>				
belum_matang	95,11%	96,15%	1,00	1,00	1,00	12
set_matang			1,00	1,00	1,00	12
matang			1,00	1,00	1,00	12
tua			1,00	1,00	1,00	12
avg			1,00	1,00	1,00	48

Tabel 14 menunjukkan hasil evaluasi pada arsitektur VGGNet11 dengan akurasi data latih sebesar 95,11% dan data validasi sebesar 96,15%. Pada metrik *precision, recall, dan F1-Score* mencapai angka 100% pada semua kelas. Hal ini menunjukkan bahwa arsitektur VGGNet11 memiliki kapasitas yang baik dalam menangkap fitur khas yang membedakan semua kelas daun teh. Namun, hasil yang sempurna dalam metrik ini menunjukkan potensi *overfitting*, dimana model terlalu spesifik untuk data latih dan kurang fleksibel terhadap data baru. Hal tersebut dapat menunjukkan bahwa arsitektur LeafNet lebih mampu menggeneralisasi data dengan lebih variatif dibandingkan arsitektur VGGNet11.

BAB VI KESIMPULAN

6.1 Kesimpulan

Dalam proses pemanenan pucuk daun teh, sangat penting untuk menentukan pucuk daun teh yang berada di tingkat kematangan yang optimal guna memastikan kualitas dan rasa yang terbaik dari teh yang dihasilkan. Pucuk daun teh yang dipanen tidak hanya terbatas pada daun teh setengah matang dan matang, tetapi juga mencakup daun tua berdasarkan jenis petikannya. Terdapat tiga kategori petikan, antara lain petikan halus yang mencakup pucuk dengan satu daun muda, petikan medium yang mencakup dua hingga tiga daun muda, dan petikan kasar yang mencakup empat daun muda dan beberapa daun tua. Oleh karena itu, kelas daun teh dipanen berdasarkan jenis petikan tertentu.

Penelitian ini menerapkan metode KDD yang meliputi proses *data selection*, *data preprocessing*, *data transformation*, *data mining*, dan evaluasi. Algoritma *data mining* yang diimplementasikan adalah algoritma CNN untuk mengklasifikasikan tingkat kematangan pucuk daun teh. Proporsi dataset dilakukan pada 3 proporsi yang berbeda untuk mengetahui pengaruh proporsi pada pelatihan model yang diantaranya 70:15:15, 80:10:10, dan 90:5:5. Perbandingan dari ketiga proporsi tersebut, proporsi 90:5:5 lebih baik dari kestabilan pelatihan serta nilai akurasi tinggi dan *loss* yang rendah, sehingga proporsi ini lebih baik dibandingkan dengan proporsi lainnya.

Arsitektur CNN yang digunakan dalam pemodelan pada penelitian ini adalah arsitektur LeafNet dan VGGNet11. Kedua arsitektur ini memiliki perbedaan kompleksitas lapisan konvolusi. Akurasi validasi yang paling tinggi diperoleh pada arsitektur LeafNet dengan akurasi sebesar 98,08% dengan *loss* yang rendah sebesar 0,117. Hasil ini menunjukkan kemampuan arsitektur LeafNet lebih baik dalam menggeneralisasikan data baru. Sementara akurasi validasi pada arsitektur VGGNet11 lebih kecil.

6.2 Saran

Saran untuk penelitian ini mencakup beberapa hal yang diharapkan meningkatkan hasil kinerja pada penelitian selanjutnya. Pertama, penelitian selanjutnya diharapkan mengumpulkan lebih banyak data dan melakukan uji coba secara berulang untuk memperbaiki akurasi model, karena dataset yang lebih besar dapat membantu model belajar lebih baik dan menghindari indikasi *overfitting*. Kedua, direkomendasikan untuk mencoba metode atau arsitektur lain seperti InceptionNet, YOLO V5, atau arsitektur *deep learning* lainnya yang mungkin lebih sesuai dalam mengklasifikasi data. Ketiga, penelitian selanjutnya diharapkan untuk menyesuaikan *hyperparameters* model seperti *batch size*, *resize*, dan jumlah *epoch* secara lebih optimal agar performa model dapat meningkat secara keseluruhan.

DAFTAR PUSTAKA

- Agarwal, M., Gupta, S. Kr., & Biswas, K. K. (2020). Development of Efficient CNN model for Tomato crop disease identification. *Sustainable Computing: Informatics and Systems*, 28, 100407. <https://doi.org/https://doi.org/10.1016/j.suscom.2020.100407>
- Agrikan. (2020). *Pemetikan Pucuk Teh*. <https://agrikan.id/pemetikan-pucuk-teh/>
- Anjarsari, I. R. D., Ariyanti, M., & Rosniawaty, S. (2020). Studi ekofisiologis tanaman teh guna meningkatkan pertumbuhan, hasil, dan kualitas teh. *Kultivasi*, 19(3), 1181–1188. <http://jurnal.unpad.ac.id/kultivasi/article/view/26623>
- Anwar hidayat. (2017). *Pengertian dan Jenis Transformasi Data*. Statistikian.
- Beikmohammadi, A., Faez, K., beikmohammadi, ali, faez, karim, & motallebi, ali. (2019). *Article in Computers and Electronics in Agriculture* . <https://www.researchgate.net/publication/338435548>
- Chen, J., Liu, Q., & Gao, L. (2019). Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry*, 11(3). <https://doi.org/10.3390/sym11030343>
- Cholissodin, I., Sutrisno, Soebroto, A. A., Hasanah, U., & Febiola, Y. I. (2020). *AI, Machine Learning & Deep Learning (Teori & Implementasi): Vol. 1.01*. <https://www.researchgate.net/publication/348003841>
- Dadan Dahman W. (2021). *TensorFlow.Keras*. Howdy Sysinfo. <https://medium.com/sysinfo/tensorflow-keras-66dd489ae52f>
- Ibrahim, N., Lestary, G. A., Hanafi, F. S., Saleh, K., Pratiwi, N. K. C., Haq, M. S., & Mastur, A. I. (2022). Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 162. <https://doi.org/10.26760/elkomika.v10i1.162>
- Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, 132, 679–688. <https://doi.org/10.1016/j.procs.2018.05.069>
- Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). Multi-label Classifier Performance Evaluation with Confusion Matrix. *Faculty of Electrical Engineering, University of Split*, 01–14. <https://doi.org/10.5121/csit.2020.100801>
- Mada, G., Pertanian Gadajah Mada, F., Ayu, L., Indradewa, D., & Ambarwati, E. (2018). *THE GROWTH, YIELD AND QUALITY OF TEA TIP (Camellia Sinensis (L.) Kuntze) IN VARIOUS ELEVATIONS*.
- Muttineni Sai Rohith. (2022). *Keras EarlyStopping Callback to train the Neural Networks Perfectly*. Medium. <https://pub.towardsai.net/keras-earlystopping-callback-to-train-the-neural-networks-perfectly-2a3f865148f7>
- Pujiati, R., & Rochmawati, N. (2022). Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode Convolutional Neural Network (CNN). *Journal of Informatics and Computer Science*, 03.
- Ranandita Putra, R., & Sunardi, S. (2022). Pengantar Ilmu Pertanian. *PT GLOBAL EKSEKUTIF TEKNOLOGI*. <https://www.researchgate.net/publication/360484602>
- Serdar Yegulalp. (2024, January 5). *What is TensorFlow? The machine learning library explained*. InfoWorld. <https://www.infoworld.com/article/2255099/what-is-tensorflow-the-machine-learning-library-explained.html>

- Sonali Gupta. (2024). *VGG-11 Architecture*. OpenGenus. <https://iq.opengenus.org/vgg-11/>
- Suraya, Andayati, D., & Sholeh, M. (2023). Penerapan Metode Clustering dengan Algoritma K-Means Pada Pengelompokan Indeks Prestasi Akademik Mahasiswa. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, 6(1), 51–60.
- Usman, K., Pratiwi, N. K. C., Ibrahim, N., Syahrin, H., & Rahadi, V. P. (2021). Evaluasi Optimizer pada Residual Network untuk Klasifikasi Klon Teh Seri GMB Berbasis Citra Daun. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 9(4), 841. <https://doi.org/10.26760/elkomika.v9i4.841>
- Victor Zhou. (2019, July 22). *A Simple Explanation of the Softmax Function*. <https://victorzhou.com/blog/softmax/>
- Whendasmoro, R. G., & Joseph, J. (2022). Analisis Penerapan Normalisasi Data Dengan Menggunakan Z-Score Pada Kinerja Algoritma K-NN. *JURIKOM (Jurnal Riset Komputer)*, 9(4), 872. <https://doi.org/10.30865/jurikom.v9i4.4526>
- Wiratno, W., Yusron, M., Syakir, M., & Effendi, D. S. (2022). *Budidaya dan Pasca Panen TEH*. <https://www.researchgate.net/publication/362426981>

LAMPIRAN

Lampiran 1. Surat Keputusan



YAYASAN PAKUAN SILIWANGI
Universitas Pakuan
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
Unggul, Mandiri & Berakhlak Dalam Bidang MIPA

KEPUTUSAN DEKAN
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
No. : 4316/KEP/D/FMIPA/XII/2023

T E N T A N G

PENGANGKATAN PEMBIMBING TUGAS AKHIR
PADA PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN

DEKAN FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN,

- Menimbang : a. bahwa setiap mahasiswa tingkat akhir Program Strata Satu (S1) harus melaksanakan Tugas Akhir sebagaimana tercantum di dalam kurikulum setiap Program Studi di lingkungan Fakultas MIPA Universitas Pakuan.
b. bahwa untuk pelaksanaan Tugas Akhir diperlukan pengawasan dari pembimbing.
c. bahwa sehubungan dengan point a dan b di atas perlu dituangkan dalam suatu Keputusan Dekan.
- Mengingat : 1. Undang-undang RI No.: 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.: 60 Tahun 1999 tentang Pendidikan Tinggi.
3. Statuta Universitas Pakuan Tahun 2019.
4. Surat Keputusan Rektor Nomor: 35/KEP/REK/VIII/2020 tanggal 03 Agustus 2020 tentang Pemberhentian Dekan dan Wakil Dekan Masa Bakti 2015-2020 serta Pengangkatan Dekan dan Wakil Dekan Masa Bakti 2020-2025 di lingkungan Universitas Pakuan.
5. Ketentuan Akademik yang tercantum dalam Buku Panduan Studi Fakultas MIPA, Universitas Pakuan Tahun 2023.
- Memperhatikan : Usulan dari Ketua Program Studi Ilmu Komputer FMIPA UNPAK.

M E M U T U S K A N

- Menetapkan :
Pertama : Mengangkat pembimbing yang namanya tersebut di bawah ini :
1. Pembimbing Utama : Dr. Hermawan Taher
2. Pembimbing Pendamping : Mulyati, M.Kom.

Untuk membimbing dalam rangka melaksanakan tugas akhir bagi mahasiswa :

Nama : Muhamad Faisal Raafi
NPM : 065119132
Program Studi : Ilmu Komputer
Judul Skripsi : Penerapan Convolutional Neural Networks (CNN) Dalam Klasifikasi Tingkat Kematangan Pucuk Daun The

- Kedua : Kepada para pembimbing diharapkan dapat menjalankan tugasnya sebagai pembimbing dengan sebaik-baiknya.
- Ketiga : Dalam waktu 1 (satu) bulan setelah diterbitkannya SK ini, mahasiswa wajib melaksanakan Seminar Rencana Penelitian yang diselenggarakan oleh Program Studi Ilmu Komputer dengan dihadiri oleh Pembimbing dan Penguji.
- Keempat : Dana untuk honorarium pembimbing dibebankan kepada mahasiswa yang ketentuannya diatur oleh Fakultas MIPA.
- Kelima : Surat Keputusan ini berlaku untuk jangka waktu 1 (satu) tahun sejak tanggal ditetapkan sampai dengan mahasiswa tersebut Lulus Sidang/Ujian Skripsi, dengan ketentuan akan diadakan perubahan/perbaikan sebagaimana mestinya bila dikemudian hari terdapat kekeliruan dalam penetapannya.

Ditetapkan di : Bogor
Pada tanggal : 06 Desember 2023

Dekan,



Asep Denih, S,Kom., M.Sc., Ph.D.

Tembusan :




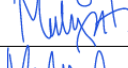





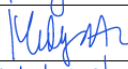

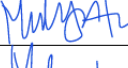

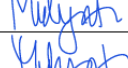

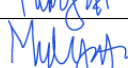

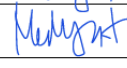

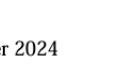

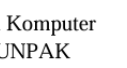
1. Yth. Ketua Program Studi Ilmu Komputer;
2. Yth. Dr. Hermawan Taher;
3. Yth. Mulyati, M.Kom.;
4. Arsip.

Lampiran 2. Kartu Bimbingan

Kartu Bimbingan Mahasiswa
Program Studi Ilmu komputer FMIPA – UNPAK

Nama Mahasiswa : Muhammad Faisal Raafi
NPM : 065119132
Judul Skripsi : Penerapan *Convolutional Neural Networks* (CNN) dalam
Klasifikasi Tingkat Kematangan Pucuk Daun Teh

Pembimbing Utama : Dr. Ir. Hermawan
Pembimbing Pendamping : Mulyati, M.Kom.

No	Hari, Tanggal	Catatan	Tanda Tangan	
			Pemb.I	Pemb.II
1.	Jum'at, 23 Juni 2023	Penentuan Judul, Metode, dan Objek Penelitian		
2.	Kamis, 31 Agustus 2023	Revisi Laporan Proposal BAB 1-4		
3.	Selasa, 21 November 2023	Revisi Laporan Proposal BAB 1-4		
4.	Senin, 4 Desember 2023	ACC Proposal		
5.	Rabu, 26 Juni 2024	Revisi Laporan Hasil BAB 3, 4, 5		
6.	Senin, 8 Juli 2024	Revisi Laporan Hasil BAB 1-6		
7.	Kamis, 11 Juli 2024	Revisi Laporan Hasil BAB 4-6		
8.	Kamis, 18 Juli 2024	ACC Laporan Hasil		
9.	Sabtu, 27 Juli 2024	Revisi Skripsi Penulisan		
10.	Selasa, 24 September 2024	Revisi Skripsi Penulisan		
11.	Kamis, 17 Oktober 2024	ACC Skripsi		

Bogor, 17 Oktober 2024

Program Studi Ilmu Komputer
Fakultas MIPA – UNPAK
Ketua,



Arie Qur'ania, M.Kom.

Lampiran 3. Surat Permohonan Pengambilan Data



YAYASAN PAKUAN SILIWANGI
Universitas Pakuan
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
Unggul, Mandiri & Berakhlak Dalam Bidang MIPA

Nomor : 4475/D/FMIPA-UP/XII/2023
Lampiran : -
Perihal : Permohonan Pengambilan Data

Kepada : Yth. PTPN VIII Gunung Mas,
Jl. Raya Puncak KM. 87 Gn Mas Puncak, Ds. Tugu Selatan,
Kec. Cisarua, Kab. Bogor, Jawa Barat, 16750

Dengan Hormat

Sehubungan dengan Pelaksanaan Tugas Akhir/Skripsi untuk Mahasiswa Program Studi Ilmu Komputer, Fakultas MIPA Universitas Pakuan dengan nama mahasiswa dibawah ini:

Nama : Muhamad Faisal Raafi
NPM : 065119132
Program Studi : Ilmu Komputer
Judul : Penerapan Convolutional Neural Networks (CNN) Dalam Klasifikasi Tingkat Kematangan Pucuk Daun Teh

Bermaksud mengadakan pengambilan data pada instansi yang Bapak/Ibu pimpin.

Adapun pengambilan data yang akan dilakukan mahasiswa kami meliputi tanya jawab, lisan, tertulis maupun observasi, sepanjang data-data yang diminta bukan merupakan rahasia yang menjadi tanggung jawab Bapak/Ibu.

Demikian permohonan ini kami sampaikan. Atas perhatian serta kerjasama yang baik, kami ucapkan terima kasih.

Bogor, 18 Desember 2023

Dekan,

Asep Denih, S.Kom., M.Sc., Ph.D.

Tembusan :

1. Yth. Wakil Dekan I FMIPA-UNPAK ;
2. Yth. Ketua Program Studi Ilmu Komputer ;
3. Arsip.

Jl. Pakuan P.O. Box 452, Bogor 16143, Telp./Fax. (0251) 8375547
Website : <https://fmipa.unpak.ac.id>

Lampiran 4. Dataset Pucuk Daun Teh

Untuk melihat dataset yang lengkap dapat mengakses pada *link* atau *QR Code* berikut:



<https://bit.ly/dataset-pucuk-daun-teh>

Lampiran 5. Kode Python

```
1 # definisikan lokasi dataset
2 data_train = "/content/drive/MyDrive/dataset/pucuk daun teh/80:10:10/data_train"
3 data_valid = "/content/drive/MyDrive/dataset/pucuk daun teh/80:10:10/data_valid"
4 data_test = "/content/drive/MyDrive/dataset/pucuk daun teh/80:10:10/data_test"
5
6 # Memuat gambar dari data latih
7 info_train = cv2.imread(data_train)
8 # Memuat gambar dari data validasi
9 info_valid = cv2.imread(data_valid)
10 # Memuat gambar dari data tes
11 info_test = cv2.imread(data_test)
12
13 datagen = ImageDataGenerator(rescale = 1.0/255.0)
14
15 train = datagen.flow_from_directory(
16     data_train,
17     target_size = (168, 244),
18     color_mode = 'rgb',
19     class_mode = 'categorical',
20     batch_size = 30
21 )
22
23 valid = datagen.flow_from_directory(
24     data_valid,
25     target_size = (168, 244),
26     color_mode = 'rgb',
27     class_mode = 'categorical',
28     batch_size = 64
29 )
30
31 test = datagen.flow_from_directory(
32     data_test,
33     target_size = (168, 244),
34     color_mode = 'rgb',
35     class_mode = 'categorical',
36     batch_size = 100,
37     shuffle=False
38 )
39
```

```

40 def resize_image(image):
41     return tf.image.resize(image, (168, 244))
42
43 # tambahkan fungsi resize_image ke pipeline data
44 train.preprocessing_function = resize_image
45 valid.preprocessing_function = resize_image
46 test.preprocessing_function = resize_image
47
48 # definisikan label data
49 train_labels = train.class_indices
50 valid_labels = valid.class_indices
51 test_labels = test.class_indices
52
53 # tampilkan label data
54 print(train_labels)
55 print(valid_labels)
56 print(test_labels)
57
58 def normalize_zscore(data):
59     mean = tf.reduce_mean(data)
60     std = tf.math.reduce_std(data)
61     return (data - mean) / std
62
63 # tambahkan fungsi normalize_zscore ke pipeline data
64 train.preprocessing_function = normalize_zscore
65 valid.preprocessing_function = normalize_zscore
66 test.preprocessing_function = normalize_zscore
67
68 # Data Mining
69 from tensorflow.keras.models import Sequential
70 from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
71
72 # Inisialisasi model
73 LeafNet = Sequential()
74
75 # Convolutional Layer 1
76 LeafNet.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(168, 244, 3)))
77 LeafNet.add(MaxPooling2D(pool_size=(2, 2)))
78
79 # Convolutional Layer 2
80 LeafNet.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
81 LeafNet.add(MaxPooling2D(pool_size=(2, 2)))
82
83 # Convolutional Layer 3
84 LeafNet.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
85
86 # Convolutional Layer 4
87 LeafNet.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
88
89 # Convolutional Layer 5
90 LeafNet.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
91 LeafNet.add(MaxPooling2D(pool_size=(2, 2)))
92
93 # Flatten Layer
94 LeafNet.add(Flatten())
95
96 # Fully Connected Layer 1
97 LeafNet.add(Dense(512, activation='relu'))
98 LeafNet.add(Dropout(0.5))
99
100 # Fully Connected Layer 2
101 LeafNet.add(Dense(256, activation='relu'))
102 LeafNet.add(Dropout(0.5))
103
104 # Fully Connected Layer 3
105 LeafNet.add(Dense(128, activation='relu'))
106 LeafNet.add(Dropout(0.5))
107
108 # Output Layer (Softmax)
109 LeafNet.add(Dense(4, activation='softmax'))
110
111 # Compile model dengan optimizer terbaik
112 LeafNet.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
113
114 # Tampilkan ringkasan model
115 LeafNet.summary()

```

Kode *Python* yang lengkap dapat mengakses pada *link* atau *QR Code* berikut:



<https://bit.ly/CNN-klasifikasi-pucuk-daun-teh>

Lampiran 6. Proses *Data Prerocessing*

```
○ ○ ○  
  
1 train = datagen.flow_from_directory(  
2     data_train,  
3     target_size = (168, 244),  
4     color_mode = 'rgb',  
5     class_mode = 'categorical',  
6     batch_size = 30  
7 )  
8  
9 valid = datagen.flow_from_directory(  
10    data_valid,  
11    target_size = (168, 244),  
12    color_mode = 'rgb',  
13    class_mode = 'categorical',  
14    batch_size = 44  
15 )  
16  
17 test = datagen.flow_from_directory(  
18    data_test,  
19    target_size = (168, 244),  
20    color_mode = 'rgb',  
21    class_mode = 'categorical',  
22    batch_size = 44,  
23    shuffle=False  
24
```

Lampiran 7. Tabel Prediksi

Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
LeafNet (70:15:15)	0_blm_matang	0	2
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0

Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	0
		1	1
		1	1
		1	1
		1	1
		1	0
		1	1
		1	1
		1	0
		1	0
		1	1
		1	2
		1	1
		1	1
		1	1
	2_matang	2	2
		2	2
		2	2
		2	1
		2	1
		2	2
		2	2
		2	3
		2	2
		2	3
2		3	
2		2	
2		2	
3_tua	3	3	
	3	3	
	3	3	
	3	3	

Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
		3	3
		3	3
		3	3
		3	0
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
LeafNet (80:10:10)	0_blm_matang	0	2
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	1
		1	1
		1	0
		1	2
		1	0
		1	1
		1	1
		1	1
		1	1
	2_matang	2	2
		2	2
		2	2
		2	2
		2	2
		2	3
		2	2
		2	3
2		2	

Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
		2	2
		2	2
	3_tua	3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
	3	2	
LeafNet (90:5:5)	0_blm_matang	0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	1
		1	1
		1	1
		1	1
		1	1
	2_matang	2	2
		2	2
		2	2
		2	2
		2	2
	3_tua	3	3
		3	3
3		3	
3		3	
3		2	
VGGNet11 (70:15:15)	0_blm_matang	0	2
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0

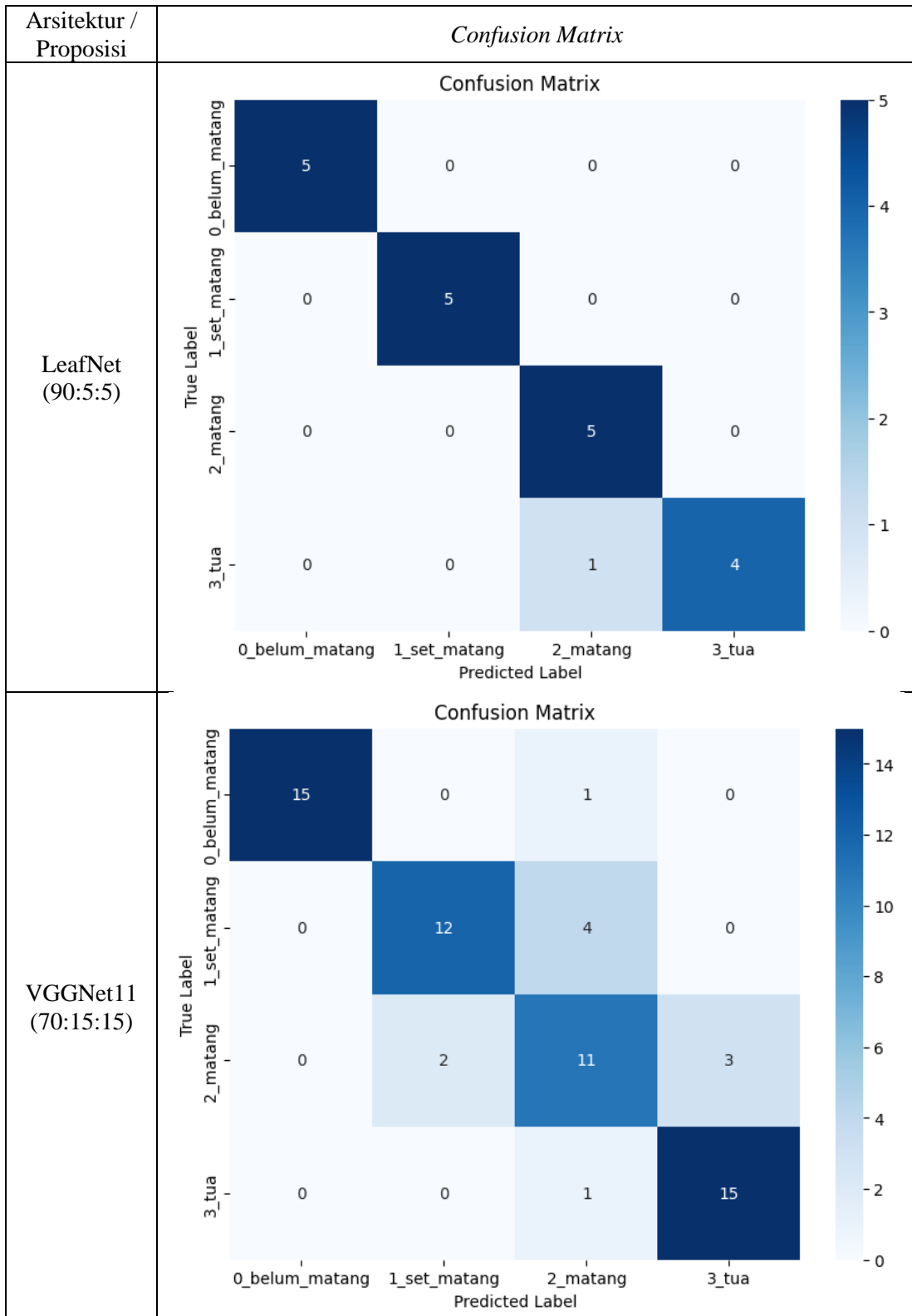
Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	1
		1	1
		1	1
		1	2
		1	2
		1	1
		1	1
		1	2
		1	1
		1	1
		1	1
		1	2
		1	1
		1	1
		1	1
	2_matang	2	2
		2	1
		2	2
		2	2
		2	2
		2	1
		2	2
		2	2
		2	3
		2	3
		2	2
		2	3
2		2	
2	2		
3_tua	3	3	
	3	3	
	3	3	
	3	3	
	3	3	

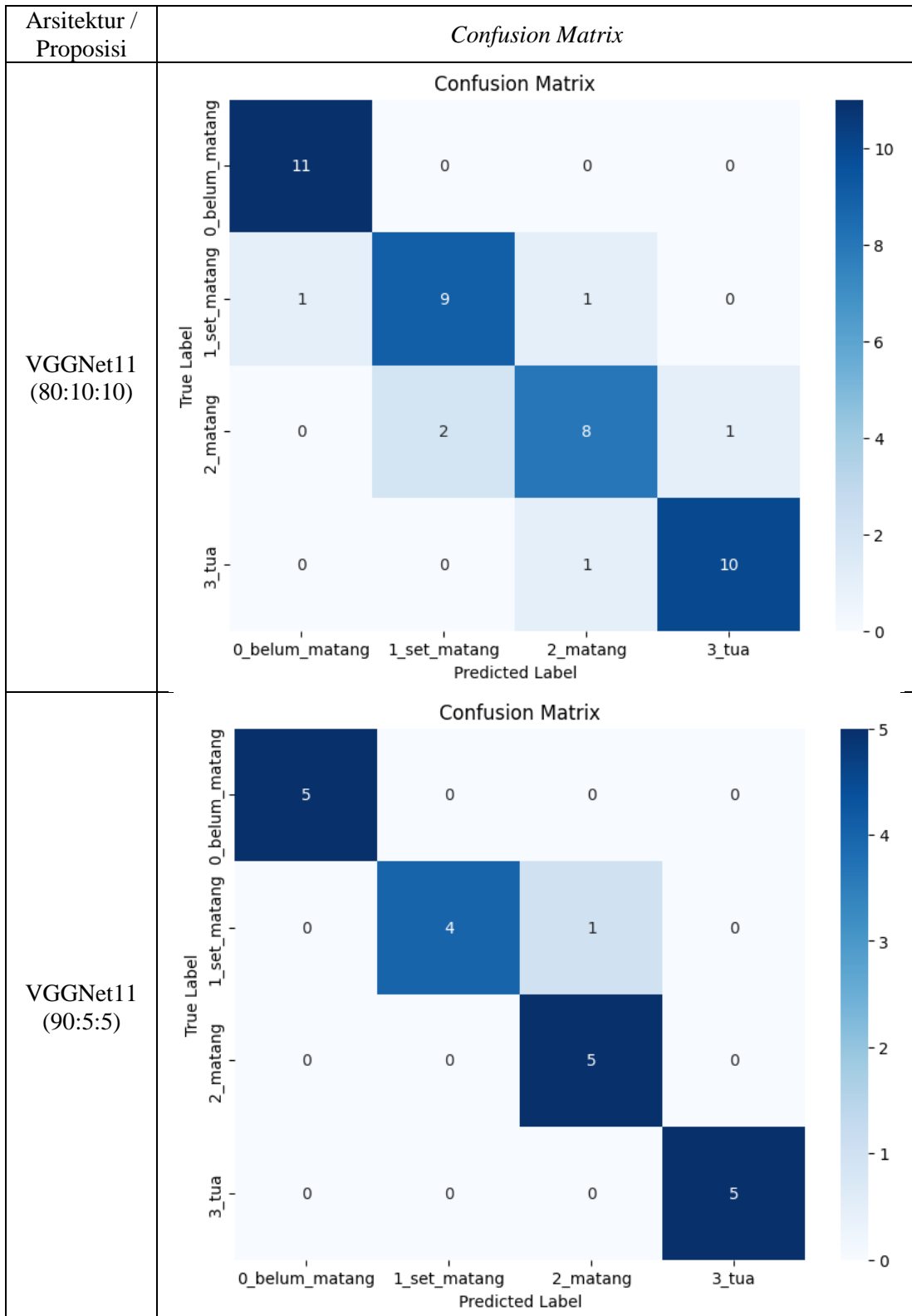
Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
		3	3
		3	3
		3	3
		3	3
		3	2
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
VGGNet11 (80:10:10)	0_blm_matang	0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	1
		1	1
		1	1
		1	2
		1	0
		1	1
		1	1
		1	1
		1	1
	2_matang	2	2
		2	2
		2	2
		2	3
		2	2
		2	2
		2	2
		2	1
2		2	

Arsitektur (proporsi)	Kelas	<i>True label</i>	<i>predicted label</i>
	3_tua	2	1
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	3
		3	2
VGGNet11 (90:5:5)	0_blm_matang	0	0
		0	0
		0	0
		0	0
		0	0
	1_set_matang	1	1
		1	1
		1	1
		1	2
	2_matang	1	1
		2	2
		2	2
		2	2
		2	2
	3_tua	2	2
		3	3
		3	3
3		3	
3		3	

Lampiran 8. Confusion Matrix

Arsitektur / Proposisi	<i>Confusion Matrix</i>																									
LeafNet (70:15:15)	<p style="text-align: center;">Confusion Matrix</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>True Label \ Predicted Label</th> <th>0_belum_matang</th> <th>1_set_matang</th> <th>2_matang</th> <th>3_tua</th> </tr> </thead> <tbody> <tr> <th>0_belum_matang</th> <td>15</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <th>1_set_matang</th> <td>5</td> <td>10</td> <td>1</td> <td>0</td> </tr> <tr> <th>2_matang</th> <td>0</td> <td>2</td> <td>11</td> <td>3</td> </tr> <tr> <th>3_tua</th> <td>1</td> <td>0</td> <td>0</td> <td>15</td> </tr> </tbody> </table>	True Label \ Predicted Label	0_belum_matang	1_set_matang	2_matang	3_tua	0_belum_matang	15	0	1	0	1_set_matang	5	10	1	0	2_matang	0	2	11	3	3_tua	1	0	0	15
True Label \ Predicted Label	0_belum_matang	1_set_matang	2_matang	3_tua																						
0_belum_matang	15	0	1	0																						
1_set_matang	5	10	1	0																						
2_matang	0	2	11	3																						
3_tua	1	0	0	15																						
LeafNet (80:10:10)	<p style="text-align: center;">Confusion Matrix</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>True Label \ Predicted Label</th> <th>0_belum_matang</th> <th>1_set_matang</th> <th>2_matang</th> <th>3_tua</th> </tr> </thead> <tbody> <tr> <th>0_belum_matang</th> <td>10</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <th>1_set_matang</th> <td>2</td> <td>8</td> <td>1</td> <td>0</td> </tr> <tr> <th>2_matang</th> <td>0</td> <td>0</td> <td>9</td> <td>2</td> </tr> <tr> <th>3_tua</th> <td>0</td> <td>0</td> <td>1</td> <td>10</td> </tr> </tbody> </table>	True Label \ Predicted Label	0_belum_matang	1_set_matang	2_matang	3_tua	0_belum_matang	10	0	1	0	1_set_matang	2	8	1	0	2_matang	0	0	9	2	3_tua	0	0	1	10
True Label \ Predicted Label	0_belum_matang	1_set_matang	2_matang	3_tua																						
0_belum_matang	10	0	1	0																						
1_set_matang	2	8	1	0																						
2_matang	0	0	9	2																						
3_tua	0	0	1	10																						





Lampiran 9. PTPN VIII Gunung Mas

