

**KLASIFIKASI KONDISI LANGIT MENGGUNAKAN
ALGORITMA *CONVOLUTIONAL NEURAL*
NETWORK DENGAN ARSITEKTUR
*NASNET***

SKRIPSI

**WANDA KHALISHAH
191402076**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

**KLASIFIKASI KONDISI LANGIT MENGGUNAKAN
ALGORITMA *CONVOLUTIONAL NEURAL*
NETWORK DENGAN ARSITEKTUR
*NASNET***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

**WANDA KHALISHAH
191402076**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2025**

PERSETUJUAN

Judul : KLASIFIKASI KONDISI LANGIT
MENGUNAKAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK
DENGAN ARSITEKTUR NASNET

Kategori : Skripsi

Nama : Wanda Khalishah

Nomor Induk Mahasiswa : 191402076

Program Studi : S-1 Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi

Medan, 10 Januari 2025

Komisi Pembimbing :

Pembimbing 1

Pembimbing 2

Prof. Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP. 198603032010121004

Dedy Arisandi ST., M.Kom
NIP. 197908312009121002

Diketahui/disetujui oleh

Program Studi S1- Teknologi Informasi

Ketua,

Dedy Arisandi ST., M.Kom
NIP.197908312009121002

PERNYATAAN**KLASIFIKASI KONDISI LANGIT MENGGUNAKAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK DENGAN
ARSITEKTUR NASNET****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Januari 2025

Wanda Khalishah

NIM. 191402076

ABSTRAK

Penelitian ini berfokus pada pengembangan sistem klasifikasi citra langit berbasis *Convolutional Neural Network* (CNN) dengan arsitektur *NASNet*, yang dikenal unggul dalam tugas klasifikasi gambar. Citra langit memiliki potensi besar untuk mendukung identifikasi lokasi optimal bagi penyerapan radiasi, dengan salah satu faktornya adalah cakupan awan di langit. Dalam studi ini, digunakan dataset *Singapore Whole sky Imaging CATegories* (SWIMCAT) yang terdiri dari lima kategori: langit cerah (*clear sky*), pola awan (*pattern cloud*), awan tebal gelap (*thick dark cloud*), awan tebal (*thick cloud*), dan kabut tipis (*veil*). Dataset ini menjadi dasar untuk melatih dan menguji model *NASNet* yang dipadukan dengan teknik *on-the-fly data augmentation* guna meningkatkan kinerja. Hasil eksperimen menunjukkan bahwa model yang dihasilkan mampu mencapai akurasi tinggi hingga 99,37%. Temuan ini tidak hanya mempertegas keandalan arsitektur *NASNet* dalam klasifikasi citra, tetapi juga membuka peluang besar untuk aplikasi lebih luas, seperti pemantauan cuaca, pengelolaan sumber daya energi berbasis matahari, dan pemrosesan gambar otomatis. Dengan akurasi tinggi dan efisiensi pemrosesan, pendekatan ini membuktikan relevansinya dalam mendukung pengembangan teknologi berbasis kecerdasan buatan di berbagai sektor.

Kata Kunci: Kondisi langit, Klasifikasi, *CNN*, *Transfer-Learning*, *NASNet*

ABSTRACT

This research focuses on developing a sky image classification system using *Convolutional Neural Network* (CNN) with the NASNet architecture, renowned for its superior performance in image classification tasks. Sky images hold significant potential for identifying optimal locations for solar radiation absorption, with cloud coverage being one of the key influencing factors. In this study, the *Singapore Whole Sky Imaging CATEgories* (SWIMCAT) dataset is utilized, comprising five categories: clear sky, pattern cloud, thick dark cloud, thick cloud, and veil. This dataset serves as the foundation for training and testing the NASNet model, enhanced with *on-the-fly data augmentation* techniques to boost its performance. Experimental results demonstrate that the proposed model achieves an outstanding accuracy of up to 99.37%. These findings highlight not only the reliability of NASNet architecture in image classification but also its potential for broader applications, including weather monitoring, solar energy management, and automated image processing. With its high accuracy and processing efficiency, this approach underscores its relevance in advancing AI-based technologies across diverse fields.

Keyword : Sky Condition, Classification, CNN, Transfer-learning, NASNet

KATA PENGANTAR

Puji serta syukur saya kepada Allah SWT, karena atas berkat rahmat-Nya, saya berhasil menyelesaikan penyusunan skripsi ini sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi S1 Teknologi Informasi, Universitas Sumatera Utara.

Saya ingin menyampaikan rasa terima kasih kepada semua pihak yang telah terlibat dalam perjalanan kuliah dan pengerjaan skripsi ini. Saya sangat berterima kasih kepada keluarga saya, terutama ibu saya yang selalu memberikan dukungan dan doa mereka selama saya menulis skripsi ini. Juga, kepada saudara saya yang tak henti-hentinya memberikan semangat hingga skripsi ini selesai.

Saya juga ingin mengucapkan terima kasih yang kepada pihak lainnya yang turut berperan dalam penyelesaian skripsi ini, yaitu:

1. Kepada kedua orang tua saya, terutama mama saya tercinta, Napsiah, yang dengan kasih sayang tanpa batas, doa-doa yang tidak pernah putus, serta dukungan yang tulus mengalir di setiap langkah saya. Skripsi ini adalah wujud kecil dari rasa terima kasih saya yang tak akan pernah cukup membalas semua cinta dan perjuangan mama.
2. Ibu Dr. Maya Silvi Lidya B.Sc., M.Sc selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Bapak Prof. Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.. selaku Dosen Pembimbing I yang telah mengarahkan peneliti dari penulisan judul hingga tahap akhir skripsi ini diselesaikan.
4. Bapak Dedy Arisandi ST., M.Kom selaku Dosen Pembimbing II sekaligus ketua program studi teknologi informasi yang telah membimbing peneliti dalam penelitian dan penulisan skripsi ini.
5. Semua Dosen Program Studi Teknologi Informasi yang telah memberikan ilmu saat penulis selama masa perkuliahan dari semester 1 hingga tahap akhir ini.
6. Staff dan pegawai Fasilkom-TI Universitas Sumatera Utara yang telah membantu segala urusan administrasi di masa perkuliahan.
7. Teman penulis dalam mengerjakan skripsi Kayla Alysa Adra yang

menemani dan memberikan dukungan.

8. Teman dari grup *WWUNO & Friends* dan semua mahasiswa Teknologi Informasi angkatan 2019 yang menemani perjalanan perkuliahan penulis dan memberikan dukungan selama ini.
9. Fathu Silmi, yang dengan penuh kesabaran dan cinta selalu menjadi sumber semangat saya, tak pernah lelah memberikan dukungan tulus, mengingatkan dengan lembut di saat saya mulai kehilangan arah, dan selalu percaya bahwa saya mampu menyelesaikan skripsi ini hingga tuntas. Kehadirannya adalah kekuatan besar yang membuat saya terus melangkah.
10. Dan semua pihak lain yang membantu penulis pada masa perkuliahan maupun saat pengerjaan skripsi yang tidak dapat penulis tulis kan satu persatu.

Semoga berkah dan anugerah dari Allah SWT mengalir kepada semua pihak yang telah memberikan bantuan, perhatian, dan dukungan kepada penulis, baik secara langsung maupun tidak langsung.

Medan, 10 Januari 2025

WANDA KHALISHAH

DAFTAR ISI	
PERSETUJUAN	2
PERNYATAAN	3
ABSTRAK	4
<i>ABSTRACT</i>	5
KATA PENGANTAR	6
DAFTAR ISI	8
DAFTAR GAMBAR	10
DAFTAR TABEL	
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1. Langit	7
2.2. Awan	7
2.3. Korelasi ketebalan optik awan dengan radiasi matahari	12
2.4. <i>Deep Learning</i>	12
2.5. <i>Convolutional Neural Network (CNN)</i>	13
2.6. <i>Transfer Learning</i>	15
2.7. <i>NASNet Mobile (Neural Architecture Search)</i>	17
2.8. <i>Preprocessing Image</i>	18
2.8.1. <i>Resize</i>	18
2.8.2. <i>Normalisasi</i>	19
2.8.3. <i>Zoom, rotate dan shift</i>	19

2.8.4. <i>Fill</i>	21
2.9. <i>On the fly data augmentation</i>	21
2.10. <i>Flask</i>	22
2.11. <i>Confusion Matrix</i>	23
2.12. <i>Penelitian Terdahulu</i>	26
2.13. Perbedaan penelitian	30
BAB III ANALISIS DAN PERANCANGAN SISTEM	33
3.1. Data Penelitian	33
3.2. Analisis Sistem	34
3.3. Perancangan antarmuka sistem	41
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	47
4.1. Implementasi Sistem	47
4.2. Implementasi Data	47
4.3. <i>Flowchart website</i>	48
4.4. Implementasi Antarmuka	49
4.5. Pelatihan Sistem	56
4.6. Pengujian Sistem	60
4.7. Diskusi	68
BAB V KESIMPULAN DAN SARAN	71
5.1. Kesimpulan	71
5.2. Saran	71
DAFTAR PUSTAKA	72

DAFTAR GAMBAR

Gambar 2. 1 Awan Cumulus (Putih Tebal)	8
Gambar 2. 2 Awan Cirrus (Selimut)	9
Gambar 2. 3 Awan Cirrocumulus (berpola)	9
Gambar 2. 4 Langit cerah	10
Gambar 2. 5 Awan Cumulonimbus (Gelap)	11
Gambar 2. 6 Radiasi matahari dan ketebalan optik awan.	12
Gambar 2. 7 <i>machine learning</i> dan <i>deep learning</i>	13
Gambar 2. 8 Arsitektur Umum CNN	14
Gambar 2. 9 Cara Kerja Max Pooling Layer	15
Gambar 2. 10 <i>machine learning</i> dan <i>transfer learning</i>	16
Gambar 2. 11 Arsitektur <i>Normal Cell</i> dan <i>Reduction Cell</i>	17
Gambar 2. 12 <i>On the fly data augmentations</i>	22
Gambar 2. 13 Contoh <i>confusion matrix</i>	23
Gambar 3. 1 Arsitektur umum sistem klasifikasi dengan NASNet	35
Gambar 3. 2 <i>Summary mode</i>	39
Gambar 3. 3 Rancangan antarmuka <i>landing page</i> web klasifikasi	42
Gambar 3. 4 Rancangan antarmuka penjelasan kategori	43
Gambar 3. 5 Rancangan antarmuka laman <i>about</i>	44
Gambar 3. 6 Rancangan antarmuka laman input gambar	44
Gambar 3. 7 Rancangan antarmuka hasil klasifikasi	45
Gambar 4. 1 <i>Flowchart</i> sistem	48
Gambar 4. 2 Hasil tampilan <i>landing page</i>	50
Gambar 4. 3 Hasil tampilan antarmuka kategori citra langit	50
Gambar 4. 4 Hasil tampilan antarmuka halaman about	51
Gambar 4. 5 Hasil tampilan halaman <i>upload</i> citra	52
Gambar 4. 6 Hasil halaman klasifikasi	52
Gambar 4. 7 Halaman <i>upload batch data</i>	53
Gambar 4. 8 Tampilan progres klasifikasi	54
Gambar 4. 9 Tampilan hasil klasifikasi <i>batch data</i>	55
Gambar 4. 10 Hasil file pdf dari klasifikasi <i>batch data</i>	55

Gambar 4. 11 Grafik <i>loss</i> dan <i>acc</i> pada training tanpa augmentasi	56
Gambar 4. 12 Grafik <i>loss</i> dan <i>acc</i> training setelah data augmentasi	59
Gambar 4. 13 <i>Confusion matrix</i> klasifikasi menggunakan <i>NASNet mobile</i>	65
Gambar 4. 14 Perbandingan <i>confussion matrix</i>	69

DAFTAR TABEL

Tabel 2. 1 Penelitian terdahulu	26
Tabel 3. 1 Rasio jumlah dataset SWIMCAT	34
Tabel 3. 2 Rasio perbandingan pembagian data	36
Tabel 4. 1 Spesifikasi perangkat keras	47
Tabel 4. 2 Spesifikasi perangkat lunak	47
Tabel 4. 3 Hasil pelatihan menggunakan arsitektur NASNet	58
Tabel 4. 4 Uji klasifikasi	60
Tabel 4. 5 Jumlah <i>TP</i> , <i>FP</i> dan <i>FN</i>	66

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kondisi langit memiliki berperan penting dalam keberlangsungan aktivitas manusia sehari-hari dan menjadi faktor yang dipertimbangkan untuk berbagai keputusan (Naufal & Kusuma, 2022) . Kondisi ini dapat diamati dari berbagai unsur, diantaranya adalah dengan mengamati langit dan awan. Awan adalah kumpulan tetesan air atau partikel es yang tersuspensi di atmosfer dan menutupi lebih dari 50% permukaan bumi (Huang et al., 2018). Awan menjadi bagian utama dalam siklus hidrologi, perubahan iklim, dan radiasi bumi (Dagan et al., 2018). Distribusi temporal dan spasial dari berbagai jenis awan dapat membedakan kondisi iklim dan cuaca. Misalnya awan cumulonimbus yang diasosiasikan sebagai ketidakstabilan atmosfer, turbulensi, dan badai petir (Zhang et al., 2019). Oleh sebab itu, memahami awan sangat penting untuk berbagai aplikasi yang berkaitan dengan siklus hidrologi, iklim, dan atmosfer.

Awan berpengaruh tinggi terhadap paparan radiasi bumi, dengan memantulkan radiasi matahari yang masuk dan menjebak radiasi yang keluar (Yabuki et al., 2014). Kumpulan awan menyebabkan banyak perubahan pada radiasi matahari yang diterima dan dapat menurunkan daya pembangkit listrik panel PV. Radiasi matahari, suhu dan data meteorologi lainnya juga mempengaruhi pembangkit listrik. Sehingga segmentasi awan disarankan sebagai salah satu sumber informasi terpenting untuk pengoperasian pembangkit listrik tenaga surya (Moughyt et al., 2015).

Dengan meningkatnya minat pada sumber energi yang dapat diperbaharui (terutama energi matahari), informasi tentang cakupan awan semakin penting untuk perkiraan produksi listrik dari sistem tenaga surya (Taravat et al., 2015). Seperti yang disampaikan pada penelitian yang membahas tentang korelasi tutupan awan dengan radiasi matahari pada tahun 2016, menyimpulkan bahwa pengurangan tutupan awan sangat mempengaruhi jumlah radiasi matahari (Yuan et al., n.d.) Sehingga analisa tutupan awan menjadi penting untuk pengembangan pembangkit listrik tenaga surya.

Data cakupan awan dapat diekstrak dari citra satelit. Namun, terdapat kelemahan utama dari citra satelit yaitu bidang cakupannya yang luas sehingga tidak dapat menghasilkan statistik lokal yang spesifik (Sun et al., 2011). Berbeda dengan citra satelit, citra langit dari darat dapat memberikan pengenalan cuaca yang lebih spesifik di suatu wilayah terbatas. Citra langit ini dapat digunakan untuk menyelesaikan banyak

studi tentang potensi pembangkit tenaga surya lokal (Moughyt et al., 2015). Dibandingkan dengan cara konvensional seperti menggunakan kemampuan visual manusia yang terbatas.

Klasifikasi citra menjadi salah satu bidang teknologi terkini yang dapat menggantikan kemampuan visual manusia (Javidi, 2002). Dengan menggunakan klasifikasi citra, komputer dapat melihat cuaca hanya berdasarkan citra langit secara real time. Sistem klasifikasi citra cuaca dapat membantu dalam pengembangan mesin otomatis seperti pembangkit listrik tenaga surya yang dapat membuat keputusan mandiri berdasarkan kondisi cuaca yang dideteksi.

Pada tahun 2022 terdapat penelitian berjudul “*Intelligent Mobile Plant Disease Diagnostic System Using NASNet-Mobile Deep Learning*” yang melakukan penelitian untuk mengembangkan sistem diagnostik penyakit tanaman yang dapat diakses melalui perangkat mobile dengan menggunakan *NASNet mobile deep learning*. Model dilatih menggunakan PlantVillage dataset, yang terdiri dari 54,309 gambar tanaman sehat dan terinfeksi. Sistem yang dikembangkan mencapai akurasi sebesar 99.31% dalam mendeteksi penyakit pada daun tanaman. Ini menunjukkan bahwa *NASNet Mobile* efektif dalam klasifikasi gambar dengan tingkat kesalahan yang sangat rendah (Adedoya et al., 2022). Oleh karena itu pada penelitian ini akan dilakukan uji metode *NASNet Mobile* pada pengelompokan citra langit untuk memperoleh hasil akurasi yang baik.

1.2. Rumusan Masalah

Pembangkit Listrik Tenaga Surya (PLTS) bergantung pada kondisi langit untuk mengoptimalkan penangkapan radiasi matahari. Langit yang cerah memungkinkan tangkapan energi yang maksimal, sementara kondisi berawan atau mendung dapat menurunkan efisiensinya. Namun, proses identifikasi kondisi langit secara manual memiliki keterbatasan, seperti keterbatasan waktu pengamatan, risiko kesalahan akibat faktor subjektivitas, dan ketidakmampuan untuk memantau secara terus-menerus. Hal ini menjadi tantangan serius dalam memastikan pengelolaan energi surya yang efisien dan andal.

Namun, sistem otomatisasi klasifikasi kondisi langit yang ada saat ini masih menghadapi tantangan terkait metode dan akurasi klasifikasi, yang dapat terus dikembangkan dengan memanfaatkan metode klasifikasi gambar terbaru. Salah satu

pendekatan terkini yang menjanjikan adalah penggunaan model deep learning seperti *NASNet*. Model ini telah terbukti dalam beberapa penelitian mampu menghasilkan akurasi tinggi dalam klasifikasi gambar.

1.3. Tujuan Penelitian

Mengembangkan model klasifikasi kondisi langit yang dapat mengelompokkan kondisi langit berdasarkan gambar yang diambil oleh kamera. Tujuan utama dari penelitian ini adalah untuk meningkatkan akurasi dan efisiensi klasifikasi citra langit menggunakan implementasi algoritma *CNN* dengan arsitektur *NASNet*. Dengan menggunakan model ini, diharapkan dapat membantu dalam mengklasifikasi kondisi langit dan memberikan informasi penting yang berguna untuk kegiatan yang terkait dengan pemanfaatan energi surya.

1.4. Batasan Penelitian

Beberapa batasan penelitian klasifikasi citra langit menggunakan algoritma *CNN* dengan arsitektur *NASNet* antara lain:

1. Data yang digunakan dalam penelitian ini hanya terbatas pada data *Singapore Whole sky IMaging CATegories Database (SWIMCAT)*.
2. Penelitian ini hanya memperhitungkan lima kelas, yaitu langit cerah, awan berpola, awan gelap tebal, awan putih tebal, dan awan tipis .
3. Pengujian dilakukan hanya pada level pengenalan objek dalam satu frame saja, sehingga tidak memperhitungkan perubahan kondisi langit dalam waktu yang berkelanjutan atau video yang terdiri dari beberapa frame.
4. Penelitian ini dilakukan dengan menggunakan arsitektur *NASNet* versi *open source*. Oleh karena itu, penelitian ini tidak mengeksplorasi kemungkinan pengembangan arsitektur *NASNet* yang lebih lanjut atau penggunaan arsitektur deteksi objek yang berbeda.

1.5. Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat potensial, di antaranya:

1. Penelitian ini dapat membuka pengembangan teknologi baru yang dapat digunakan untuk mencari lokasi strategis yang mampu menghasilkan radiasi matahari lebih maksimal untuk pengembangan pembangkit listrik tenaga surya.

2. Penelitian ini dapat menguji efektifitas arsitektur *NASNet* dalam membantu meningkatkan akurasi klasifikasi kondisi langit, yang dapat meningkatkan kinerja sistem dalam berbagai sektor.
3. Dengan menggunakan metode *CNN*, penelitian ini dapat membantu mengurangi biaya dan waktu yang diperlukan untuk klasifikasi, karena metode ini dapat mempercepat proses klasifikasi.

1.6. Metodologi Penelitian

Berikut merupakan arsitektur umum serta gambaran metodologi penelitian pada klasifikasi citra langit menggunakan algoritma *CNN* dengan arsitektur *NASNET*:

1.6.1 Studi literatur

Studi literatur merupakan proses terdahulu dalam penelitian ini dimana pada tahap ini setiap informasi yang dibutuhkan akan dikumpulkan. Informasi yang diperoleh dapat berupa jurnal, informasi dari ahli, serta informasi mengenai kondisi langit dan arsitektur *NASNET*.

1.6.2 Analisis permasalahan

Analisis permasalahan adalah tahapan menentukan masalah dan batasan yang akan diselesaikan oleh penelitian ini. Dengan menganalisis permasalahan, penelitian ini akan berjalan sesuai dengan kebutuhan dan mampu mencapai tujuan penelitian.

1.6.3 Perancangan

Pada tahap ini , arsitektur umum sistem akan dirancang untuk menggambarkan alur program yang akan dibangun. Rancangan arsitektur ini mampu memudahkan dalam proses pengembangan sistem agar lebih tertata dan sesuai dengan tujuan penelitian.

1.6.4 Implementasi

Pada tahapan implementasi ini, model *CNN* dengan arsitektur *NASNET* yang memiliki hasil akurasi tertinggi akan di implementasikan guna mendapatkan hasil dari penelitian yang dibuat.

1.6.5 Pengujian program

Tahap ini dilakukan untuk menguji sistem yang telah dibangun sehingga memastikan program yang sudah dibuat dapat digunakan dan bekerja secara baik.

1.6.6 Penyusunan Laporan

Tahapan akhir dari penelitian ini adalah menyusun laporan, dimana proses penelitian yang telah berjalan didokumentasikan dalam laporan oleh peneliti disertai dengan lampiran proses sehingga menampilkan hasil akhir pada penelitian.

1.7. Sistematika Penulisan

Pada Penelitian ini, sistematika penulisan terdiri atas lima bagian, yaitu:

BAB I: PENDAHULUAN

Pada bab ini peneliti menguraikan dan merinci aspek-aspek penting yang mendasari pelaksanaan penelitian ini. Bab ini mencakup beberapa subbab, termasuk latar belakang penelitian, rumusan masalah yang dihadapi dan menjadi dasar penelitian, tujuan penelitian, batasan-batasan yang ditetapkan, manfaat penelitian, metodologi yang digunakan, dan sistematika penulisan.

BAB II: LANDASAN TEORI

Pada bab ini, peneliti menguraikan teori-teori yang relevan untuk menjelaskan penelitian yang dilakukan. Teori-teori yang dibahas mencakup berbagai topik yang terkait dengan objek dan algoritma yang digunakan dalam penelitian, seperti langit, awan, *deep learning*, *convolutional neural network (CNN)*, *Arsitektur NASNET*. Selain itu, juga akan dibahas penelitian-penelitian terdahulu yang relevan.

BAB III: ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini, peneliti menjelaskan dan menguraikan sistem yang dikembangkan dalam penelitian ini. Aspek-aspek yang mendukung penelitian meliputi data yang digunakan, arsitektur umum sistem, proses preprocessing, pembangunan sistem, dan desain antarmuka sistem penelitian.

BAB IV: IMPLEMENTASI DAN PENGUJIAN SISTEM

Dalam bab ini, peneliti menjelaskan hasil dari implementasi dan pengujian sistem yang telah dirancang. Hasil tersebut merupakan kelanjutan dari proses yang telah dibahas secara rinci pada bab sebelumnya. Peneliti juga menganalisis kinerja sistem

untuk memastikan bahwa sistem berjalan sesuai dengan tujuan yang diharapkan.

BAB V: KESIMPULAN DAN SARAN

Dalam bab ini, peneliti menyampaikan kesimpulan yang diperoleh dari penelitian yang telah dilakukan. Kesimpulan tersebut merangkum temuan-temuan penting dan pencapaian yang dicapai selama proses penelitian. Selain itu, peneliti juga memberikan saran yang dapat dijadikan pertimbangan atau masukan bagi penelitian selanjutnya, untuk memperbaiki atau mengembangkan aspek-aspek yang belum optimal.

BAB II

LANDASAN TEORI

2.1. Langit

Langit, yang meliputi bagian atas permukaan Bumi, merupakan lapisan tersendiri yang dikenal sebagai atmosfer. Langit tersusun dari berbagai lapisan gas dan udara, masing-masing dengan komposisi uniknya sendiri. Karena refraksi atmosfer terhadap cahaya biru dari sinar matahari lebih besar daripada cahaya merah, langit sering kali tampak biru pada pagi dan sore hari (Tyndall, 1868).

Salah satu faktor utama dalam prediksi cuaca pegunungan adalah kondisi langit, yang berkaitan dengan tren tekanan udara. Akumulasi awan yang lebih tebal menunjukkan kemungkinan akan terjadinya presipitasi. Di sisi lain, jika tidak ada awan di langit, itu bisa berarti cuaca cerah akan segera datang. Pada malam hari, halo dapat terbentuk di sekitar Bulan oleh awan cirrostratus tipis dan tinggi, yang menandakan datangnya arus udara hangat dan hujan. Kehadiran kabut di pagi hari merupakan tanda cuaca tenang dan langit cerah, yang mungkin disertai oleh lapisan laut. Ketika awan atau angin mencegah terbentuknya kabut, maka sudah hampir waktunya hujan. Badai petir merupakan tanda udara dingin di cakrawala (Marlon, 2015).

2.2. Awan

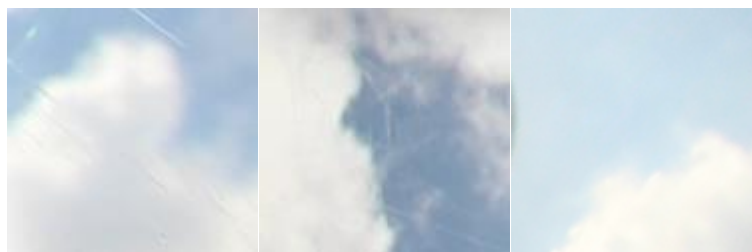
Awan adalah salah satu kekuatan terpenting dari keseimbangan radiasi Bumi dan siklus hidrologi, dan pada saat yang sama salah satunya yang paling sedikit dipahami. Diketahui bahwa awan rendah menyediakan umpan balik negatif dan tinggi, awan tipis umpan balik positif pada anggaran radiasi. Masih belum diketahui yang menyebabkan ketidakpastian iklim yang besar dan prediksi iklim (Solomon et al., 2007).

Efek awan pada radiasi matahari dan terestrial disebabkan refleksi dan penyerapan oleh partikel awan dan tergantung kuat pada volume, bentuk, ketebalan dan komposisi awan. Informasi awan berskala besar tersedia dari beberapa satelit, tetapi data tersebut disediakan dalam resolusi rendah dan mungkin mengandung kesalahan. Misalnya, awan kecil sering diabaikan karena bidang pandang radiometer terbatas. Contohnya, awan tipis yang rendah sering membuat kekeliruan karena kecerahan dan suhu yang serupa (Ricciardelli et al., 2008; Dybroe et al., 2005). Menyebabkan, radiasi

matahari yang mencapai permukaan sehubungan dengan jenis awan tidak bisa ditentukan, meskipun ini penting untuk studi radiasi awan.

Menurut *International Cloud Classification System (ICCS)* yang diterbitkan dalam WMO (1987) terdapat beberapa jenis awan yaitu diantaranya :

1. *Cumulus* (Awan putih tebal)

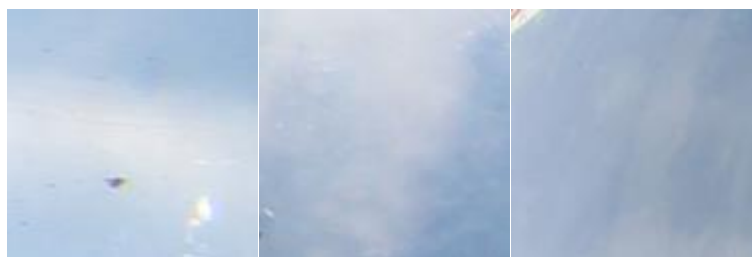


Gambar 2. 1 Awan Cumulus (Putih Tebal)

Awan *cumulus* adalah jenis awan rendah hingga menengah yang biasanya terbentuk pada ketinggian antara 500 hingga 2.000 meter di atas permukaan laut. Proses pembentukan awan ini terutama terjadi akibat naiknya udara hangat yang lembab, sehingga awan ini umum ditemukan di daerah tropis atau subtropis. Dalam interaksinya dengan radiasi matahari, awan cumulus dapat berperan penting dalam memengaruhi albedo atau jumlah radiasi yang dipantulkan kembali ke atmosfer, yang berimplikasi pada keseimbangan energi permukaan bumi. (Hinkelman et al., 2007)

Kehadiran awan *cumulus* menyebabkan variabilitas dalam radiasi matahari yang mencapai permukaan bumi karena efek penyebaran (scattering) dan refleksi pada sisi-sisi awan. Efek ini, sering disebut sebagai cloud enhancement, dapat meningkatkan radiasi matahari lokal saat sinar matahari langsung menembus sela-sela awan. Sebagai contoh, penelitian yang menggunakan model transfer radiasi tiga dimensi menemukan bahwa penyerapan dan refleksi radiasi oleh awan *cumulus* dapat bervariasi tergantung pada posisi matahari dan orientasi awan, terutama dalam kondisi cuaca yang tenang tanpa turbulensi besar. (Hinkelman et al., 2007)

2. *Cirrus & Cirrostratus*

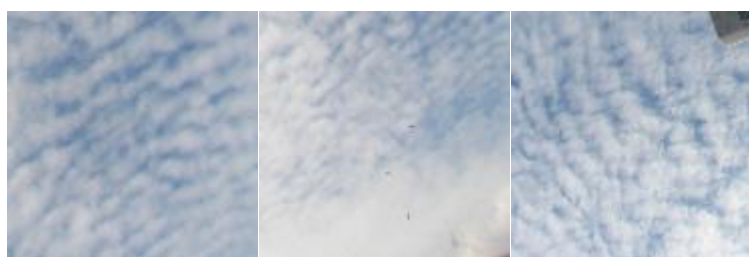


Gambar 2. 2 Awan Cirrus (Selimut)

Awan yang menyelimuti langit, dikenal sebagai awan *cirrus*, merupakan salah satu jenis awan yang memiliki karakteristik khusus dan terbentuk di ketinggian yang sangat tinggi di atmosfer. Awan *cirrus* memiliki penampilan yang tipis dan transparan, dengan serat-serat halus menyerupai kapas. Bentuknya sering terlihat seperti jaring laba-laba atau untaian panjang yang terpisah di langit. Awan ini terbentuk di ketinggian lebih dari 6.000 meter (20.000 kaki), biasanya di lapisan troposfer atas atau stratosfer bawah. (Heymsfield et al., 2017)

Awan *cirrus* tersusun dari kristal-kristal es yang terbentuk ketika uap air membeku di suhu yang sangat rendah. Karena terbentuk di kondisi udara yang sangat dingin dan kering, awan *cirrus* biasanya tidak membawa hujan dan sering dikaitkan dengan cuaca cerah dan stabil. Meskipun demikian, keberadaan awan *cirrus* juga bisa menjadi tanda perubahan cuaca yang akan datang, terutama jika mereka menebal atau berubah menjadi awan *cirrostratus* atau *cirrocumulus* yang lebih rendah.

3. Awan *Cirrocumulus*



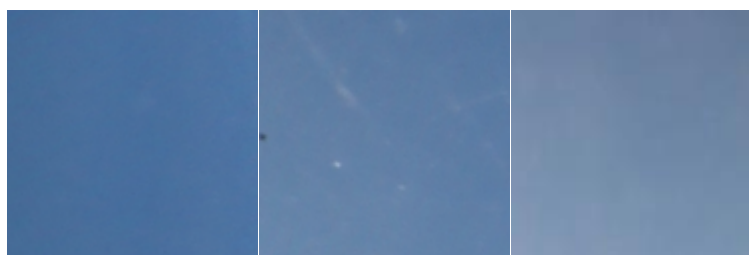
Gambar 2. 3 Awan Cirrocumulus (berpola)

Awan *cirrocumulus* adalah jenis awan tinggi yang terbentuk pada ketinggian antara 5.000 hingga 15.000 kaki (sekitar 1.500 hingga 5.000 meter) di atas permukaan laut. Awan ini terdiri dari patch-patch tipis yang memiliki karakteristik gumpalan, sering kali terlihat seperti butiran beras. *Cirrocumulus* merupakan satu-satunya awan di lapisan tinggi yang memiliki sifat gumpalan, dan meskipun tidak umum menghasilkan halo matahari, mereka dapat menunjukkan fenomena optik seperti iridescensi dan korona.

pada kesempatan tertentu. Keberadaan awan ini sering kali menandakan bahwa awan *cirrus* atau *cirrostratus* juga ada di dekatnya, dan mereka dapat menciptakan efek langit mackerel yang menyerupai sisik ikan, terutama saat matahari terbenam. (Goodson & Kopknit, n.d.)

Kaitan *cirrocumulus* dengan radiasi matahari sangat penting. Karena berada pada ketinggian tinggi, awan ini terpapar langsung oleh sinar matahari, yang memungkinkan mereka untuk memantulkan dan membiaskan radiasi tersebut. Hal ini dapat mempengaruhi jumlah radiasi matahari yang mencapai permukaan bumi; *cirrocumulus* dapat berfungsi sebagai reflektor radiasi, mengurangi pemanasan lokal dengan memantulkan sebagian sinar matahari kembali ke luar angkasa. Di sisi lain, mereka juga dapat membantu menjaga suhu permukaan dengan menangkap radiasi inframerah yang dipancarkan oleh bumi, berkontribusi pada efek rumah kaca. Dengan demikian, *cirrocumulus* memainkan peran penting dalam dinamika iklim dan interaksi antara awan dan radiasi matahari.

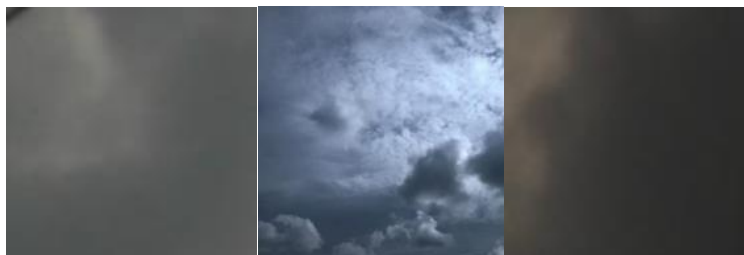
4. *Clear sky*



Gambar 2. 4 Langit cerah

Langit terlihat sangat cerah dengan kurang dari 10% area yang tertutupi awan, atau bahkan sama sekali tanpa awan. Hampir seluruh permukaan langit tampak bebas dari awan, menciptakan pemandangan yang sangat jernih. Kondisi ini sering menandakan cuaca yang stabil dan baik, tanpa tanda-tanda hujan atau gangguan atmosfer lainnya. Dalam kondisi ini, radiasi matahari dapat mencapai permukaan bumi secara langsung tanpa halangan. Radiasi langsung ini mempengaruhi suhu permukaan dan proses fotosintesis pada tanaman. Dan menunjukkan bahwa pada hari-hari cerah, intensitas radiasi matahari bisa mencapai maksimum sekitar 1000 W/m^2 di permukaan bumi. (Anggreni & Adriat, 2018)

5. *Cumulonimbus*



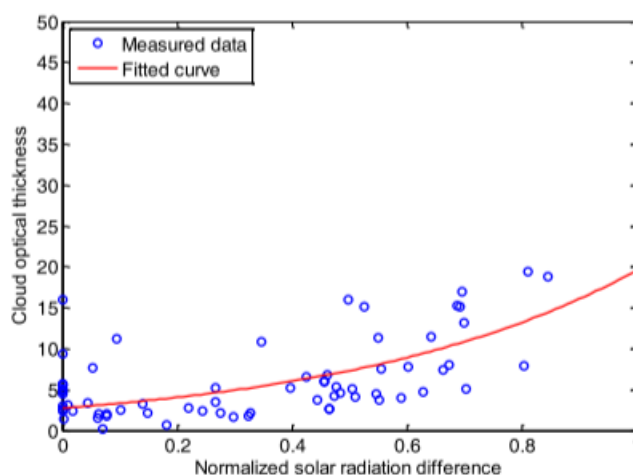
Gambar 2. 5 Awan Cumulonimbus (Gelap)

Awan *cumulonimbus* adalah jenis awan yang paling dikenal karena kemampuannya menciptakan cuaca buruk, termasuk hujan deras, angin kencang, hail, petir, dan guntur. *Cumulonimbus* memiliki tinggi yang sangat mengesankan, menjadi satu-satunya jenis awan yang dapat menjangkau semua tiga lapisan awan di atmosfer. Dasar awan ini bisa sangat rendah, sementara puncaknya dapat mencapai lapisan tertinggi troposfer. Puncak cumulonimbus sering kali memiliki penampilan botak (spesies awan *calvus*) atau tampak berbulu dan berserat (spesies awan *capillatus*). Awan cumulonimbus yang lebih kuat dapat menghasilkan berbagai fitur menarik, seperti awan anvil (fitur awan *incus*) yang terbentuk ketika awan menyentuh puncak troposfer dan menyebar di langit, serta contoh dramatis dari awan *mammatus* (fitur awan *mamma*) yang disebabkan oleh kantong udara dingin yang tenggelam. (Goodson & Kopknit, n.d.)

Kaitan *cumulonimbus* dengan radiasi matahari sangat signifikan. Karena ketinggiannya yang besar, cumulonimbus dapat mempengaruhi jumlah radiasi matahari yang mencapai permukaan bumi. Puncak awan ini dapat memantulkan radiasi matahari kembali ke luar angkasa, berpotensi mengurangi pemanasan lokal. Di sisi lain, cumulonimbus juga mampu menangkap radiasi inframerah yang dipancarkan oleh permukaan bumi, berkontribusi pada efek rumah kaca. Dengan demikian, cumulonimbus tidak hanya berperan dalam menciptakan cuaca ekstrem tetapi juga dalam mempengaruhi keseimbangan energi di atmosfer secara keseluruhan. Awan ini dapat diidentifikasi dengan fakta-fakta bahwa ketinggiannya berkisar antara 0,5 hingga 16 km (2.000 hingga 52.000 kaki), berasal dari istilah Latin "*cumulo*-" yang berarti tumpukan dan "*nimbo*-" yang berarti hujan, serta disingkat sebagai Cb.

2.3. Korelasi ketebalan optik awan dengan radiasi matahari

Berdasarkan penelitian oleh Yuan et al pada tahun 2016 berjudul “*Correlation between Cloud Optical Thickness and Solar Radiation*” yang dilakukan di singapura. Menunjukkan bahwa penurunan radiasi matahari meningkat dengan cepat seiring dengan peningkatan ketebalan optik awan. Perubahan ini dapat dilihat pada gambar 2.6.

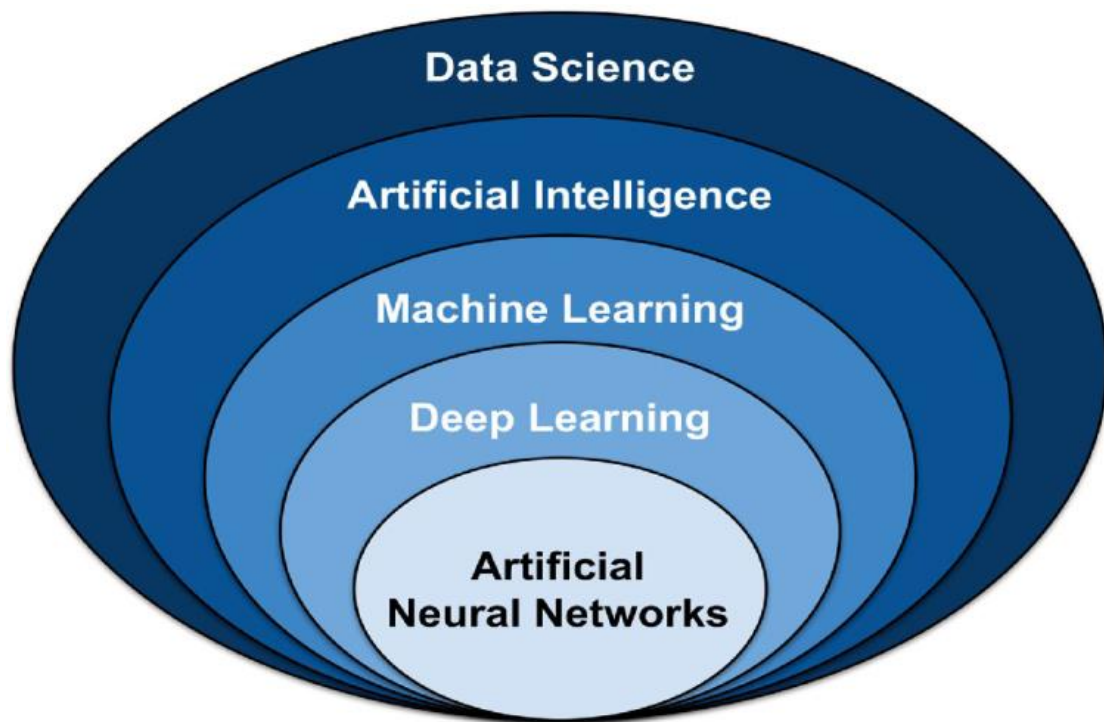


Gambar 2. 6 Radiasi matahari dan ketebalan optik awan. (Yuan et al., n.d.)

Dari hasil yang diperoleh, terlihat bahwa ketika ketebalan optik awan lebih besar dari 10, terdapat potensi yang cukup tinggi bahwa radiasi matahari yang tercatat di stasiun cuaca akan menurun hingga 60%. Selain itu, ketika ketebalan optik awan meningkat lebih dari 20, kurva yang dipasang menunjukkan bahwa tidak ada radiasi matahari yang akan terdeteksi oleh stasiun cuaca.(Yuan et al., n.d.)

2.4. Deep Learning

Deep Learning dengan banyak lapisan pemrosesan dapat mempelajari representasi data dengan berbagai tingkat abstraksi melalui pembelajaran mendalam. Dengan memanfaatkan algoritma backpropagation, yang menunjukkan kepada mesin cara memodifikasi parameter internal yang digunakan untuk menghitung representasi di setiap lapisan berikutnya berdasarkan representasi di lapisan sebelumnya, pembelajaran mendalam mampu mengungkap struktur rumit dalam kumpulan data besar (Lv et al., 2022).

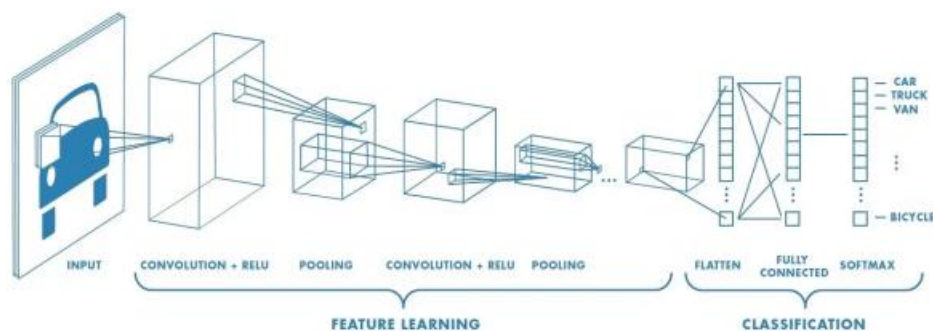


Gambar 2. 7 *machine learning* dan *deep learning* (Agarwal et al, 2024)

Pada gambar menunjukkan bahwa *deep learning* merupakan bagian lebih lanjut dari *machine learning* yang memproses data lebih kompleks. Dalam *Machine learning* masih menggunakan metode manual untuk melakukan ekstraksi fitur pada data, lalu dilanjutkan dengan tahapan klasifikasi. Sedangkan pada *deep learning*, *feature extraction* dilakukan secara otomatis yang dibangun di atas *artificial neural networks* (ANN) (Agarwal et al, 2024).

2.5. *Convolutional Neural Network (CNN)*

Convolution neural network (CNN) adalah jaringan struktur dalam model *deep learning* (Guo Z.M et al.). Berbeda dari pembelajaran mesin tradisional, *Convolutional neural network* bisa lebih baik digunakan untuk pemrosesan data gambar dan deret waktu, terutama untuk klasifikasi gambar dan pengenalan bahasa. Arsitektur CNN dapat dilihat pada gambar berikut.

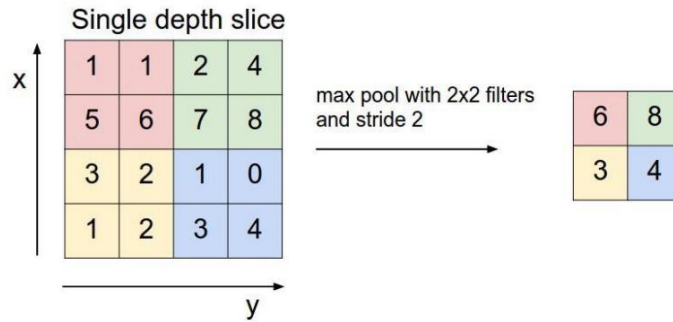


Gambar 2. 8 Arsitektur Umum CNN (Agarwal et al, 2024).

Convolution layer, *pooling layer*, dan *full connection* atau *fully connected layer* merupakan 3 layer yang terdapat pada CNN. Data masukan diterima oleh *input layer*, kemudian di proses dalam *convolution layer*, *pooling layer*, dan *fully connected layer* sebelum dikeluarkan oleh *output layer*. Ekstraksi fitur dilakukan oleh *convolution layer*, pemetaan fitur dilakukan oleh *pooling layer*, sedangkan node – node pada *fully connected layer* terhubung dengan *layer – layer* sebelumnya. (Lv et al., 2022). Menggunakan dua fungsi dengan argumen bernilai riil, lapisan *konvolusional* melakukan operasi. Dengan mengubah gambar input menjadi Peta Fitur, prosedur ini menerapkan output fungsi. Anda mungkin menganggap kedua argumen ini, input dan output, memiliki nilai riil. Ini adalah rumus yang menggambarkan operasi konvolusi dalam pengertian matematis murni.

Biasanya, lapisan penggabungan mengikuti lapisan *konvolusional*. Filter dengan ukuran dan langkah tetap yang bergerak di seluruh wilayah peta fitur adalah lapisan penggabungan secara teori. *Max Pooling* dan *Average Pooling* adalah metode penggabungan yang paling populer. Berbeda dengan *Average Pooling*, yang memilih nilai rata-rata, *Max Pooling* 2x2 33 dengan langkah 2 memilih nilai maksimum di wilayah piksel 2x2 untuk setiap pergeseran filter (Lv et al., 2022a).

Peta aktivasi dan peta fitur adalah nama umum untuk output yang keluar dari lapisan penggabungan.



Gambar 2. 9 Cara Kerja Max Pooling Layer (Dzulqarnain et al., 2019)

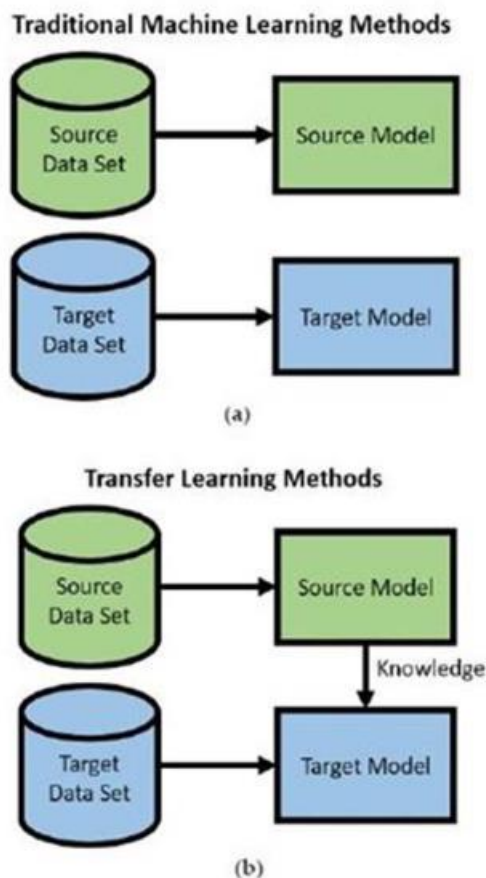
Untuk menghindari overfitting dan mempercepat komputasi, lapisan pooling digunakan untuk menurunkan sampel peta fitur, yang mengurangi jumlah parameter yang perlu diperbarui.

Di antara sekian banyak manfaat pembelajaran mendalam menggunakan teknik CNN adalah yang tercantum oleh Yu dan Shi (2018):

1. Selama pelatihan, pembelajaran representasi didukung oleh proses CNN hierarkis dan penyederhanaan fitur ini, yang memungkinkannya untuk secara otomatis beradaptasi dengan data dan tugas prediksi di bidang tertentu.
2. Peningkatan kinerja prediksi, kemampuan representasi model yang lebih besar, dan kesederhanaan penggunaan dalam membangun jaringan saraf yang lebih dalam semuanya merupakan hasil dari peningkatan daya pemrosesannya.
3. Pengguna tidak memerlukan pengetahuan domain tentang item data yang digunakan; metode pembelajaran CNN menyediakan pelatihan menyeluruh.

2.6. *Transfer Learning*

Transfer learning adalah teknik dalam machine learning yang memungkinkan model yang dilatih pada satu tugas untuk diadaptasi ke tugas lain yang terkait, dengan memanfaatkan pengetahuan yang diperoleh dari domain sumber untuk meningkatkan kinerja di domain target. Pendekatan ini sangat berguna ketika domain target memiliki data berlabel yang terbatas. Metode kunci dalam transfer learning meliputi ekstraksi fitur, di mana model yang telah dilatih sebelumnya digunakan untuk mengekstrak fitur relevan dari data baru; fine-tuning, yang melibatkan penyesuaian bobot model yang telah dilatih sebelumnya pada dataset baru untuk meningkatkan kinerjanya pada tugas tertentu; dan adaptasi domain, yang memodifikasi model agar dapat berfungsi dengan baik di berbagai domain dengan distribusi data yang berbeda (Ali et al., n.d.).



Gambar 2. 10 Perbedaan *machine learning* dan *transfer learning* (Ali et al., n.d.)

Seperti yang tergambar pada gambar 2.10, metode transfer learning melibatkan penggunaan pengetahuan dari model yang dilatih pada dataset sumber untuk membantu pelatihan model pada dataset target. Model sumber dilatih terlebih dahulu, dan pengetahuannya (seperti fitur dan parameter) diintegrasikan ke dalam model target, memungkinkan pelatihan tidak dimulai dari nol. Hal ini meningkatkan efisiensi, memungkinkan model target mencapai performa baik bahkan dengan dataset yang lebih kecil, dengan aliran pengetahuan dari model sumber ke model target mempercepat proses pelatihan. (Ali et al., n.d.)

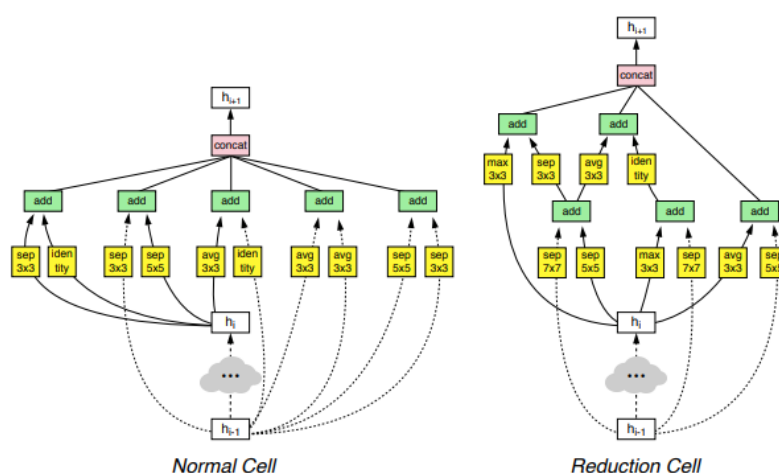
Transfer learning telah berhasil diterapkan di berbagai bidang, terutama dalam penglihatan komputer, di mana model seperti ResNet dan VGG umum digunakan untuk tugas-tugas seperti klasifikasi gambar dan deteksi objek. Studi terbaru menunjukkan bahwa transfer learning dapat secara signifikan meningkatkan akurasi dalam tugas pencitraan medis, seperti mendiagnosis penyakit dari gambar X-ray, dengan memanfaatkan model yang telah dilatih sebelumnya pada dataset besar seperti ImageNet. Dalam ranah pemrosesan bahasa alami (NLP), model bahasa yang telah dilatih sebelumnya seperti BERT dan GPT telah merevolusi berbagai tugas dengan memungkinkan fine-tuning pada aplikasi spesifik seperti analisis sentimen dan tanya jawab. Model-model ini menangkap struktur linguistik yang kompleks dan dapat diadaptasi untuk berbagai tugas bahasa dengan data terbatas. Selain itu, penelitian terbaru dalam penemuan obat telah mengeksplorasi transfer learning dengan jaringan saraf graf untuk

meningkatkan prediksi sifat molekul dengan memanfaatkan data berfidelitas rendah untuk meningkatkan prediksi berfidelitas tinggi, menunjukkan peningkatan kinerja prediktif hingga delapan kali lipat sambil menggunakan jauh lebih sedikit data pelatihan berfidelitas tinggi (Ali et al., 2024).

2.7. NASNet Mobile (Neural Architecture Search)

NASNet adalah arsitektur CNN yang dibangun menggunakan metode NAS oleh tim Google ML yang didasari pada *reinforcement learning* (O Adedoja et al., n.d.). NASNet berlatih dengan metode *reinforcement* saat akurasi R adalah dicapai pada sebuah dataset. Akurasi R digunakan sebagai hadiah sinyal, menggunakan *reinforcement learning* untuk melatih RNN pengontrol. Untuk mengaktifkan pengontrol untuk memprediksi koneksi, NASNet menggunakan jenis *set-selection* yang dibangun di atas mekanisme *attention*. Pada lapisan N, dengan menambahkan titik jangkar yang memiliki sigmoid berbasis konten N – 1 untuk menunjukkan lapisan sebelumnya yang perlu dihubungkan (Zoph & Le, 2016).

Di Arsitektur NASNet terdapat dua jenis *convolutional cell* yang digunakan pada saat ekstraksi fitur yaitu :



Gambar 2. 11 Arsitektur Normal Cell dan Reduction Cell (Zoph & Le, 2016)

- Sel Normal: Ini adalah *convolutional cell* yang biasanya menampilkan peta fitur dengan dimensi yang sama. Misalnya, jika sel mengizinkan input blok yang memiliki peta fitur ukuran $H \times W$ dengan langkah 1, output dihitung pada akhirnya akan berukuran sama dengan fitur tersebut peta.

- Sel Reduksi: Ini juga merupakan *convolutional cell* yang mengembalikan peta fitur, dengan tinggi dan lebar peta fitur dikurangi dengan faktor dua (misalnya jika langkahnya adalah 2, ukurannya dikurangi 2) .

2.8. *Preprocessing Image*

Preprocessing citra adalah langkah awal yang penting dalam pengolahan citra digital, terutama dalam konteks aplikasi pembelajaran mesin dan pengenalan pola. Proses ini bertujuan untuk meningkatkan kualitas citra dan menyiapkan data agar lebih mudah dianalisis oleh algoritma, seperti CNN. Dalam penelitian yang dilakukan oleh Şaban Öztürk dan Bayram Akdemir (2018), efek *preprocessing* pada citra histopatologi dieksplorasi dengan menggunakan berbagai teknik untuk meningkatkan kinerja klasifikasi CNN.

Metode *preprocessing* yang diterapkan termasuk pengurangan *noise* dan peningkatan fitur sel untuk *preprocessing* normal, serta penggunaan *thresholding* dan operasi morfologis untuk *over preprocessing*. Hasil eksperimen menunjukkan bahwa algoritma *preprocessing* normal menghasilkan hasil klasifikasi yang paling sukses, karena mempertahankan fitur-fitur penting dari citra yang diperlukan untuk pembelajaran otomatis oleh CNN. Sebaliknya, *over preprocessing* cenderung menghilangkan banyak fitur penting, sehingga mengurangi akurasi klasifikasi. Penelitian ini menekankan pentingnya memilih teknik *preprocessing* yang tepat untuk meningkatkan kualitas data input dan mendukung kinerja model pembelajaran mesin. (Öztürk & Akdemir, 2018)

Adapun tahapan *preprocessing image* yang dapat meningkatkan performa klasifikasi diantaranya adalah sebagai berikut.

2.8.1. *Resize*

Resize image processing merupakan teknik untuk mengubah ukuran gambar digital, baik dengan cara memperbesar (*upsampling*) maupun memperkecil (*downsampling*) dimensi gambar. Proses *resize* ini penting dalam berbagai aplikasi pengolahan citra dan visi komputer, di mana ukuran gambar harus disesuaikan untuk memenuhi kebutuhan spesifik. Metode *downsampling* sering kali menjadi pilihan dalam pemrosesan image. Hal ini dilakukan karena beberapa alasan: pelatihan dengan *mini-batch* membutuhkan semua gambar dalam satu *batch* memiliki ukuran yang sama, memori yang terbatas tidak memungkinkan pelatihan CNN pada resolusi tinggi, dan gambar berukuran besar membuat

proses pelatihan dan prediksi menjadi lebih lambat. Dengan memori yang terbatas, maka perlu memilih antara ukuran gambar yang lebih besar atau *batch* yang lebih banyak, dan pilihan ini dapat memengaruhi akurasi model CNN secara signifikan (Wang & Yuan, 2014). Proses resize dapat dijelaskan dengan persamaan dasar berikut:

$$I_{new}(x, y) = f(I(x', y')) \quad 2.1$$

Dengan keterangan f adalah fungsi interpolasi yang digunakan untuk memperkirakan nilai piksel baru berdasarkan piksel yang ada.

2.8.2. Normalisasi

Normalisasi adalah tahap penting dalam preprocessing citra yang bertujuan untuk mengatur skala nilai piksel dalam citra agar berada dalam rentang tertentu. Proses ini membantu meningkatkan kualitas dan konsistensi data yang digunakan dalam algoritma pembelajaran mesin. Normalisasi sering kali dilakukan setelah resizing dan sebelum langkah-langkah analisis lebih lanjut, seperti ekstraksi fitur atau klasifikasi. Dengan menormalkan citra, kita dapat mengurangi variasi yang tidak diinginkan yang dapat disebabkan oleh perubahan pencahayaan atau kondisi pengambilan gambar, sehingga model pembelajaran mesin dapat belajar dengan lebih efektif dari data yang disediakan. (Öztürk & Akdemir, 2018)

Salah satu metode umum untuk normalisasi adalah menggunakan persamaan berikut:

$$I' = \frac{I - \min(I)}{\max(I) - \min(I)} \quad 2.2$$

Keterangan:

- I' adalah citra setelah dinormalisasi,
- I adalah citra asli,
- $\min(I)$ dan $\max(I)$ adalah nilai piksel minimum dan maksimum dari citra asli.

2.8.3. Zoom, rotate dan shift

Zoom dalam konteks pengolahan citra merujuk pada proses memperbesar atau memperkecil ukuran tampilan gambar digital. Menurut jurnal yang ditulis oleh Sharma dan Gupta (2015), *zooming* adalah proses di mana gambar diperluas dengan faktor yang lebih besar dari satu, yang dikenal sebagai faktor pembesaran. Proses ini sering kali melibatkan interpolasi untuk menentukan nilai piksel baru yang dihasilkan dari gambar asli, yang berfungsi untuk mengurangi efek samping seperti aliasing dan blur pada tepi

gambar. Persamaan dasar pada proses zoom adalah sebagai berikut.

$$I_{zoom}(x, y) = I\left(\frac{x}{S}, \frac{y}{S}\right) \quad 2.3$$

Keterangan:

- $I_{zoom}(x, y)$ adalah nilai intensitas piksel pada gambar hasil *zoom* di koordinat (x, y) .
- $I(x, y)$ adalah nilai intensitas piksel pada gambar asli di koordinat (x, y) .
- S adalah faktor skala.
- $\left(\frac{x}{S}, \frac{y}{S}\right)$ adalah koordinat piksel pada gambar asli yang sesuai dengan posisi piksel di gambar hasil *zoom*.

Selanjutnya ada proses *rotate* dalam pengolahan citra yang merujuk pada proses memutar gambar pada sudut tertentu, yang dapat dilakukan searah jarum jam atau berlawanan arah jarum jam. Proses rotasi ini penting dalam berbagai aplikasi seperti pengenalan pola, pemrosesan citra medis, dan pengeditan foto. Menurut penelitian yang dilakukan oleh Xu et al. (2008), rotasi gambar dapat mempengaruhi nilai warna piksel, yang pada gilirannya mempengaruhi kualitas visual dari gambar yang telah diputar (Yan-Fang et al., 2008). Persamaan berikut ini yang digunakan pada proses *rotate*.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad 2.4$$

Dengan keterangan (x', y') adalah koordinat baru setelah rotasi, (x, y) adalah koordinat asli, dan θ adalah sudut rotasi.

Selanjutnya ada proses *shift* yang menurut penelitian oleh Azimi (2017), shift merupakan salah satu operasi dasar dalam pemrosesan citra yang digunakan untuk mengubah posisi objek dalam gambar. Dalam konteks konvolusi, proses shift dapat dijelaskan sebagai berikut: ketika sebuah gambar dikonvolusi dengan fungsi delta dua dimensi, hasilnya adalah gambar yang telah digeser sesuai dengan parameter yang ditentukan. Proses ini sangat penting dalam aplikasi seperti penyelarasan citra (image alignment) dan pengolahan sinyal, di mana posisi objek dalam gambar perlu disesuaikan agar sesuai dengan referensi atau norma tertentu (Parker J, 2011). Shift dapat dinyatakan secara matematis sebagai:

$$I_{shift}(x, y) = I(x + dx, y + dy) \quad 2.5$$

Dengan keteranga dx dan dy adalah perubahan posisi pada sumbu x dan y . Dengan menggunakan metode ini, setiap piksel dalam gambar asli dipindahkan ke lokasi baru sesuai dengan nilai offset yang ditentukan.

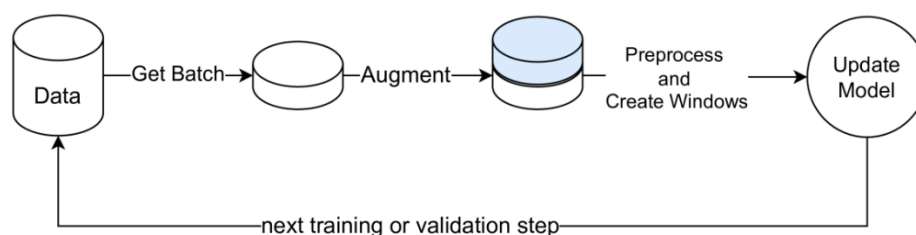
2.8.4. Fill

Fill dalam pengolahan citra merujuk pada proses mengisi area kosong atau lubang yang ada dalam gambar. Proses ini sering kali diperlukan untuk memperbaiki gambar yang memiliki cacat, seperti lubang di objek atau area yang hilang karena kesalahan pengambilan gambar. Dalam "Algorithms for Image Processing and Computer Vision," oleh Parker j menjelaskan bahwa teknik pengisian ini dapat dilakukan dengan berbagai metode interpolasi, termasuk metode flood-fill dan algoritma berbasis region growing (Parker J, 2011).

2.9. On the fly data augmentation

On-the-fly data augmentation, atau augmentasi data secara *real-time*, adalah teknik yang digunakan dalam pelatihan model pembelajaran mendalam untuk meningkatkan keragaman dataset tanpa memerlukan penyimpanan tambahan. Dalam jurnal oleh Shorten dan Khoshgoftaar (2019), teknik ini dijelaskan sebagai proses di mana transformasi acak diterapkan pada gambar saat mereka diumpankan ke model selama pelatihan. Dengan cara ini, setiap gambar yang diterima oleh model mungkin telah dimodifikasi dengan cara yang berbeda, seperti rotasi, pemotongan, atau perubahan warna, sehingga model dapat belajar dari berbagai variasi data.

Salah satu keuntungan utama dari *on-the-fly data augmentation* adalah efisiensi memori. Karena augmentasi dilakukan secara langsung pada gambar yang ada, tidak perlu menyimpan salinan tambahan dari dataset yang diperbesar. Ini sangat berguna ketika bekerja dengan dataset besar atau ketika sumber daya penyimpanan terbatas. Selain itu, teknik ini membantu mengurangi risiko overfitting dengan memberikan variasi yang cukup dalam data pelatihan, sehingga model dapat belajar fitur yang lebih umum dan relevan. (Shorten & Khoshgoftaar, 2019)



Gambar 2. 12 *On the fly data augmentations*(Cerqueira et al., 2024)

Gambar 2.12 menunjukkan alur proses *on-the-fly data augmentation* dalam pelatihan model pembelajaran mesin. Dalam metode ini, data diambil dari dataset utama dalam bentuk batch, yang kemudian akan diproses secara dinamis tanpa penyimpanan permanen dari data yang di augmentasi. Setelah batch diambil, berbagai teknik augmentasi diterapkan, seperti rotasi, *flipping*, *scaling*, atau pengubahan warna, untuk menciptakan variasi dari data asli. Variasi ini membantu memperkaya data yang diberikan pada model, yang meningkatkan kemampuan generalisasi dan mencegah *overfitting*.

2.10. Flask

Flask adalah sebuah micro web *framework* yang ditulis dalam bahasa pemrograman Python, dirancang untuk memudahkan pengembangan aplikasi web dengan cara yang sederhana dan efisien. *Flask* dikembangkan oleh Armin Ronacher dan pertama kali dirilis pada tahun 2010. *Framework* ini disebut "micro" karena tidak memerlukan alat atau pustaka tertentu untuk berfungsi, sehingga memberikan fleksibilitas kepada pengembang untuk memilih komponen tambahan sesuai kebutuhan proyek mereka. *Flask* dibangun di atas dua komponen utama: *Werkzeug*, yang merupakan *toolkit WSGI* (*Web Server Gateway Interface*), dan *Jinja2*, yang berfungsi sebagai mesin template (Walecha, 2021).

Salah satu karakteristik utama dari *Flask* adalah kemampuannya untuk melakukan routing URL, yang memungkinkan pengembang untuk menentukan bagaimana aplikasi merespons permintaan HTTP berdasarkan URL yang diminta. Selain itu, *Flask* mendukung penggunaan berbagai ekstensi yang dapat menambah fungsionalitas aplikasi, seperti *ORM* (*Object Relational Mapping*), validasi formulir, dan pengelolaan file. Dengan fitur-fitur ini, *Flask* sangat cocok digunakan untuk membangun aplikasi kecil hingga menengah, termasuk *API RESTful* dan aplikasi berbasis data (Idris et al., 2020). *Flask* juga dikenal karena kemudahan penggunaan dan dokumentasinya yang luas. Pengembang dapat dengan cepat membangun prototipe aplikasi dan mengimplementasikan fungsionalitas baru tanpa harus terikat pada

struktur yang kaku. Ini menjadikan *Flask* pilihan populer di kalangan pengembang *Python*, terutama untuk proyek-proyek yang memerlukan kecepatan dan efisiensi dalam pengembangan.

Dalam konteks pengembangan aplikasi web, *Flask* memungkinkan integrasi dengan berbagai database seperti *MySQL*, *PostgreSQL*, dan *MongoDB*. Hal ini memberikan fleksibilitas lebih dalam memilih solusi penyimpanan data yang paling sesuai dengan kebutuhan aplikasi. Dengan dukungan untuk pengujian unit dan keamanan cookie yang terintegrasi, *Flask* membantu pengembang memastikan bahwa aplikasi mereka tidak hanya berfungsi dengan baik tetapi juga aman dari potensi kerentanan (Idris et al., 2020).

2.11. *Confusion Matrix*

Confusion matrix adalah tabel dua dimensi yang mencatat jumlah hasil klasifikasi berdasarkan perbandingan antara nilai aktual (klasifikasi sebenarnya) dan nilai prediksi (klasifikasi oleh model). Pada gambar 2.13 dibawah ini, kolom mewakili hasil prediksi model, sementara baris menunjukkan klasifikasi yang sebenarnya.

		PREDICTED classification					
		Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9	
	b	3	9	1	1	14	
	c	1	0	10	2	13	
	d	1	2	1	12	16	
Total		11	11	13	17	52	

Gambar 2. 13 Contoh *confusion matrix*(Grandini et al., 2020)

Kelas-kelas dicantumkan dalam urutan yang sama pada baris dan kolom untuk menjaga konsistensi. Dengan demikian, elemen yang diklasifikasikan dengan benar terletak pada diagonal utama, yaitu dari kiri atas ke kanan bawah. Elemen-elemen ini

merepresentasikan jumlah kasus di mana model sepakat dengan nilai sebenarnya, menunjukkan akurasi klasifikasi untuk masing-masing kelas.(Grandini et al., 2020)

. *Confusion Matrix* digunakan untuk mengukur kinerja model, dengan menggunakan metrik-metrik *accuracy*, *precision*, *recall*, dan *F1-score* Berikut adalah penjelasan dari masing-masing metrik tersebut:

1. *Accuracy*

Nilai yang dihasilkan dari jumlah data yang memiliki nilai positif bersama dengan prediksi yang memiliki nilai negatif juga disebut nilai akurasi. Nilai-nilai ini kemudian dibagi dengan jumlah total data yang digunakan dalam penelitian. Akurasi penelitian berkorelasi positif dengan jumlah data yang diprediksi dengan benar.

$$Accuracy = \frac{TP}{Total\ Data} \quad 2.6$$

2. *Precision*

Nilai peluang yang diprediksi positif dibandingkan dengan nilai keseluruhan hasil yang diprediksi positif dikenal sebagai nilai *Precision*.

$$Precision = \frac{TP}{TP + FP} \quad 2.7$$

3. *Recall*

Nilai *recall* adalah nilai presentase keberhasilan model dalam mengumpulkan informasi yang diinginkan dan nilai *recall* juga merupakan nilai peluang prediksi benar positif dibandingkan dengan nilai keseluruhan bernilai positif.

2.8

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

4. *F1-Score*

F1-Score merupakan perbandingan *mean* dari *precision* dan *recall*. Berikut persamaan dalam menghitung *F1-Score*.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad 2.9$$

2.12. Penelitian Terdahulu

Terdapat beberapa penelitian yang berkaitan dengan citra langit dan juga implementasi transfer learning seperti NASNet yang dapat dilihat pada tabel 2.1.

Tabel 2. 1 Penelitian terdahulu

No	Penulis	Tahun	Judul	Keterangan
1	Naufal, M. F., & Kusuma, S. F.	2022	<i>Weather image classification using convolutional neural network with transfer learning</i>	Hasil eksperimen menunjukkan bahwa arsitektur Xception memberikan akurasi rata-rata terbaik sebesar 90.21%, sementara MobileNetV2 memiliki waktu pelatihan tercepat dengan akurasi 83.51%.
2	Dev, S., S., Hui Lee, Y., S., & Winkler, S.	2016	<i>Color-based Segmentation of Sky/Cloud Images From Ground-based Cameras</i>	Menerapkan teknik segmentasi berbasis warna untuk memisahkan area langit dan awan dalam gambar.
3	A. Taravat, F. del Frate, C. Cornaro, dan S. Vergari	2015	<i>Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images</i>	Penelitian ini mencatat akurasi mencapai 95,07% untuk model jaringan saraf yang digunakan, sementara SVM menunjukkan akurasi yang lebih rendah.
4	K. Tran-Trung, H. Duong Thi Hong, dan V. Truong	2022	<i>Weather Forecast Based on Color Cloud Image Recognition</i>	Mengkaji penggunaan teknik pengenalan citra untuk klasifikasi dan prediksi cuaca berdasarkan gambar awan. Metode yang digunakan adalah kombinasi dari

No	Penulis	Tahun	Judul	Keterangan
	Hoang		<i>under the Combination of Local Image Descriptor and Histogram Selection</i>	deskriptor citra lokal dan pemilihan histogram.
5	S. Liu, M. Li, Z. Zhang, X. Cao, dan T.S. Durrani	2020	<i>Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network</i>	Model TGCN berhasil mengklasifikasikan 8000 gambar awan ke dalam tujuh kategori, mencapai tingkat akurasi 84%.
6	Huang, W., Wang, Y., & Chen, X.	2018	<i>Cloud detection for high-resolution remote-sensing images of urban areas using colour and edge features based on dual-colour</i>	Mengusulkan metode untuk deteksi awan dalam citra penginderaan jauh resolusi tinggi di daerah perkotaan, dengan memanfaatkan fitur warna dan fitur tepi berdasarkan pendekatan dual-color.
7	Nassar S et al.,	2024	<i>A robust MRI-based brain tumor classification via a hybrid deep learning technique</i>	Mengembangkan klasifikasi tumor otak berbasis MRI menggunakan metode transfer learning. Hasil akhir diperoleh dengan akurasi 98,3% menggunakan arsitektur NASNet-mobile.
8	Adedaja et al.,	2022	<i>Intelligent Mobile Plant</i>	Sistem yang dikembangkan mencapai akurasi

No	Penulis	Tahun	Judul	Keterangan
			<i>Disease Diagnostic System Using NASNet-Mobile Deep Learning</i>	sebesar 99.31% dalam mendeteksi penyakit pada daun tanaman. Ini menunjukkan bahwa <i>NASNet Mobile</i> efektif dalam klasifikasi gambar dengan tingkat kesalahan yang sangat rendah
9	Dev S et al,	2015	<i>Categorization of cloud image patches using an improved texton-based approach</i>	Melakukan klasifikasi dataset SWIMCAT dengan metode <i>texton-based</i> yang menghasilkan akurasi 95%.
10	Liu S et al	2020	<i>Multi-Evidence and Multi-Modal Fusion Network for Ground-Based Cloud Evidence and Multi-Modal Fusion Recognition</i>	Klasifikasi dataset MGCD dalam 7 kelas yang menghasilkan akurasi maksimal 88% menggunakan metode <i>Multi-Based Cloud Evidence and Multi-Modal Fusion Network</i>

Naufal, M. F., & Kusuma, S. F. (2022). dengan penelitian yang berjudul “*Weather Image Classification using Convolutional Neural Network with Transfer Learning*”, melakukan klasifikasi gambar cuaca menggunakan *Convolutional Neural Network* (CNN) dengan Transfer Learning. Empat arsitektur CNN, MobileNetV2, VGG16, DenseNet201, dan Xception, digunakan untuk melakukan klasifikasi gambar cuaca. *Transfer learning* digunakan untuk mempercepat proses pelatihan model agar mendapatkan kinerja yang lebih baik dengan lebih cepat. Metode yang diusulkan akan diterapkan pada gambar cuaca yang terdiri dari enam kelas, berawan, hujan, terang, matahari terbit, bersalju, dan kabut. Hasil eksperimen dengan validasi 5-cross dan 50 epoch menunjukkan bahwa Xception memiliki akurasi rata-rata terbaik sebesar 90,21% dengan waktu pelatihan rata-rata selama 10.962 detik dan MobileNetV2 memiliki waktu pelatihan rata-rata tercepat sebesar 2.438 detik dengan akurasi rata-rata sebesar 83,51%. Hasil eksperimen ini menunjukkan bahwa metode yang diusulkan dapat digunakan secara efektif untuk melakukan klasifikasi gambar cuaca yang akurat. Metode ini juga merupakan alternatif yang lebih efisien daripada klasifikasi cuaca

manual oleh manusia, serta dapat memberikan informasi cuaca yang diperbarui secara real-time pada lokasi tertentu (Naufal & Kusuma, 2022).

Dev, S., Member, S., Hui Lee, Y., Member, S., & Winkler, S. (2016). melakukan penelitian Penggunaan gambar langit/awan yang diambil oleh kamera berbasis tanah (seluler yang disebut sebagai pengamat langit seluruhnya) semakin meningkat saat ini karena berbagai aplikasinya di berbagai bidang, termasuk model iklim, prediksi cuaca, generasi energi terbarukan, dan komunikasi satelit. Akibat adanya berbagai jenis awan dan kondisi cahaya yang berbeda-beda pada gambar-gambar tersebut, segmentasi awan yang akurat dan tahan lama menjadi tantangan. Dalam makalah ini, penulis memperkenalkan kerangka segmentasi yang diawasi untuk gambar langit/awan berbasis tanah berdasarkan analisis sistematis dari berbagai ruang warna dan komponen, menggunakan regresi *least squares parsial (PLS)*. Berbeda dengan metode terbaik dalam bidang ini, pendekatan yang di tawarkan sepenuhnya didasarkan pada pembelajaran dan tidak membutuhkan parameter yang didefinisikan secara manual. Selain itu, peneliti juga merilis *Singapore Whole Sky IMaging SEGmentation Database (SWIMSEG)*, sebuah database gambar langit/awan yang dilabelkan yang dapat digunakan oleh komunitas peneliti (Dev et al., 2016).

Huang, W., Wang, Y., & Chen, X. (2018). Membahas deteksi awan pada citra jarak jauh beresolusi tinggi dari wilayah perkotaan menggunakan fitur warna dan tepi berdasarkan model dual-warna. Hasil pengujian menunjukkan bahwa metode ini dapat menghasilkan segmentasi awan yang baik dibandingkan beberapa algoritma pembandingnya yaitu SVM dan PRS (Huang et al., 2018).

S. Liu, M. Li, Z. Zhang, X. Cao, dan T.S. Durrani (2020) mengadakan penelitian yang membahas tentang bagaimana awan dapat meningkatkan performa klasifikasi gambar awan dari bawah tanah. Metode yang digunakan dalam penelitian ini adalah *Context Graph Attention Network (CGAT)*, yang dapat memanfaatkan hubungan antara gambar awan. Selain itu, peneliti juga merilis database awan bawah tanah terbesar untuk meningkatkan penelitian dan membandingkan metode yang berbeda. Hasil dari eksperimen yang dilakukan menunjukkan bahwa CGA dapat digunakan dengan efektif untuk klasifikasi gambar awan dengan akurasi sebesar 84%. peneliti dapat menyimpulkan bahwa penggunaan CGA akan membantu dalam meningkatkan performa klasifikasi gambar awan bawah tanah, yang akan membantu dalam proses prediksi cuaca, siklus air, dan sistem iklim (Liu et al., 2022).

S. Moughyt, Z. el Jaouhari, O. el Kadmiri, L. Masmoudi, Z. el Kadmiri, dan Y.

Zaz (2015) mengusulkan pendekatan yang menggabungkan teknik segmentasi citra dan pengolahan citra untuk estimasi penutupan awan pada citra yang diambil dari kamera dengan metode Otsu's dan *multi-objective optimization*. Penulis menggunakan dataset yang diambil dari beberapa lokasi di Maroko dan menunjukkan bahwa pendekatan mereka dapat menghasilkan hasil yang lebih baik dibandingkan dengan teknik segmentasi citra konvensional (Moughyt et al., 2015).

A. Taravat, F. del Frate, C. Cornaro, dan S. Vergari (2015) menggunakan teknik klasifikasi awan berbasis jaringan saraf dan SVM untuk mengklasifikasikan citra yang diambil dari kamera. Penulis menggunakan dataset dari beberapa lokasi di Italia untuk mengelompokkan citra langit kedalam 4 jenis yaitu *clear-sky*, *dark thick cloud*, *high thin cloud* dan *patched cloud* (Taravat et al., 2015). Penelitian ini mengeksplorasi penggunaan berbagai fitur, termasuk fitur warna dan tekstur pada citra untuk melatih model. Mereka membandingkan kinerja dua algoritma pembelajaran mesin yang populer, yaitu Neural Networks dan algoritma Support Vector Machine, dalam hal akurasi dan kecepatan. Dari hasil penelitian tersebut, performa dari MLP lebih baik dari pada SVM dengan akurasi hingga 95,07% sedangkan SVM sebesar 93,66% (Taravat et al., 2015).

K. Tran-Trung, H. Duong Thi Hong, dan V. Truong Hoang (2022), melakukan penelitian yang bertujuan untuk mengurangi jumlah histogram yang digunakan dalam mengidentifikasi dan klasifikasi awan dengan menggabungkan karakteristik warna, fitur LBP, dan fitur LTP. Penelitian ini juga menggunakan teknik Intra-Class Similarity (ICS) sebagai cara untuk memilih histogram yang potensial. Penelitian ini menyimpulkan bahwa dengan menggabungkan fitur LBP dan LTP dan dengan menggunakan teknik ICS, maka diperoleh hasil yang lebih baik dalam pengenalan awan tanpa mengurangi keseluruhan jumlah fitur yang digunakan dengan akurasi akhir yang diperoleh sebesar 92,20 %. Hasil penelitian ini bermanfaat bagi para peneliti dalam bidang pengenalan pola guna memecahkan masalah pengenalan pola yang lebih kompleks (Tran-Trung et al., 2022).

2.13. Perbedaan penelitian

Naufal, M. F., & Kusuma, S. F. pada tahun 2022 juga menggunakan beberapa metode transfer learning. Meskipun memiliki persamaan yaitu menggunakan metode transfer learning, namun penelitian kali ini akan menggunakan objek yang berbeda dari penelitian

tersebut. Dimana penelitian ini berfokus pada citra langit saja tanpa objek lain didalamnya.

Lalu ada juga penelitian lain yang menggunakan citra langit, seperti penelitian oleh Dev, S., et al, S. Moughyt et al, dan juga Huang et al namun ketiga penelitian tersebut melakukan proses segmentasi bukan klasifikasi citra langit. Ketiga nya melakukan segmentasi citra dengan metode yang berbeda-beda, namun sama-sama menggunakan citra yang diambil dari daratan bukan citra dari satelit.

Selanjutnya ada penelitian yang dilakukan oleh A. Taravat, F. del Frate, C. Cornaro, dan S. Vergari pada tahun 2015 menggunakan MLP dan juga SVM sebagai perbandingannya. Pada penelitian oleh Taravat et al., ini hasil maksimal akurasi yang peroleh sebesar 95,07% yang diperoleh dengan pemanfaatan algoritma MLP dengan mengelompokkan 4 jenis kondisi yang berbeda. Sedangkan untuk penelitian yang dilakukan ini akan memanfaatkan salah satu algoritma transfer learning yaitu NASNet. Dan penggunaan dataset yang berbeda yaitu dataset SWIMCAT (Singapore Whole Sky IMaging CATegory) dengan melakukan pengelompokkan terhadap 5 jenis kondisi yang berbeda yaitu *clear sky*, *pattern*, *thick dark cloud*, *thick white cloud* dan *veil*. Begitupun dengan jumlah data yang akan diolah lebih besar dari pada penelitian sebelumnya dengan jumlah hingga 782 citra yang akan dibagi menjadi data latih dan data uji. Dengan menggunakan data lebih besar diharapkan dapat memberikan hasil yang lebih maksimal dibandingkan penelitian-penelitian sebelumnya.

Adapun berdasarkan penelitian Nassar S et al., terbaru pada tahun 2024, penelitian tersebut membandingkan beberapa jenis algoritma transfer learning yaitu *GoogleNet*, *AlexNet*, *SqueezeNet*, *ShufNet*, dan *NASNet-Mobile*. Klasifikasi menggunakan NASNet sebagai salah satu jenis *transfer learning* dapat memberikan akurasi yang tinggi mencapai 98,3% untuk mengklasifikasi penyakit tumor otak. Bahkan pada penelitian oleh Adedoja et al pada tahun 2022, NASNet mobile mampu menghasilkan akurasi hingga 99,31% untuk melakukan klasifikasi penyakit berdasarkan citra tumbuhan. Sedangkan untuk penelitian ini akan menggunakan dataset citra langit.

Secara keseluruhan, klasifikasi citra langit menggunakan metode *CNN* dengan arsitektur *NASNet* memiliki beberapa perbedaan dibandingkan penelitian lain dengan arsitektur *CNN* tradisional. *NASNet* dikembangkan melalui *Neural Architecture Search*, yang secara otomatis menghasilkan struktur jaringan optimal untuk meningkatkan akurasi dan efisiensi seperti yang telah dilakukan oleh Nassar S et al.

Objek penelitian kali ini difokuskan pada citra langit, yang memerlukan penanganan khusus terhadap variasi awan dan kondisi cuaca. Penelitian ini juga melakukan implementasi *on-the-fly data augmentation* dalam *NASNet* memungkinkan augmentasi data real-time selama pelatihan, meningkatkan variasi data dan generalisasi model, sedangkan penelitian lain sering menggunakan augmentasi statis. Pendekatan ini membuat *NASNet* lebih efisien secara komputasi dan lebih fleksibel untuk perangkat dengan daya terbatas, serta meningkatkan peluang untuk menghasilkan akurasi tinggi pada tugas klasifikasi citra langit.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Di bab ini, peneliti akan membahas analisis dan perancangan sistem untuk aplikasi deteksi citra awan menggunakan arsitektur *NASNET*. Pertama, peneliti akan menjelaskan analisis data yang digunakan dalam penelitian, diikuti oleh penjabaran implementasi metode deteksi citra awan. Selain itu, bab ini juga akan mencakup pembahasan mengenai proses pelatihan (*training*) dan pengujian (*testing*).

3.1. Data Penelitian

Dataset SWIMCAT (Singapore Whole sky IMaging CATegories) adalah sebuah kumpulan data citra langit yang dikumpulkan dari berbagai waktu dan kondisi cuaca di Singapura. Dataset ini digunakan untuk berbagai penelitian terkait pengenalan fenomena langit dan klasifikasi citra langit. Berikut beberapa karakteristik dataset SWIMCAT:

1. Keanekaragaman Citra Langit, Dataset ini mencakup berbagai citra langit yang menampilkan kondisi cuaca yang berbeda, seperti cuaca cerah, berawan, hujan, matahari terbenam, dan lainnya. Ini memungkinkan analisis dan klasifikasi citra langit dalam berbagai konteks.



Gambar 3.3. Class dalam SWIMCAT

2. Labeling, Setiap citra langit dalam dataset SWIMCAT dilengkapi dengan label yang mengidentifikasi jenis fenomena langit yang terdapat dalam citra tersebut. Label-label ini memfasilitasi pembuatan model klasifikasi citra.
3. Jumlah Citra, Dataset ini mungkin terdiri dari 784 citra langit yang dikumpulkan dari berbagai sumber dan waktu. Jumlah citra yang besar dapat meningkatkan kemampuan model dalam mengenali pola-pola pada citra.

Tabel 3. 1 Rasio jumlah dataset SWIMCAT

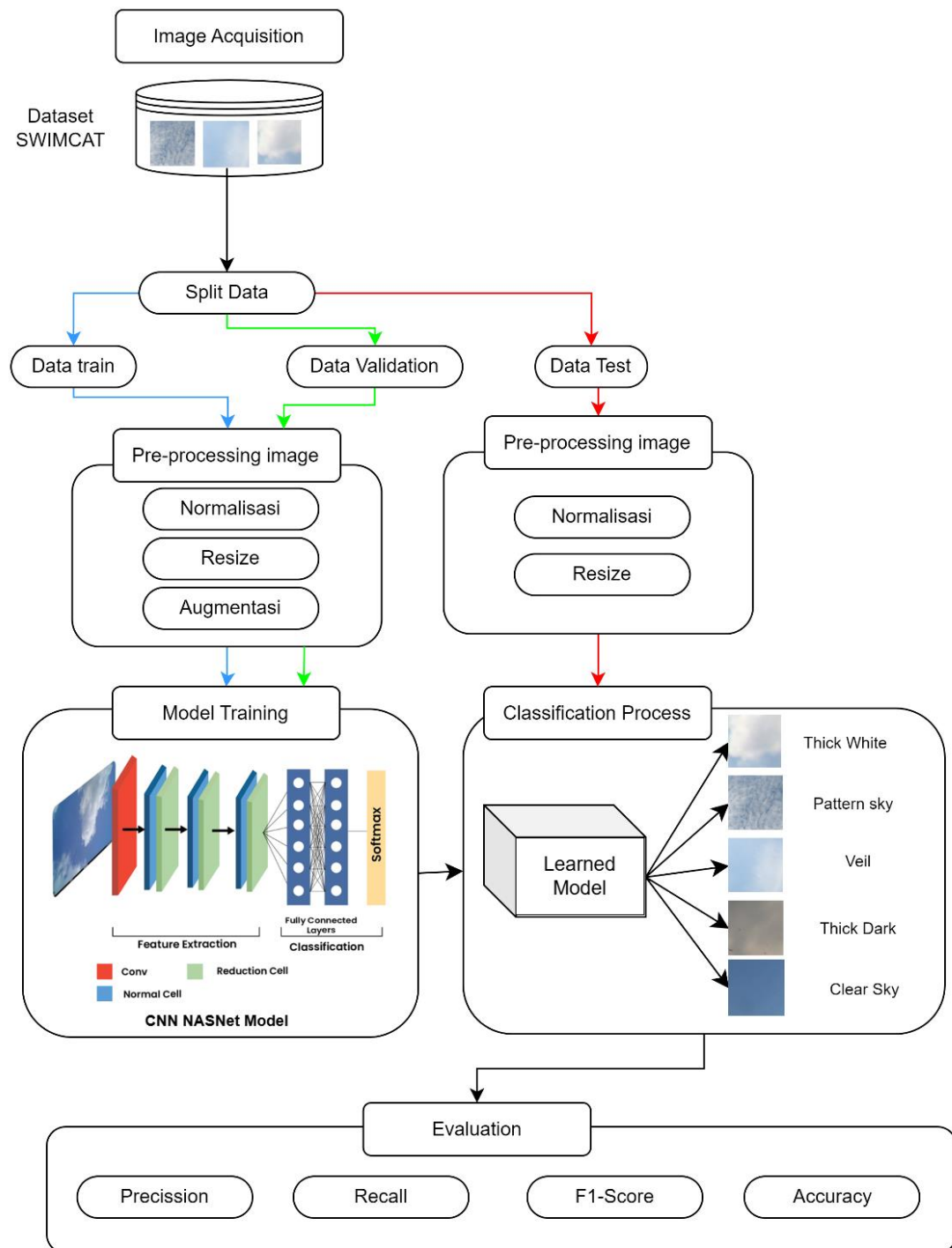
Label	Jumlah	Rasio
<i>Clear Sky</i>	224	28.5%
<i>Pattern</i>	89	11.3%
<i>Thick Dark Cloud</i>	251	32.0%
<i>Thick White Cloud</i>	135	17.2%
<i>Veil</i>	85	10.8%

4. Kualitas Resolusi, Citra-citra dalam dataset ini mungkin memiliki resolusi 125x125 pixel.
5. Referensi Geografis, Dataset ini dikumpulkan di Singapura, sehingga mungkin mencerminkan karakteristik langit dan cuaca kawasan tersebut.
6. Akses Dataset, Dataset SWIMCAT dapat tersedia secara publik atau melalui permintaan khusus kepada pihak yang mengelolanya. Pengguna harus mematuhi lisensi dan persyaratan penggunaan yang telah ditentukan oleh penyedia dataset.

Data input ini akan digunakan dalam proses pelatihan model CNN untuk mengenali dan mengklasifikasikan berbagai fenomena langit pada siang hari. Setiap citra akan diumpankan ke model sebagai input, dan model akan menghasilkan prediksi tentang apa yang terdapat dalam citra tersebut, seperti langit dengan awan tipis, langit dengan awan tebal dan lain-lain. Hasil prediksi ini kemudian akan digunakan untuk evaluasi dan analisis performa model.

3.2. Analisis Sistem

Sistem klasifikasi yang akan dibangun dirancang menggunakan algoritma NASNet, sebuah arsitektur jaringan saraf yang dioptimalkan melalui Neural Architecture Search (NAS). Sistem ini diharapkan mampu memberikan performa tinggi dalam mengklasifikasikan data gambar dengan presisi dan efisiensi. Dengan memanfaatkan fleksibilitas NASNet, sistem ini dapat diadaptasi untuk berbagai tugas klasifikasi, baik untuk dataset sederhana maupun kompleks, sambil memastikan penggunaan sumber daya komputasi yang optimal. Hasil dari sistem ini akan dikembangkan dalam bentuk website agar mudah diakses oleh pengguna. Berikut ini adalah gambaran dari sistem klasifikasi yang akan dibangun menggunakan algoritma NASNet.



Gambar 3. 1 Arsitektur umum sistem klasifikasi dengan *NASNet*

Gambar 3.1 diatas menggambarkan setiap tahapan pada penelitian "Klasifikasi kondisi langit menggunakan *CNN* dengan arsitektur *NASNet*". Berikut penjelasan dari tiap tahapan tersebut:

3.2.1 Image Acquisition

Data yang akan digunakan berupa citra langit yang diperoleh dari Singapore Whole sky IMaging CATegories Database (Dev et al., n.d.) . Data ini digunakan karena

diambil dari negara Singapura yang memiliki iklim tropis mirip dengan Indonesia. Dataset ini berisikan 784 gambar langit/awan, yang dikategorikan dalam 5 kategori yang berbeda. Citra yang tersedia memiliki resolusi 125 x 125 pixel yang diambil dari Nanyang Technological University Singapore. Data citra langit yang sudah terkumpul dibagi menjadi tiga bagian yaitu untuk data testing, data training dan juga data validation . Berikut adalah rasio pembagian setiap bagian data.

Tabel 3. 2 Rasio perbandingan pembagian data

Data	Jumlah	Rasio
<i>Training</i>	555	70%
<i>Testing</i>	159	20%
<i>Validation</i>	80	10%

Pada tabel tabel 3.2 dapat diketahui bahwa pembagian data training memiliki jumlah yang lebih besar bertujuan untuk memaksimalkan pembelajaran model dan menghindari *overfitting* sehingga mampu mengenali pola-pola yang lebih luas. Adapun penggunaan data *validation* agar selama proses pelatihan sistem dapat memantau performa model dan membantu menilai kinerja model pada data yang tidak dilihat selama pelatihan model.

3.2.2 Image Pre-Processing

Pada tahapan ini dilakukan proses pre-processing atau proses awal untuk mengubah citra asli menjadi bentuk yang dapat diproses untuk tahap selanjutnya. Proses ini penting agar meningkatkan efisiensi dan hasil prediksi yang lebih akurat. Adapun beberapa tahapan yang dilakukan dapat dilihat pada bagan pseudocode dibawah ini.

Algorithm *image preprocessing*

Input: SWIMCAT dataset

Read: Data

FOREACH DATA

1. Data \leftarrow Resize 2.1
2. Data \leftarrow Normalize 2.2
3. Data \leftarrow Randomly Rotate 30 degrees 2.4
4. Data \leftarrow Zooming range within 20% 2.3
5. Data \leftarrow Shift Width and Height range within 10% 2.5
6. Data \leftarrow Flip horizontal
7. Data \leftarrow Fill empty pixel with nearest pixel value

END

return Cleaned data

Pada kode di atas, preprocessing dilakukan pada data *train* dan juga valid dengan menggunakan *ImageDataGenerator* untuk augmentasi dan normalisasi data gambar. Proses pertama adalah normalisasi dengan parameter $\text{rescale} = \frac{1}{255}$, yang mengubah skala nilai piksel dari rentang [0, 255] menjadi [0, 1]. Normalisasi ini bertujuan untuk mencegah nilai piksel yang besar mendominasi komputasi dan membantu optimisasi model agar lebih stabil dan cepat.

Selanjutnya ada tahapan augmentasi yang pertama adalah rotasi dengan $\text{rotation_range}=30$, yang mengizinkan gambar berotasi hingga 30 derajat ke kiri atau ke kanan. Jika titik asal dari koordinat gambar adalah (x, y), setelah rotasi sebesar θ derajat, koordinat baru (x', y'). Rotasi ini memungkinkan model belajar mengenali objek pada sudut yang berbeda, sehingga lebih robust terhadap orientasi objek.

Proses berikutnya adalah *zoom* dengan $\text{zoom_range}=0.2$, yang memperbesar atau memperkecil gambar dalam rentang 20%. Pada proses ini, setiap piksel diperbesar atau diperkecil sesuai faktor skala (s) yang diambil secara acak dalam rentang ini. Zoom menambah variasi pada gambar sehingga model belajar mengenali objek dalam berbagai ukuran.

Kemudian, terdapat pergeseran lebar dan tinggi menggunakan $\text{width_shift_range}=0.1$ dan $\text{height_shift_range}=0.1$, yang menggeser gambar secara horizontal dan vertikal hingga 10% dari total lebar atau tinggi gambar. Proses ini memungkinkan model belajar dari posisi objek yang berbeda-beda dalam gambar, baik ke kiri-kanan maupun ke atas-bawah.

Selanjutnya, proses *flip horizontal* dengan $\text{horizontal_flip} = \text{True}$ menghasilkan gambar yang dicerminkan secara horizontal, dari kiri ke kanan. Jika gambar memiliki koordinat (x, y), maka setelah flip horizontal, koordinat baru akan diperoleh. Flip horizontal ini membantu model lebih fleksibel terhadap arah objek, khususnya pada gambar simetris.

Terakhir, $\text{fill_mode}='nearest'$ digunakan untuk mengisi area kosong yang muncul akibat rotasi, pergeseran, atau zoom. Dengan pengaturan ini, area kosong diisi dengan nilai piksel terdekat, sehingga gambar hasil augmentasi tidak memiliki bagian hitam atau kosong. Dengan cara ini, gambar terlihat lebih natural tanpa

kehilangan informasi penting di area kosong. Secara keseluruhan, preprocessing ini bertujuan untuk meningkatkan variasi data yang dilihat model, memperkaya dataset tanpa menambah data fisik, serta membuat model lebih general dan robust terhadap variasi pada gambar.

Proses augmentasi di atas tidak secara langsung menambah jumlah data yang disimpan di dataset, tetapi menambah variasi data secara dinamis selama proses pelatihan. Augmentasi data menggunakan *ImageDataGenerator* menghasilkan transformasi berbeda dari gambar yang sama setiap kali gambar tersebut diambil untuk satu batch pelatihan. Artinya, setiap epoch, model menerima versi baru dari gambar asli, misalnya dengan rotasi, pergeseran, atau flip yang berbeda. Ini membantu model belajar dari variasi yang lebih luas tanpa benar-benar menambah ukuran dataset. Jadi, jumlah file gambar atau data fisik tidak bertambah di penyimpanan. Augmentasi hanya menambah variasi data yang dilihat oleh model selama pelatihan, membantu mengurangi risiko model menghafal data dan meningkatkan kemampuan generalisasi tanpa memperbesar dataset secara eksplisit.

3.2.3 Model

Pada penelitian ini, algoritma yang telah ditentukan adalah *CNN* dengan arsitektur *NASNet Mobile* dalam klasifikasi citra awan. Setelah proses *pre-processing* dan *data splitting*, *data train* dan *data valid* yang sudah diolah akan diproses dengan algoritma pada saat training.

1. Install Requirement

Pada proses ini semua package yang akan digunakan pada proses training maupun testing akan di install. seperti *ultralytics*, *pytorch* dan *dependencies* lain yang digunakan. Berikut adalah *library* yang digunakan pada sistem ini diantaranya *tensorflow*, *keras*, *numpy*, *pandas*, *sklearn*, *seaborn*, dan *matplotlib*.

2. Pretrained Model

Pada tahap pelatihan model (*training*), digunakan algoritma *Convolutional Neural Network (CNN)* dengan arsitektur *NASNet Mobile*. Arsitektur *NASNet Mobile* yang digunakan merupakan pretrained model, yaitu model yang telah dilatih sebelumnya pada dataset besar seperti ImageNet. Penggunaan pretrained model ini memberikan keuntungan signifikan, termasuk kemampuan untuk memanfaatkan bobot awal yang telah dioptimalkan, sehingga proses pelatihan menjadi lebih cepat dan akurat, terutama

saat dataset pelatihan relatif kecil. Berikut ini adalah struktur model yang digunakan dalam tahap pelatihan, yang mencakup lapisan-lapisan utama *NASNet Mobile* yang disesuaikan untuk mengklasifikasikan data sesuai kebutuhan penelitian.

Layer (type)	Output Shape	Param #
NASNet (Functional)	(None, 7, 7, 1056)	4269716
conv2d (Conv2D)	(None, 7, 7, 32)	304160
max_pooling2d (MaxPooling2D)	(None, 3, 3, 32)	0
dropout (Dropout)	(None, 3, 3, 32)	0
flatten (Flatten)	(None, 288)	0
dense (Dense)	(None, 5)	1445
Total params: 4,575,321		
Trainable params: 305,605		
Non-trainable params: 4,269,716		

Gambar 3. 2 *Summary model*

Berdasarkan *summary model* yang ditampilkan pada gambar 4.1 menggunakan arsitektur *NASNet* sebagai lapisan dasar. *NASNet* merupakan arsitektur CNN yang dirancang oleh Google menggunakan teknik *neural architecture search*. Pada model ini, *NASNet* menghasilkan keluaran dengan dimensi (7, 7, 1056), yang berarti memiliki ukuran spasial 7x7 dengan 1056 fitur. Lapisan *NASNet* memiliki jumlah parameter yang sangat besar, yaitu 4.269.716, namun sebagian besar parameter ini bersifat *non-trainable* atau tidak dapat dioptimasi.

Setelah *NASNet*, model dilanjutkan dengan lapisan konvolusi (*Conv2D*) yang menerapkan 32 filter untuk memproses keluaran dari *NASNet*, menghasilkan keluaran berukuran (7, 7, 32) dengan total parameter yang bisa dioptimasi sebanyak 304.160. Selanjutnya, ada lapisan *MaxPooling2D* yang digunakan untuk mengurangi ukuran spasial keluaran menjadi (3, 3, 32) tanpa menambah parameter baru. Setelah *MaxPooling2D*, terdapat lapisan *Dropout* untuk membantu mengurangi kemungkinan *overfitting* dengan menonaktifkan beberapa neuron secara acak selama pelatihan. Lapisan ini tidak memiliki parameter tambahan.

Lapisan berikutnya adalah *Flatten*, yang mengubah keluaran berbentuk matriks (3, 3, 32) menjadi vektor satu dimensi berukuran 288 untuk disiapkan bagi lapisan akhir. Akhirnya, lapisan *Dense* berfungsi sebagai lapisan output dengan 5 unit, menghasilkan keluaran yang sesuai untuk klasifikasi 5 kelas. Lapisan ini memiliki 1.445 parameter yang dapat dioptimasi.

3. *Evaluation*

Pada tahap **ini**, model yang telah dilatih digunakan untuk memprediksi kelas dari data uji (*testing set*). Hasil prediksi ini kemudian dibandingkan dengan label sebenarnya untuk mengevaluasi kinerja model. Berikut adalah kode semu yang menampilkan alur

proses pengembangan model klasifikasi menggunakan CNN dengan arsitektur NASNet.

Algorithm *NASNet mobile model*

Input: *SWIMCAT dataset*

Install requirement

Read: *Dataset (X: array of images, Y: array of output classes)*

PretrainedModel \leftarrow *Load NASNet Mobile with ImageNet weights*

CNNModel \leftarrow *Modify NASNet Mobile layer*

XC \leftarrow *CNNModel.predict(XP)*

(XCT, YT), (XCV, YV), (XCS, YS) \leftarrow *Split XC and Y into:*

- 70% Training Set
- 10% Validation Set
- 20% Testing Set

FC_OutputLayer \leftarrow *Add fully connected layer with:*

- Number of neurons = 5 (number of dataset classes)
- Activation = softmax

For *epoch* **in** *range(1, 50)* **do**

Retrain *FC_OutputLayer* **using:**

- Training Set: *(XCT, YT)* \leftarrow *augmentation_images (X)*
- Validation Set: *(XCV, YV)* \leftarrow *augmentation_images(X)*

Update:

- *Loss.append(validation_loss)*
- *Acc.append(validation_accuracy)*

If *stopping criteria met (epochs > 50)* **do**

Break

Save *model*

Predictions (P) \leftarrow *Predict(XCS)*

OverallAccuracy \leftarrow *CalculateAccuracy(P, YS)*

return *Trained model*

Kode semu ini menjelaskan proses pelatihan model *NASNet Mobile* untuk dataset *SWIMCAT*. Proses dimulai dengan membaca dataset, di mana *X* adalah array gambar dan *Y* adalah array kelas output. Model *NASNet Mobile* dimuat menggunakan

bobot pra-latih dari dataset *ImageNet*, lalu disesuaikan dengan mengganti lapisan atasnya sesuai dengan rangkuman model pada gambar 3.2. Selanjutnya, fitur gambar diekstraksi menjadi *XC*, dan dataset dibagi menjadi tiga bagian: *training set* (70%), *validation set* (10%), dan *testing set* (20%).

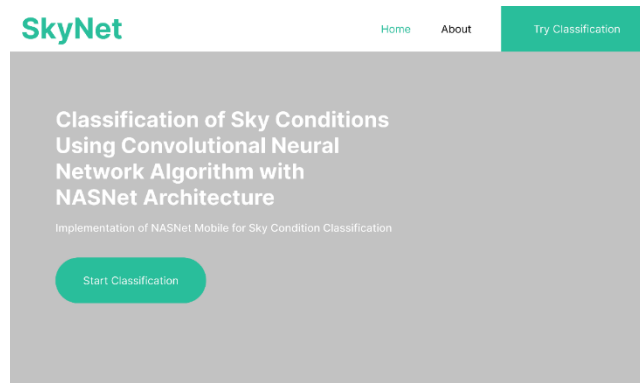
Sebuah lapisan *fully connected* baru ditambahkan ke model dengan jumlah neuron sama dengan jumlah kelas dalam dataset (yaitu 5 kelas) dan fungsi aktivasi *softmax* untuk menghasilkan probabilitas kelas. Model kemudian dilatih selama maksimal 50 *epochs* dengan menggunakan *data augmentation* pada data pelatihan dan validasi untuk meningkatkan keragaman data. Setelah setiap *epoch*, performa model dicatat dalam bentuk kehilangan validasi (*validation loss*) dan akurasi validasi (*validation accuracy*). Pelatihan dihentikan jika mencapai batas *epochs* atau jika model telah memenuhi kriteria tertentu untuk menghindari *overfitting*. Setelah pelatihan selesai, model yang telah dilatih disimpan dan dikembalikan untuk evaluasi atau penggunaan lebih lanjut. Proses ini memastikan model dapat beradaptasi dengan dataset yang spesifik sambil mempertahankan performa yang optimal.

3.3. Perancangan antarmuka sistem

Perancangan antarmuka sistem bertujuan untuk memudahkan proses membangun sistem yang efektif untuk digunakan sebelum akhirnya melanjutkan ke proses pengembangan sistem lebih lanjut. Rancangan ini dibuat dengan struktur website sesuai dengan hasil sistem yang ingin dibuat. Platform website memudahkan pengguna untuk mengakses sistem dari berbagai jenis perangkat dan tidak membutuhkan sumber daya yang besar untuk mengembangkan sistem nya.

3.3.1 Tampilan *Landing Page*

Landing page SkyNet memiliki desain sederhana seperti yang terlihat pada gambar 3.3, namun efektif untuk memprioritaskan kejelasan dan kemudahan navigasi bagi pengguna. Dengan tata letak yang rapi dan antarmuka minimalis, halaman ini memastikan setiap elemen mudah dipahami dan digunakan. Warna-warna yang dipilih memberikan kontras yang nyaman di mata, sementara tombol aksi yang menonjol memandu pengguna menuju fitur utama dengan intuitif.

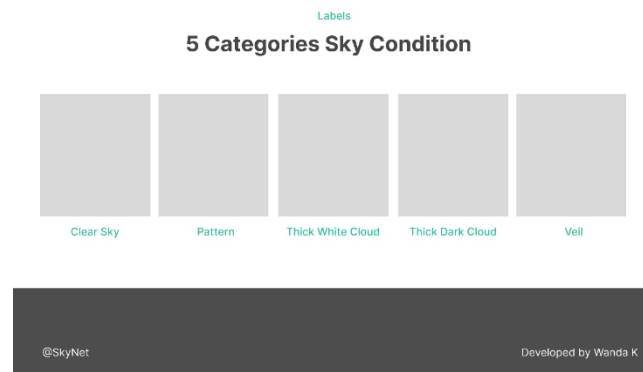


Gambar 3. 3 Rancangan antarmuka *landing page* web klasifikasi

Logo "SkyNet" ditempatkan di sudut kiri atas halaman dengan warna hijau cerah, memberikan identitas visual yang langsung menarik perhatian. Di sebelahnya, terdapat menu navigasi "Home," "About," dan "Try Classification," yang memudahkan pengguna untuk berpindah ke berbagai halaman penting. Penempatan menu di sisi kanan atas mengikuti standar desain *web*, sehingga pengguna dapat dengan mudah menemukan dan mengakses pilihan navigasi.

Di tengah halaman, terdapat judul utama berukuran besar, "*Classification of Sky Conditions Using Convolutional Neural Network Algorithm with NASNet Architecture*," yang secara langsung menjelaskan tujuan utama platform ini. Di bawah judul, ada deskripsi singkat yang memberi informasi lebih lanjut tentang teknologi yang digunakan, yaitu *NASNet Mobile*, sehingga menarik minat pengguna yang memiliki ketertarikan pada aspek teknis. Kedua elemen ini disusun dengan baik untuk memberikan gambaran cepat mengenai fungsi dan keunggulan dari situs.

Bagian terpenting dari halaman ini adalah tombol hijau "*Start Classification*" yang terletak tepat di bawah teks utama. Tombol ini berfungsi sebagai ajakan bagi pengguna untuk memulai proses klasifikasi kondisi langit, dengan penempatan dan warna yang kontras untuk meningkatkan visibilitas dan mendorong tindakan. Latar belakang abu-abu yang sederhana akan digantikan dengan foto bertema langit atau cuaca, yang tidak hanya relevan dengan konten tetapi juga menambah estetika halaman dan memperkuat kesan tematik. Penggunaan gambar ini juga akan menambah kedalaman visual, tanpa mengurangi fokus pada teks dan tombol utama. Secara keseluruhan, desain halaman ini menonjolkan keprofesionalan, kesederhanaan, dan memberikan panduan jelas bagi pengguna untuk langsung mencoba fitur yang ditawarkan.



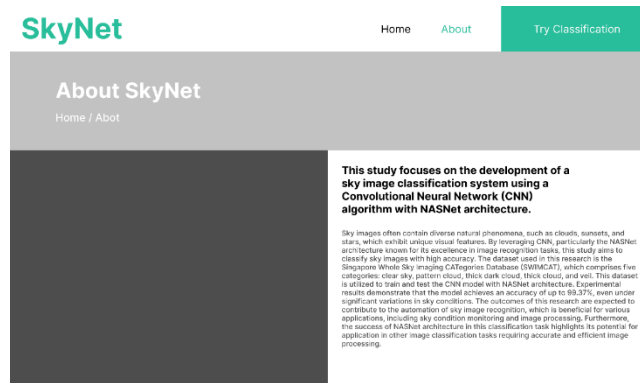
Gambar 3. 4 Rancangan antarmuka penjelasan kategori

Bagian kategori label kondisi langit pada halaman *landing page* SkyNet memiliki desain yang sederhana namun informatif. Di bagian atas, terdapat judul utama "*5 Categories Sky Condition*" dengan ukuran font besar untuk menarik perhatian, dilengkapi sub-judul "*Labels*" yang memberikan konteks bahwa kategori-kategori di bawahnya adalah label kondisi langit yang dikenali oleh sistem. Di bawah judul, lima kolom sejajar menampilkan masing-masing kategori: *Clear Sky*, *Pattern*, *Thick White Cloud*, *Thick Dark Cloud*, dan *Veil*. Bagian abu-abu muda dalam kolom ini nantinya akan diisi gambar contoh kondisi langit, sementara label berwarna hijau di bawahnya membantu pengguna mengenali setiap kategori dengan mudah.

Bagian bawah halaman dilengkapi *footer* berwarna abu-abu gelap yang berfungsi sebagai penutup, mencakup informasi seperti "*@SkyNet*" dan "*Developed by Wanda K.*" Penempatan *footer* ini memberi ruang untuk informasi tambahan tanpa mengganggu fokus utama, sementara warna gelapnya menciptakan kontras agar tetap terlihat namun tidak mencolok. Desain keseluruhan halaman ini membantu pengguna memahami klasifikasi kondisi langit dengan tampilan yang terstruktur dan mudah dipahami.

3.3.2 Tampilan laman *about*

Laman *About* pada halaman SkyNet dibawah ini dirancang untuk memberi gambaran lengkap tentang tujuan dan latar belakang sistem yang dibangun.

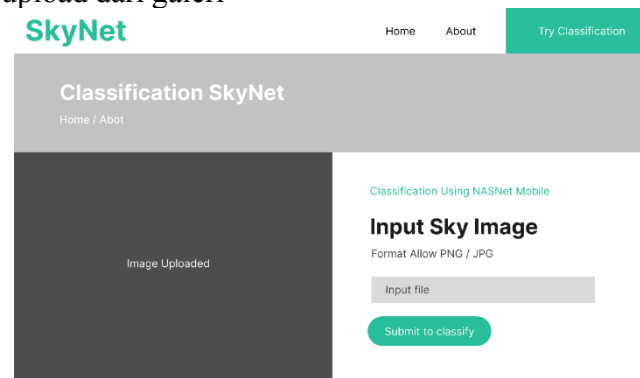


Gambar 3. 5 Rancangan antarmuka laman *about*

Di bagian atas, ada latar abu-abu yang akan diisi dengan foto, tujuannya untuk membuat halaman ini lebih hidup dan menarik. Tepat di tengahnya, terlihat tulisan besar "*About SkyNet*" yang langsung memberi tahu pengguna bahwa halaman ini berisi informasi tentang proyek SkyNet. Di bawah judul utama, ada tulisan kecil "*Home / About*" sebagai penanda jalur, jadi pengguna bisa tahu posisi mereka di situs ini.

Di bagian bawah, ada teks panjang yang menjelaskan inti dari proyek ini. Teks ini diawali dengan kalimat yang ditebalkan untuk menarik perhatian, menjelaskan bahwa penelitian ini fokus pada pengembangan sistem klasifikasi gambar langit dengan menggunakan *Convolutional Neural Network (CNN)* berbasis arsitektur *NASNet*. Dijelaskan juga kenapa NASNet dipilih, yaitu karena kemampuannya yang sangat baik dalam pengenalan gambar. Di sini, pengguna bisa tahu bahwa model ini sudah diuji dengan dataset yang berisi berbagai macam kondisi langit, seperti langit cerah, awan tebal, dan lain-lain. Hasilnya, sistem ini bisa mencapai tingkat akurasi yang tinggi. Penjelasan ini memberi pemahaman tentang manfaat dan potensi sistem SkyNet untuk berbagai aplikasi terkait langit dan pemrosesan gambar.

3.3.3 Tampilan upload dari galeri



Gambar 3. 6 Rancangan antarmuka laman input gambar

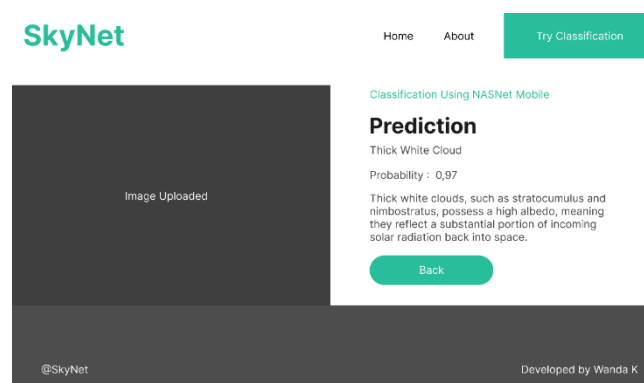
Pada rancangan gambar 3.5 diatas, terdapat judul utama *Classification SkyNet* yang

menjelaskan fungsi utama halaman ini, yaitu klasifikasi citra langit. Pengguna diarahkan ke bagian tengah halaman yang terbagi menjadi dua kolom. Kolom kiri, dengan latar belakang abu-abu tua, berfungsi sebagai area pratinjau, di mana gambar yang diunggah oleh pengguna akan ditampilkan setelah proses unggah selesai. Saat belum ada gambar yang diunggah, area ini hanya menampilkan teks *Image Uploaded* sebagai indikator tempat tampilan gambar nantinya.

Kolom kanan berisi panel *input* untuk memulai proses klasifikasi. Teks *Classification Using NASNet Mobile* menjelaskan bahwa model yang digunakan adalah *NASNet Mobile*. Di bawahnya, terdapat judul *Input Sky Image* yang diikuti oleh keterangan format file yang diterima, yaitu *PNG* dan *JPG*. Kotak *input* di bawahnya menyediakan area bagi pengguna untuk memilih file gambar dari perangkat mereka, dengan tombol *Submit to classify* di bawahnya untuk memulai proses *klasifikasi* setelah gambar dipilih. Tombol ini berwarna hijau, menarik perhatian dan memberi kesan siap diklik. Tata letak yang minimalis ini membuat alur interaksi pengguna menjadi sederhana dan intuitif, memberikan pengalaman yang mulus dalam mengunggah serta mengklasifikasi gambar langit.

3.3.4 Rancangan tampilan hasil klasifikasi

Antarmuka halaman ini menampilkan hasil prediksi klasifikasi citra langit yang telah diunggah oleh pengguna. Pada bagian atas, terdapat header dengan logo aplikasi *SkyNet* di sebelah kiri serta menu navigasi seperti *Home*, *About*, dan *Try Classification* di sebelah kanan, memberikan kemudahan akses ke berbagai halaman dalam aplikasi.



Gambar 3. 7 Rancangan antarmuka hasil klasifikasi

Bagian utama halaman ini terbagi menjadi dua kolom. Di kolom kiri dengan latar belakang abu-abu tua, gambar yang diunggah oleh pengguna akan tetap

ditampilkan sebagai pratinjau dengan teks “*Image Uploaded.*” Kolom ini berfungsi sebagai tempat gambar yang diproses.

Kolom kanan berisi informasi hasil klasifikasi dengan teks “*Classification Using NASNet Mobile*” yang menunjukkan model yang digunakan untuk klasifikasi gambar langit. Judul utama “*Prediction*” muncul diikuti dengan label hasil klasifikasi, dalam contoh ini “*Thick White Cloud.*” Di bawahnya, ditampilkan nilai *probability* sebesar 0,97, yang menunjukkan tingkat kepastian model terhadap prediksi ini. Teks deskriptif di bawahnya menjelaskan karakteristik awan putih tebal seperti *stratocumulus* dan *nimbostratus*, yang memiliki albedo tinggi, artinya mereka memantulkan sebagian besar radiasi matahari kembali ke angkasa.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Berikut ini adalah konfigurasi sistem yang diperlukan untuk sistem ini:

Perangkat keras yang digunakan untuk membangun sistem ini yaitu:

Tabel 4. 1 Spesifikasi perangkat keras

No	Komponen	Spesifikasi
1	Laptop	Lenovo Ideapad S340
2	Processor	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
3	RAM	8GB
4	Monitor	LED 14"

Sedangkan perangkat lunak yang digunakan untuk sistem ini yaitu:

Tabel 4. 2 Spesifikasi perangkat lunak

No	Komponen	Aplikasi
1	Sistem Operasi	Windows 11 Proffesional 64 bit
2	<i>Programming Language</i>	Python 3.9.19
3	<i>Text Editor</i>	Visual Studio Code
4	<i>Library</i>	Keras, Numpy, tensorflow, matplotlib, sklearn

4.2. Implementasi Data

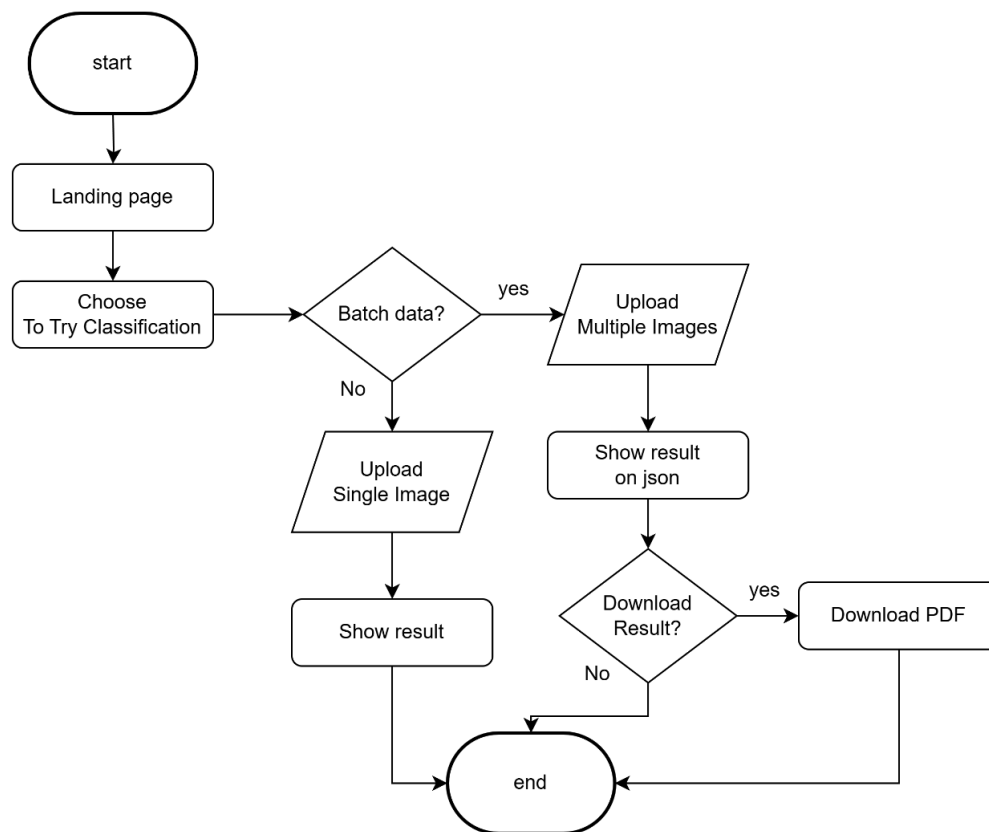
Implementasi data yang dikumpulkan pada penelitian ini diperoleh dari jurnal dengan izin penggunaan, nama dataset ini SWIMCAT (*Singapore Wholesky Imaging Categories*). Jumlah data SWIMCAT adalah 782 dengan pembagian *class* yaitu *clear sky*, *thick dark clouds*, *thick white clouds*, *patterned cloud* dan *veil cloud*. Data dipisahkan menjadi 3 bagian *train data* , *test data* dan *validation data* yang masing-masing berjumlah 555, 159, dan 62.

Untuk pengujian pertama dilakukan tanpa menggunakan metode on-the-fly augmentation data, dan pengujian selanjut akan mengimplementasikan metode

tersebut untuk melihat perbandingan dari evaluasi hasil model. Dengan augmentasi yang dilakukan saat proses pelatihan, model dapat belajar dari data yang lebih beragam tanpa memerlukan penyimpanan tambahan untuk setiap variasi. Hal ini diharapkan dapat mengurangi *overfitting* dan meningkatkan kemampuan generalisasi model ketika dihadapkan dengan data baru. Melalui pengujian ini, perbandingan akurasi dan performa antara model yang menggunakan *on-the-fly augmentation* dan model yang tidak menggunakan teknik ini akan menunjukkan sejauh mana teknik augmentasi dapat memengaruhi hasil klasifikasi citra langit.

4.3. Flowchart website

Flowchart menggambarkan alur proses kerja dari sebuah sistem klasifikasi gambar, mulai dari awal hingga proses selesai seperti pada gambar 4.1 dibawah ini.



Gambar 4. 1 Flowchart sistem

Proses dimulai dari tahap awal (start), di mana pengguna diarahkan ke halaman utama (landing page) sistem. Pada halaman ini, pengguna diberikan opsi untuk mencoba fitur klasifikasi gambar yang tersedia. Setelah memilih untuk menggunakan fitur klasifikasi, pengguna diarahkan pada tahapan pengambilan keputusan untuk menentukan apakah

akan melakukan klasifikasi pada data batch (beberapa gambar sekaligus) atau hanya pada satu gambar saja.

Jika pengguna memilih untuk melakukan klasifikasi secara batch, mereka diarahkan ke langkah *Upload Multiple Images*, di mana mereka dapat mengunggah beberapa gambar sekaligus. Setelah proses unggahan selesai, sistem akan memproses gambar-gambar tersebut dan menampilkan hasil klasifikasi dalam format JSON. Selanjutnya, pengguna diberikan opsi untuk mengunduh hasil klasifikasi tersebut. Jika pengguna memilih untuk mengunduh hasil, sistem akan menyediakan file dalam format PDF.

Sebaliknya, jika pengguna memilih untuk melakukan klasifikasi pada satu gambar saja, mereka diarahkan ke langkah *Upload Single Image*. Pada tahap ini, pengguna mengunggah satu gambar yang ingin diklasifikasi. Setelah proses unggahan selesai, sistem akan memproses gambar tersebut dan menampilkan hasil klasifikasi langsung pada layar.

Proses berakhir setelah pengguna melihat hasil atau mengunduhnya, sehingga alur kerja sistem ini dinyatakan selesai. Flowchart ini memberikan gambaran yang jelas tentang bagaimana sistem bekerja, mulai dari interaksi pengguna dengan antarmuka hingga keluaran yang diberikan sistem, baik untuk data batch maupun data tunggal. Alur ini juga menunjukkan fleksibilitas sistem dalam menangani berbagai kebutuhan pengguna.

4.4. Implementasi Antarmuka

Pengembangan tampilan antarmuka disesuaikan dengan rancangan yang telah dibuat sebelumnya. Antarmuka dibangun dalam bentuk website sehingga dapat diakses oleh perangkat sistem melalui *browser*. Sistem ini dibangun dengan *front end stack* diantaranya HTML dan CSS. Sedangkan untuk back end menggunakan *Flask* yaitu *framework back end* minimalis yang bekerja menggunakan bahasa python. Flask biasa digunakan untuk membangun website sederhana karena penggunaannya yang mudah dan ringan.

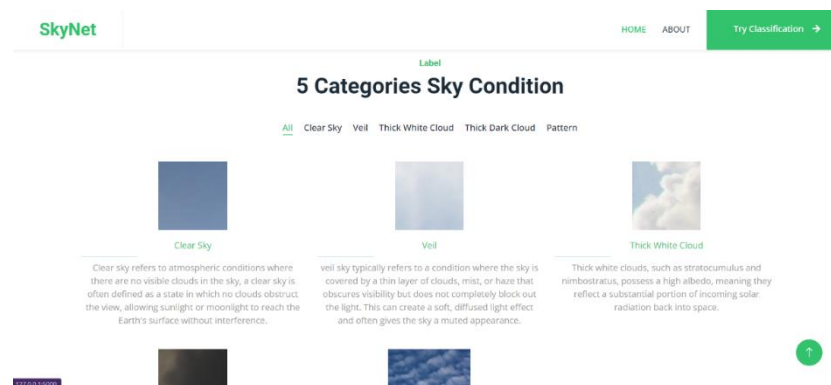
4.4.1. Tampilan *landing page*

Pada halaman *landing page* ini dirancang dengan tampilan sederhana dan profesional untuk memperkenalkan aplikasi klasifikasi kondisi langit berbasis *Convolutional Neural Network (CNN)* menggunakan arsitektur *NASNet*.



Gambar 4. 2 Hasil tampilan *landing page*

Judul besar yang menonjol dan deskripsi singkat membantu pengguna memahami tujuan aplikasi secara cepat. Tombol *Try Classification* yang terlihat jelas memudahkan akses untuk memulai proses klasifikasi. Latar belakang gambar turbin angin di bawah langit berawan menambah relevansi visual dengan topik, sementara ikon navigasi visual menambah dinamika tanpa mengganggu konten utama, menciptakan pengalaman pengguna yang informatif dan menarik.



Gambar 4. 3 Hasil tampilan antarmuka kategori citra langit

Bagian kategori citra langit diatas memberikan pengalaman yang informatif bagi pengguna dalam memahami lima kategori kondisi langit yang tersedia, yaitu *Clear Sky*, *Veil*, *Thick White Cloud*, *Thick Dark Cloud*, dan *Pattern*. Setiap kategori dilengkapi dengan gambar visual yang representatif dan deskripsi singkat yang menjelaskan karakteristik unik dari masing-masing tipe langit. Ini membantu pengguna untuk mengenali dan membedakan kondisi langit dengan lebih mudah. Penempatan tab

di bagian atas yang memungkinkan pengguna untuk memilih kategori secara spesifik, atau menampilkan semua kategori sekaligus, memberikan fleksibilitas dalam penelusuran.

4.4.2. Tampilan halaman *about*

Halaman *About* pada gambar 4.4 dirancang untuk memberikan gambaran lengkap tentang fokus dan tujuan aplikasi *SkyNet*, yaitu mengembangkan sistem klasifikasi citra langit menggunakan algoritma *Convolutional Neural Network (CNN)* dengan arsitektur *NASNet*.

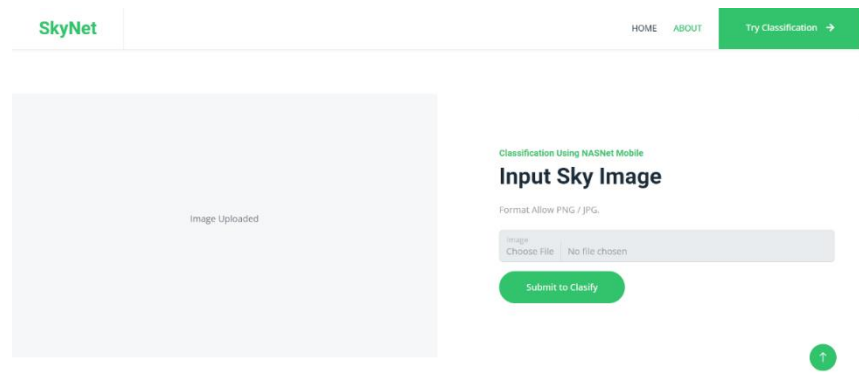


Gambar 4. 4 Hasil tampilan antarmuka halaman about

Pengguna disambut dengan teks yang besar dan jelas di sisi kanan, yang langsung menjelaskan inti penelitian ini. Deskripsi yang lebih rinci di bawahnya mencakup fenomena visual langit dan penjelasan mengenai *dataset* yang digunakan, yaitu *Singapore Whole Sky Imaging Categorizations Database (SWIMCAT)*, serta kategori-kategori citra langit yang diklasifikasikan. Penggunaan gambar teknisi dengan panel surya sebagai latar belakang memberikan konteks visual yang relevan, menunjukkan aplikasi teknologi terhadap lingkungan nyata.

4.4.3. Tampilan halaman *upload*

Pada halaman ini, menampilkan bagian "*Image Uploaded*" di sebelah kiri berfungsi sebagai area pratinjau yang akan menampilkan gambar yang diunggah oleh pengguna. Sehingga pengguna dapat memastikan apakah data yang diupload sudah sesuai dengan yang diinginkan atau belum. Tampilan rancangan antarmuka ini dapat diperhatikan pada gambar 4.5.

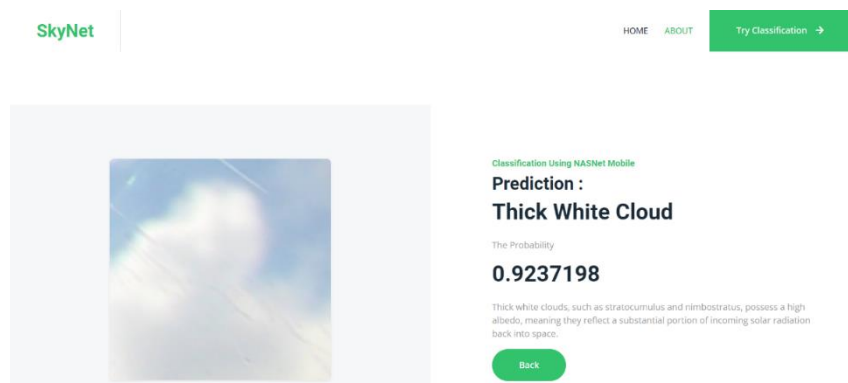


Gambar 4. 5 Hasil tampilan halaman *upload* citra

Setelah pengguna memilih dan mengunggah gambar melalui tombol *Choose File* di sebelah kanan, gambar tersebut akan muncul di area ini, memungkinkan pengguna untuk melihat pratinjau sebelum memproses klasifikasi dengan tombol "*Submit to Classify*".

4.4.4. Tampilan halaman *hasil*

Halaman ini menampilkan hasil klasifikasi gambar langit yang telah diunggah. Di sebelah kiri, terdapat pratinjau gambar langit yang telah diklasifikasikan seperti pada gambar berikut.

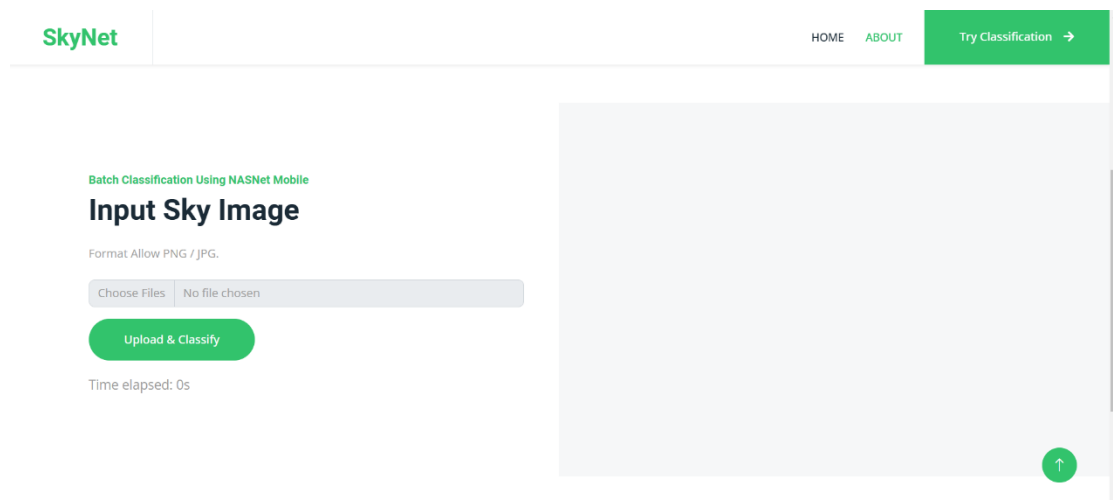


Gambar 4. 6 Hasil halaman klasifikasi

Di sebelah kanan, ditampilkan hasil prediksi dengan label "*Thick White Cloud*" (awan putih tebal) dan tingkat probabilitas sebesar 0.9237198. Selain itu, terdapat deskripsi singkat tentang jenis awan tebal ini, misalnya *stratocumulus* dan *nimbostratus*, yang memiliki albedo tinggi dan mampu memantulkan sebagian besar radiasi matahari kembali ke luar angkasa. Di bagian bawah, terdapat tombol "Back" untuk kembali ke halaman sebelumnya.

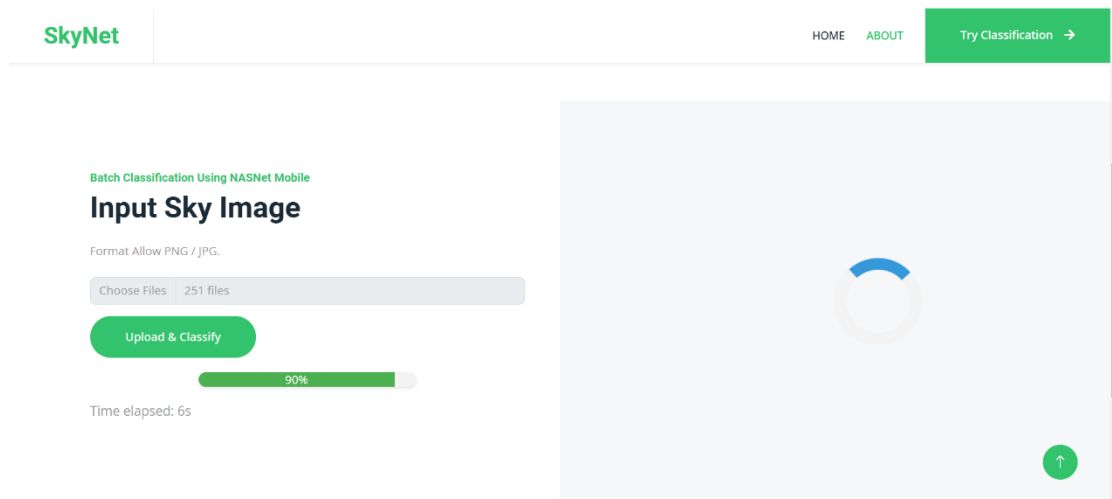
4.4.5. Tampilan halaman *upload* klasifikasi *batch data*

Pada halaman ini, terdapat formulir unggah gambar dalam bentuk batch atau banyak data yang memungkinkan pengguna memilih lebih dari 1 file berformat PNG atau JPG dengan tombol "Choose Files". Setelah memilih beberapa file, pengguna dapat memulai proses klasifikasi dengan mengklik tombol hijau "Upload & Classify". Selain itu, di bagian bawah, terdapat indikator waktu yang menunjukkan durasi proses klasifikasi, dimulai dari "Time elapsed: 0s". Desain antarmuka tersebut dapat diamati pada gambar 4.7 dibawah ini.



Gambar 4. 7 Halaman *upload batch data*

Selanjutnya pada gambar 4.8 akan menunjukkan antarmuka website SkyNet yang sedang memproses klasifikasi gambar langit. Pada tahap ini, pengguna telah mengunggah sejumlah gambar (dalam kasus ini, 251 file) melalui formulir unggahan, dan proses klasifikasi telah berjalan. Progres proses ditampilkan dalam bentuk bar hijau yang menunjukkan persentase, yang pada gambar ini mencapai 90%. Di bawahnya terdapat indikator waktu yang mencatat durasi pemrosesan, yaitu 6 detik. Di sisi kanan, sebuah animasi loader berbentuk lingkaran biru memperkuat visualisasi bahwa sistem sedang bekerja.



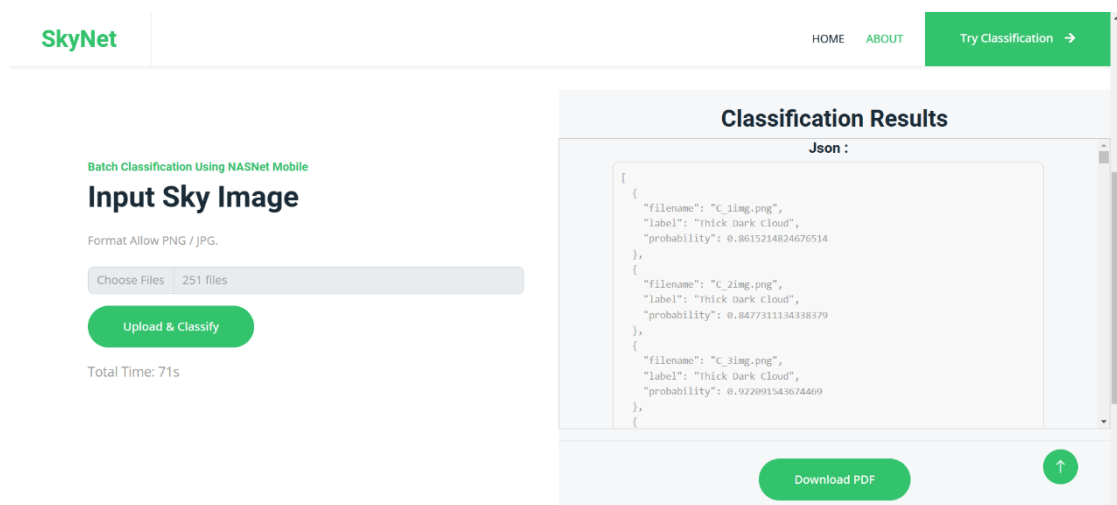
Gambar 4. 8 Tampilan progres klasifikasi

Antarmuka ini dirancang agar pengguna dapat memantau status klasifikasi dengan jelas dan intuitif. Dengan adanya indikator progres dan waktu, pengguna dapat mengetahui efisiensi sistem serta memastikan bahwa proses berjalan sesuai harapan. Visualisasi ini memberikan pengalaman pengguna yang lebih transparan dan interaktif.

4.4.6. Tampilan hasil klasifikasi *batch data*

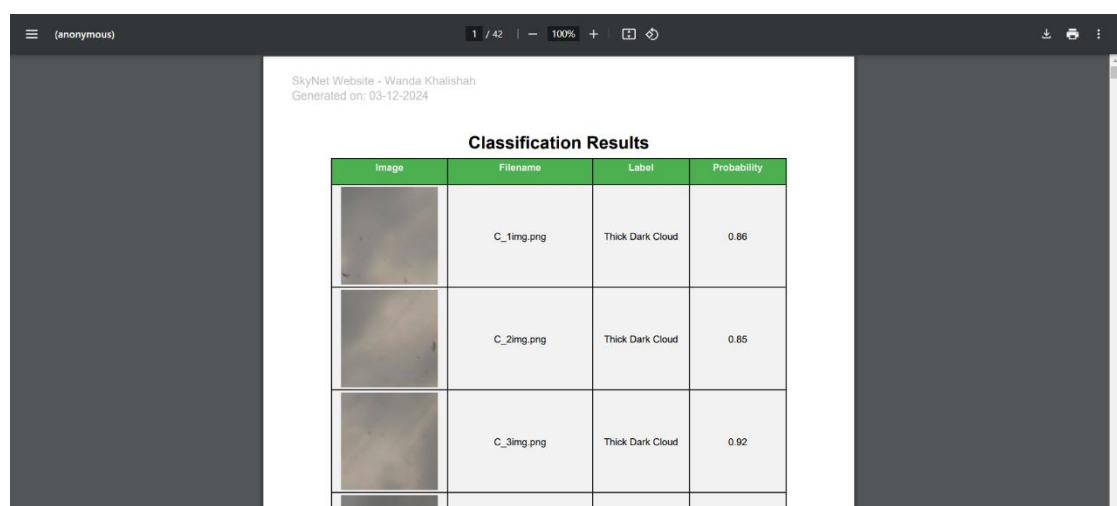
Bagian hasil yang dapat dilihat pada gambar 4.9 dibawah ini, menunjukkan output klasifikasi gambar dalam format *JSON*, yang mencakup informasi seperti nama file (*filename*), kategori hasil klasifikasi (*label*), dan tingkat kepercayaan model terhadap prediksi tersebut (*probability*). Sebagai contoh, gambar bernama "*C_img.png*" diklasifikasikan sebagai "*Thick Dark Cloud*" dengan probabilitas 86,15%. Format *JSON* digunakan karena sifatnya yang terstruktur dan mudah diproses oleh sistem lain, baik untuk kebutuhan analisis lebih lanjut, integrasi dengan aplikasi lain, atau penyimpanan data. Dengan format ini, pengguna dapat dengan mudah memanfaatkan hasil klasifikasi untuk otomatisasi atau pelaporan. Adapun bagian download pdf untuk mengunduh hasil klasifikasi secara

keseluruhan.



Gambar 4. 9 Tampilan hasil klasifikasi *batch data*

Bagian hasil unduhan PDF dapat dilihat pada gambar 4.9 dibawah ini yang menampilkan laporan klasifikasi gambar dalam bentuk tabel yang terstruktur, mencakup beberapa kolom utama: *Image*, yang menampilkan *thumbnail* dari setiap gambar untuk mempermudah identifikasi visual; *Filename*, yaitu nama file gambar seperti "*C_1img.png*"; *Label*, yang menunjukkan kategori hasil klasifikasi seperti "*Thick Dark Cloud*", dan *Probability*, yaitu tingkat kepercayaan model dalam bentuk angka desimal, seperti 0.86 (86%). Format laporan ini dirancang agar mudah dibaca dan informatif, memberikan gambaran jelas tentang hasil klasifikasi setiap gambar



Gambar 4. 10 Hasil file pdf dari klasifikasi *batch data*

Dengan fitur unduhan PDF ini, pengguna dapat menyimpan hasil klasifikasi dalam format yang lebih terorganisasi, memudahkan dokumentasi, berbagi, atau analisis lebih lanjut. Laporan ini juga mencerminkan transparansi

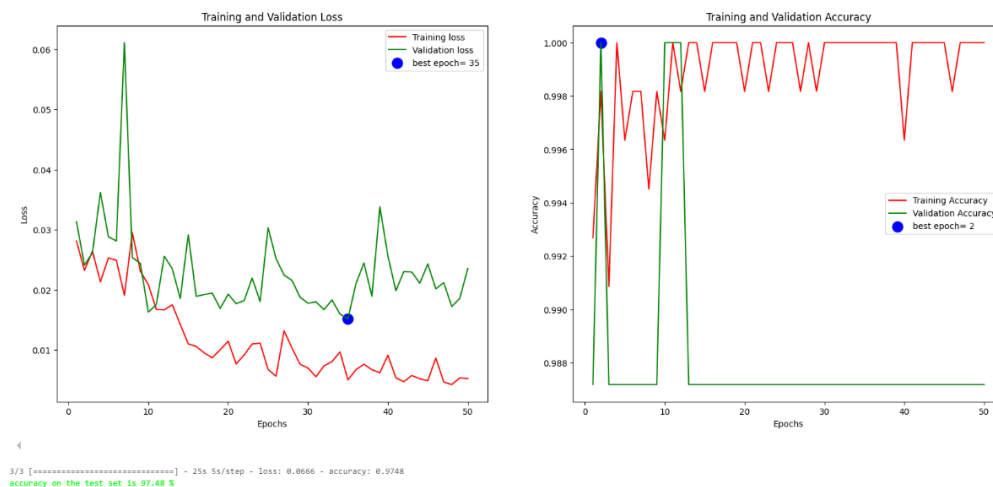
dan akurasi sistem, sehingga pengguna dapat memanfaatkan data dengan lebih optimal.

4.5. Pelatihan Sistem

Proses training menjadi tahap paling menentukan hasil dari program, dimana pada tahap ini semua data training yang telah dikumpulkan akan melewati proses pelatihan model. Beberapa parameter akan di atur untuk memperoleh trained model dengan akurasi terbaik untuk implementasi selanjutnya. Proses *training* ini akan membandingkan penggunaan *on-the-fly data augmentation* pada data yang sama. Kedua percobaan menggunakan struktur model yang sama yang dapat dilihat pada gambar 3.2.

4.5.1. Training tanpa data augmentation

Dengan menggunakan struktur model pada gambar 3.2, data train melalui proses pelatihan tanpa menggunakan data augmentasi terlebih dahulu. Batch size yang digunakan adalah 32 dan melakukan epoch sebanyak 50. Hasil dari proses training ini dapat diamati dari bagan grafik dibawah ini.



Gambar 4. 11 Grafik *loss* dan *acc* pada training tanpa augmentasi

Hasil klasifikasi model NASNet tanpa *data augmentation* dengan model dilatih menggunakan optimizer Adam dengan *learning rate* 0.0001, *batch size* 32, dan 50 *epochs*. Pada grafik *training* dan *validation loss*, terlihat bahwa *training loss* menurun dengan konsisten, menunjukkan bahwa model mampu belajar dari data pelatihan. Namun, *validation loss* yang fluktuatif dan kadang lebih tinggi daripada *training loss* menandakan bahwa model mungkin mengalami *overfitting*, yaitu kondisi di mana model terlalu cocok pada data pelatihan namun kurang mampu mengenali pola pada data baru. Hal ini semakin diperkuat dengan grafik *training* dan *validation accuracy*, di

mana *training accuracy* mencapai hampir 100% sejak awal, tetapi *validation accuracy* menunjukkan ketidakkonsistenan dan bahkan penurunan yang tajam pada beberapa *epoch*. Meskipun model ini mencapai akurasi set tes yang tinggi, yaitu 97.48%, hasil ini tidak selalu menjamin kemampuan generalisasi model karena adanya indikasi *overfitting*.

Untuk mengatasi masalah *overfitting*, penerapan *on-the-fly data augmentation* bisa menjadi solusi efektif. *On-the-fly data augmentation* memungkinkan variasi data diterapkan secara dinamis selama proses pelatihan, sehingga setiap *batch* yang dilatih model memiliki variasi gambar yang berbeda-beda. Beberapa teknik yang bisa diterapkan meliputi rotasi acak, pemotongan (*cropping*), pembalikan horizontal atau vertikal, perubahan kecerahan, kontras, serta zoom in atau out. Pendekatan ini akan membuat model lebih mampu mengenali berbagai bentuk dan skala dari objek yang sama, yang pada akhirnya membantu model belajar pola yang lebih umum dan mengurangi risiko *overfitting*. Maka pada percobaan kedua akan dilakukan proses *data augmentation* dengan metode *on-the-fly* menggunakan fungsi *ImageDataGenerator* pada *Keras*.

4.5.2. Training menggunakan *on-the-fly data augmentation*

Selanjutnya dilakukan eksperimen *hypertuning* parameter pada model *neural network* dengan menggunakan dua *optimizer* yang berbeda, yaitu *RMSprop* dan *Adam*. Pengujian dilakukan dengan memvariasikan parameter *batch size* dan *epoch*, serta mencatat hasil perhitungan *train loss*, *accuracy*, dan waktu pelatihan (*training time*). Penelitian ini bertujuan untuk mengevaluasi pengaruh kombinasi *batch size* dan *epoch* terhadap performa model, baik dalam hal akurasi maupun efisiensi waktu pelatihan, dengan membandingkan kinerja kedua algoritma optimasi tersebut. Dengan demikian, analisis tabel ini diharapkan dapat memberikan wawasan lebih mendalam tentang bagaimana pengaturan parameter dapat memengaruhi hasil pelatihan *neural network* secara keseluruhan.

Tabel 4. 3 Hasil pelatihan menggunakan arsitektur NASNet dengan dataset SWIMCAT

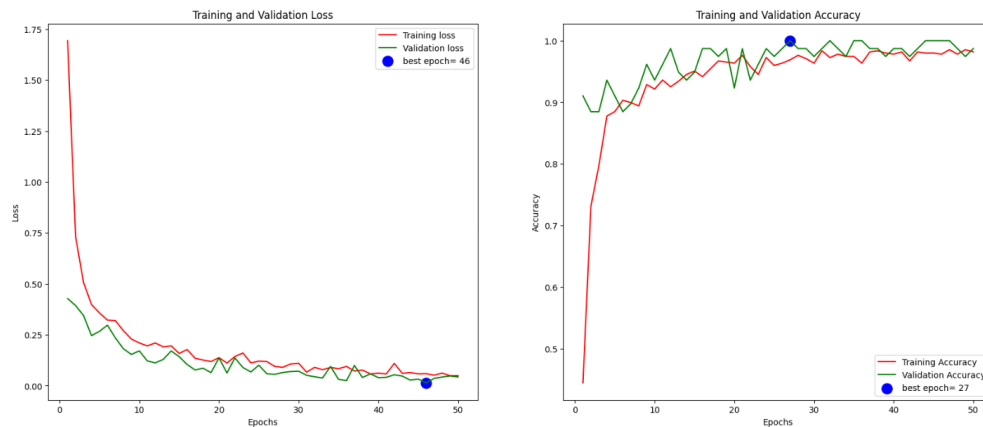
<i>Optimizer</i>	<i>Batch</i>	<i>Epoch</i>	<i>Train/loss</i>	<i>Accuracy</i>
<i>RMSprop</i>	32	30	0.0759	0.9811
<i>RMSprop</i>	32	50	0.0706	0.9821
<i>RMSprop</i>	64	30	0.048	0.9748
<i>RMSprop</i>	64	50	0.0976	0.9623
<i>RMSprop</i>	64	100	0.0221	0.9874
<i>Adam</i>	32	30	0.1067	0.9560
<i>Adam</i>	32	50	0.0453	0.9937
<i>Adam</i>	64	30	0.0662	0.9874

Pertama, terlihat bahwa *RMSprop*, yang merupakan algoritma optimasi yang secara khusus bekerja dengan baik pada data berisik atau skenario yang memerlukan pergerakan adaptif, menunjukkan akurasi yang tinggi namun cenderung lebih lambat dibandingkan *Adam* pada beberapa skenario tertentu. Misalnya, dengan *batch size* 64 dan *epoch* 100, *RMSprop* mencapai akurasi tertinggi sebesar 0.9874 dengan nilai *train/loss* sangat rendah di 0.0221, tetapi waktu pelatihannya memerlukan 1 jam 20 menit, menunjukkan bahwa *RMSprop* cenderung membutuhkan waktu lebih lama untuk mencapai konvergensi optimal.

Di sisi lain, *Adam* (*Adaptive Moment Estimation*) menunjukkan keunggulan dalam efisiensi waktu dan stabilitas performa. Pada *batch size* 32 dan *epoch* 50, *Adam* menghasilkan *train/loss* yang lebih rendah (0.0453) dan akurasi tertinggi pada pengaturan ini, yaitu 0.9937, dengan waktu pelatihan yang lebih cepat, yaitu 47 menit. Ini menunjukkan bahwa *Adam* mampu mencapai konvergensi yang optimal lebih cepat tanpa harus mengorbankan akurasi yang signifikan. *Adam* sangat cocok dalam menangani *gradien* yang lebih kompleks dan fluktuatif karena menggabungkan keunggulan *momentum* dari *RMSprop* dengan akumulasi *gradien*.

Dengan demikian, jika efisiensi waktu pelatihan menjadi prioritas utama, algoritma *Adam* merupakan pilihan yang lebih unggul dibandingkan algoritma lain. Hal ini terutama karena *Adam* mampu mencapai keseimbangan yang optimal antara akurasi yang tinggi dan waktu pelatihan yang lebih singkat. Dibandingkan dengan *RMSprop*, *Adam* tidak hanya lebih efisien dalam hal durasi pelatihan, tetapi juga

cenderung menghasilkan performa yang lebih konsisten. Sebaliknya, *RMSprop* membutuhkan waktu pelatihan yang relatif lebih lama tanpa memberikan peningkatan signifikan terhadap akurasi, sehingga kurang ideal untuk situasi di mana efisiensi dan hasil optimal diperlukan secara bersamaan.



Gambar 4. 12 Grafik loss dan acc training setelah data augmentasi

Grafik di atas menggambarkan hasil pelatihan dan validasi dari model CNN menggunakan arsitektur *NASNet Mobile*. Pada grafik pertama, terlihat perbandingan antara *training loss* dan *validation loss* sepanjang 50 epoch. Nilai *loss* yang lebih rendah menunjukkan prediksi model yang lebih akurat. Di awal pelatihan, *training loss* menurun tajam, menunjukkan bahwa model dengan cepat belajar mengenali pola-pola pada data pelatihan. Penurunan ini berlanjut hingga mencapai titik stabil, yang mengindikasikan bahwa model mulai memantapkan kemampuannya dalam memprediksi data dengan baik.

Sementara itu, *validation loss* juga menunjukkan tren menurun, yang berarti model semakin mampu memprediksi data yang belum pernah dilihat sebelumnya. Meskipun kurva *validation loss* lebih fluktuatif daripada *training loss*, hal ini dapat dipahami sebagai respon model terhadap data yang lebih beragam di tahap validasi. Pada epoch ke-46, ditandai dengan titik biru, terlihat *validation loss* mencapai nilai terendahnya, menunjukkan kinerja optimal dari model pada data validasi.

Pada grafik kedua, perbandingan akurasi pelatihan dan validasi menunjukkan hasil yang serupa. Akurasi model meningkat secara signifikan di awal pelatihan dan terus bertambah secara bertahap, menandakan bahwa model berhasil mengenali lebih banyak pola dari data pelatihan. Akurasi validasi yang mengikuti pola serupa, meskipun terdapat beberapa fluktuasi, menunjukkan konsistensi model dalam menggeneralisasi pola-pola pada data yang tidak terlihat sebelumnya. Pencapaian



akurasi tertinggi pada epoch ke-27, ditandai dengan titik biru pada grafik, menandakan titik optimal di mana model memiliki akurasi validasi tertinggi.

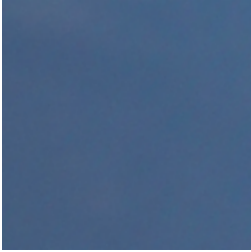
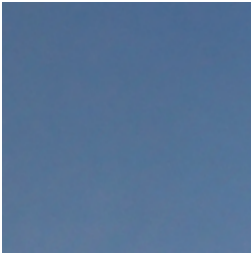

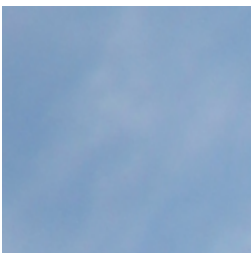

Hasil dari grafik ini menunjukkan bahwa model NASNet Mobile mampu mempelajari data pelatihan dengan baik dan memiliki performa validasi yang tinggi, yang terlihat dari akurasi validasi yang mendekati nilai maksimal. Kombinasi dari penurunan loss yang konsisten dan akurasi yang tinggi pada data validasi mengindikasikan bahwa model memiliki potensi yang baik untuk melakukan klasifikasi gambar dengan hasil yang akurat.


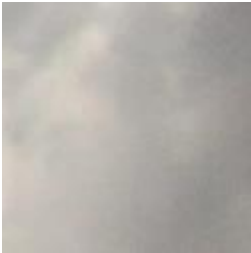


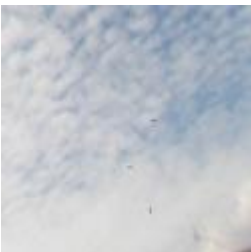
4.6. Pengujian Sistem

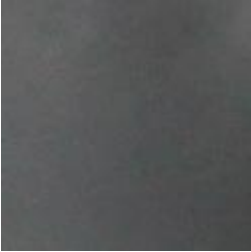




Tahap pengujian ini bertujuan untuk mengevaluasi kemampuan algoritma *NASNet* dalam melakukan klasifikasi terhadap lima jenis kategori langit. Proses pengujian dilakukan untuk menilai sejauh mana model mampu mengklasifikasikan data dengan akurat sesuai dengan kategori yang ditentukan. Pada tahap ini, model dievaluasi menggunakan *testing set* yang terdiri dari 159 data. Evaluasi dilakukan untuk mengukur performa sistem klasifikasi secara keseluruhan, termasuk akurasi dan efisiensi dalam memprediksi setiap kategori langit. Hasil evaluasi sistem ditampilkan pada Tabel 4.4 dibawah ini.


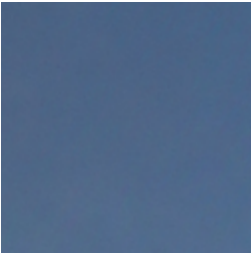
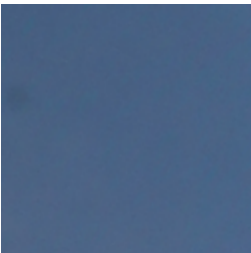
Tabel 4. 4 Uji klasifikasi

No	Gambar	Aktual	Prediksi	Status
1		<i>Thick White Cloud</i>	<i>Thick White Cloud</i>	Benar
2		<i>Thick White Cloud</i>	<i>Thick White Cloud</i>	Benar

No	Gambar	Aktual	Prediksi	Status
3		<i>Clear Sky</i>	<i>Clear Sky</i>	Benar
4		<i>Clear Sky</i>	<i>Clear Sky</i>	Benar
5		<i>Veil</i>	<i>Veil</i>	Benar
6		<i>Veil</i>	<i>Clear Sky</i>	Salah
7		<i>Pattern</i>	<i>Pattern</i>	Benar

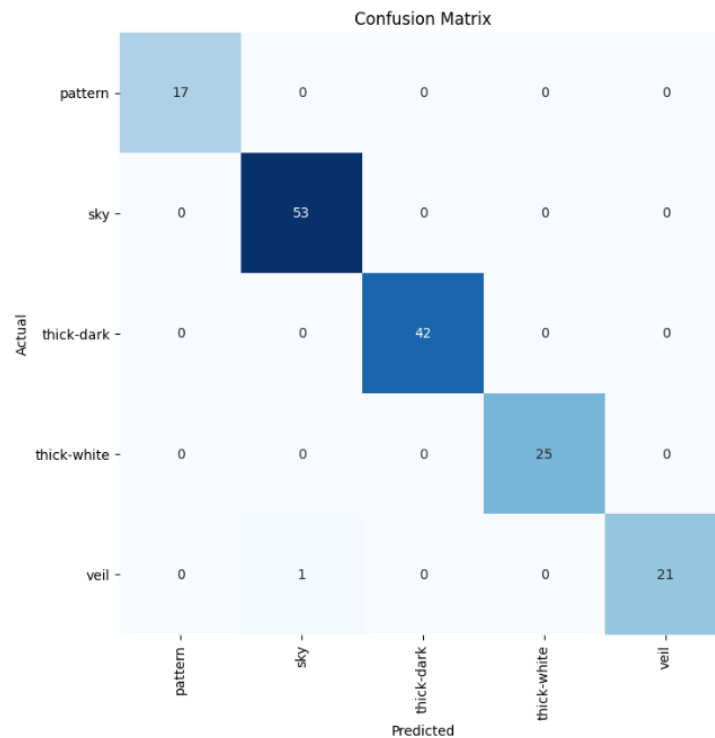
No	Gambar	Aktual	Prediksi	Status
8		<i>Pattern</i>	<i>Pattern</i>	Benar
9		<i>Thick dark cloud</i>	<i>Thick dark cloud</i>	Benar
10		<i>Thick dark cloud</i>	<i>Thick dark cloud</i>	Benar
11		<i>Pattern</i>	<i>Pattern</i>	Benar
12		<i>Pattern</i>	<i>Pattern</i>	Benar

No	Gambar	Aktual	Prediksi	Status
13		<i>Thick dark cloud</i>	<i>Thick dark cloud</i>	Benar
14		<i>Thick dark cloud</i>	<i>Thick dark cloud</i>	Benar
15		<i>Thick White Cloud</i>	<i>Thick White Cloud</i>	Benar
16		<i>Thick White Cloud</i>	<i>Thick White Cloud</i>	Benar
17		<i>Veil</i>	<i>Veil</i>	Benar

No	Gambar	Aktual	Prediksi	Status
18		<i>Veil</i>	<i>Veil</i>	Benar
19		<i>Clear Sky</i>	<i>Clear Sky</i>	Benar
20		<i>Clear Sky</i>	<i>Clear Sky</i>	Benar

Tabel 4.4 diatas merupakan sebagian data hasil uji sistem menggunakan data testing. Kesalahan prediksi pada data ke-6, yang seharusnya terklasifikasi sebagai *veil* namun diprediksi sebagai *clear sky*, bisa disebabkan oleh beberapa faktor. Dimana perbedaan visual antara *veil* dan *clear sky* sering kali sangat tipis dan sulit dikenali, terutama jika pencahayaan atau kontras pada gambar rendah. Langit cerah dan awan putih tipis dapat memiliki tekstur serta bentuk yang mirip, yang membuat model kesulitan dalam membedakan keduanya, terutama jika gambar berada dalam kondisi cahaya yang cenderung samar atau abu-abu.

Setelah melakukan pengujian pada sistem selanjutnya peneliti melakukan evaluasi berdasarkan hasil tersebut menggunakan *confusion matrix*. *Confusion matrix* dapat diperhatikan pada gambar 4.13 dibawah ini.



Gambar 4. 13 *Confusion matrix* klasifikasi menggunakan *NASNet mobile*

Confusion matrix di atas menunjukkan performa klasifikasi model *Convolutional Neural Network* (CNN) berbasis arsitektur *NASNet* untuk membedakan gambar dalam lima kategori: *pattern*, *sky*, *thick-dark*, *thick-white*, dan *veil*. Hasil ini mengindikasikan akurasi model yang sangat baik dalam mengklasifikasikan gambar pada tiap kelas dengan beberapa temuan khusus.

Pada kelas *pattern*, model berhasil mengidentifikasi 17 gambar dengan benar tanpa kesalahan klasifikasi. Hal ini menunjukkan bahwa *NASNet* mampu mengenali pola visual pada kelas *pattern* dengan sangat akurat. Selanjutnya, pada kelas *sky*, model menunjukkan performa yang baik dengan 53 prediksi yang benar dan hanya satu kesalahan, di mana gambar dari kelas *veil* diklasifikasikan sebagai *sky*. Kesalahan ini mungkin disebabkan oleh kemiripan visual antara gambar *veil* dan *sky*, yang dapat membingungkan model. Pada kelas *thick-dark*, model berhasil mengklasifikasikan 42 gambar dengan benar tanpa ada kesalahan klasifikasi, menunjukkan bahwa model memiliki kemampuan deteksi yang akurat untuk karakteristik *thick-dark*. Kelas *thick-white* juga menunjukkan hasil serupa dengan 25 prediksi yang benar tanpa adanya kesalahan klasifikasi, yang menandakan keandalan model dalam mengidentifikasi objek dengan karakteristik *thick-white*.

Pada kelas *veil*, model mengklasifikasikan 21 gambar dengan benar, tetapi terdapat satu kesalahan di mana gambar dari kelas ini diklasifikasikan sebagai *sky*. Kesalahan ini menunjukkan bahwa beberapa gambar pada kelas *veil* mungkin memiliki kemiripan karakteristik dengan *sky*, sehingga membingungkan model dalam proses klasifikasi.

Hasil klasifikasi ini menunjukkan akurasi yang tinggi dengan hanya 1 dari total 159 gambar yang diklasifikasikan salah. Tingkat akurasi ini menunjukkan bahwa model *NASNet* memiliki kinerja yang konsisten dan andal dalam mengenali pola visual di sebagian besar kelas. Meskipun demikian, potensi peningkatan dapat diarahkan pada pemisahan lebih lanjut antara kelas *veil* dan *sky*, guna mengurangi kesalahan klasifikasi yang terjadi akibat kemiripan karakteristik visual pada kedua kelas tersebut.

Tabel 4. 5 Jumlah *TP*, *FP* dan *FN*

Tipe Langit	<i>TP</i>	<i>FP</i>	<i>FN</i>
<i>Pattern</i>	17	0	0
<i>Clear Sky</i>	53	0	1
<i>Thick Dark</i>	42	0	0
<i>Thick White</i>	25	0	0
<i>Veil</i>	21	1	0
Total	158	1	1

Tabel 4.5 menunjukkan jumlah *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) untuk masing-masing tipe langit: *Pattern*, *Clear Sky*, *Thick Dark*, *Thick White*, dan *Veil*. Pada kolom TP, nilai menunjukkan jumlah prediksi yang benar untuk setiap tipe langit. Misalnya, *Sky* memiliki 53 prediksi benar, dan *Thick Dark* memiliki 42 prediksi benar.

Kolom FP (*False Positive*) menunjukkan jumlah prediksi salah di mana model mengklasifikasikan tipe langit tertentu padahal seharusnya tidak. Dalam tabel ini, hanya tipe *Veil* yang memiliki satu FP, yang berarti model salah mengenali satu gambar sebagai *Veil*.

Kolom FN (*False Negative*) menunjukkan jumlah kesalahan di mana model gagal mengidentifikasi tipe langit yang benar. Dalam tabel ini, hanya tipe *Clear Sky*

yang memiliki satu FN, menunjukkan bahwa ada satu gambar langit yang seharusnya terklasifikasi sebagai *Clear Sky*, tetapi model gagal mengenalinya dengan benar. Secara keseluruhan, dari total 159 gambar, model membuat dua kesalahan: satu *False Positive* untuk *Veil* dan satu *False Negative* untuk *Sky*, menghasilkan performa model yang cukup baik dengan hanya sedikit kesalahan. Selanjutnya untuk tahap evaluasi akan menggunakan parameter *precision*, *recall*, *F1-score* dan *accuracy* yang dapat diketahui dengan persamaan berikut ini.

4.7.1. Precision

Penentuan *precision* dilakukan dengan menggunakan persamaan 2.2 yaitu sebagai berikut.

$$Precision\ pattern = \frac{17}{17} \times 100\% = 100\%$$

$$Precision\ clear\ sky = \frac{53}{53} \times 100\% = 100\%$$

$$Precision\ thick\ white\ cloud = \frac{25}{25} \times 100\% = 100\%$$

$$Precision\ thick\ dark\ cloud = \frac{42}{42} \times 100\% = 100\%$$

$$Precision\ veil = \frac{21}{22} \times 100\% = 95\%$$

4.7.2. Recall

Penentuan *recall* dilakukan dengan menggunakan persamaan 2.3 yaitu sebagai berikut.

$$Recall\ pattern = \frac{17}{17} \times 100\% = 100\%$$

$$Recall\ clear\ sky = \frac{53}{54} \times 100\% = 98\%$$

$$Recall\ thick\ white\ cloud = \frac{25}{25} \times 100\% = 100\%$$

$$Recall\ thick\ dark\ cloud = \frac{42}{42} \times 100\% = 100\%$$

$$Recall\ veil = \frac{21}{21} \times 100\% = 100\%$$

4.7.3. F1-score

Penentuan *F1-score* dilakukan dengan menggunakan persamaan 2.4 yaitu sebagai berikut.

$$F1 - score\ pattern = 2 \frac{1 \times 1}{1+1} \times 100\% = 100\%$$

$$F1 - score\ clear\ sky = 2 \frac{1 \times 0.98}{1+0.98} \times 100\% = 98\%$$

$$F1 - score\ thick\ white\ cloud = 2 \frac{1 \times 1}{1+1} \times 100\% = 100\%$$

$$F1 - score\ thick\ dark\ cloud = 2 \frac{1 \times 1}{1+1} \times 100\% = 100\%$$

$$F1 - score\ veil = 2 \frac{0.95 \times 1}{0.95 + 1} \times 100\% = 97\%$$

4.7.4. Accuracy

Penentuan akurasi dilakukan dengan menggunakan persamaan 2.1 yaitu sebagai berikut.

$$Akurasi = \frac{158}{159} = 99,37\%$$

Berdasarkan perhitungan tersebut pada metrik *precision*, hampir semua kategori memiliki nilai 100%, kecuali kategori *veil* yang memperoleh *precision* sebesar 95%. Hal ini menunjukkan bahwa model sangat akurat dalam memprediksi sebagian besar kategori, namun terdapat sedikit ketidakakuratan pada kategori *veil*. Pada metrik *recall*, yang mengukur kemampuan model dalam mendeteksi sampel yang benar untuk setiap kategori, semua kategori mencapai nilai *recall* 100% kecuali *clear sky* dengan *recall* 98%. Ini berarti model hampir selalu berhasil mengenali semua contoh yang relevan di setiap kategori.

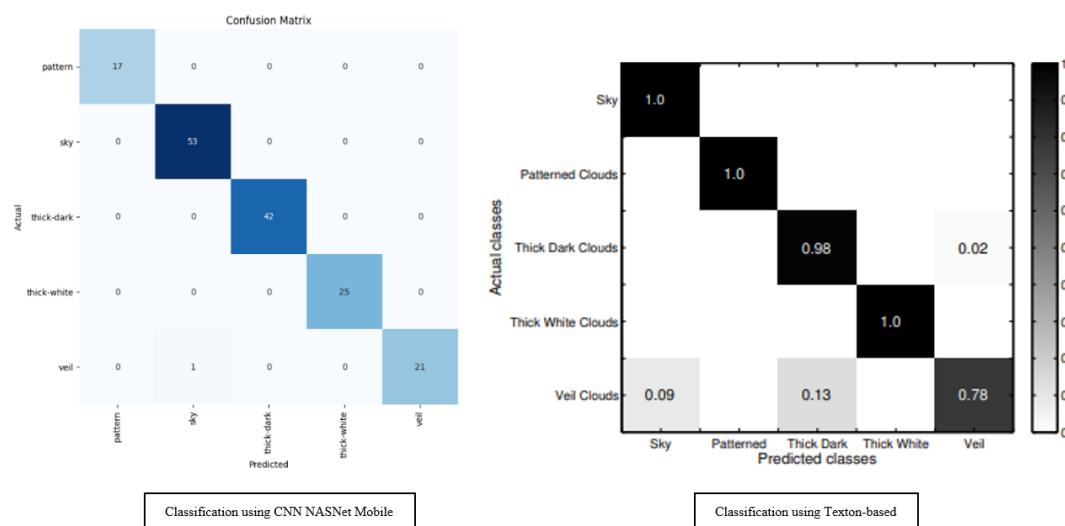
Selanjutnya, *F1-score*, yang merupakan rata-rata dari *precision* dan *recall*, menunjukkan hasil yang sangat baik dengan nilai 100% pada hampir semua kategori, kecuali *clear sky* dengan nilai 98% dan *veil* dengan nilai 97%. Hal ini mengindikasikan bahwa model memiliki keseimbangan yang baik antara ketepatan dan kemampuan deteksinya pada sebagian besar kategori. Secara keseluruhan, *accuracy* model mencapai 99,37%, yang menunjukkan bahwa model mampu mengklasifikasikan sampel dengan sangat baik.

4.7. Diskusi

Berdasarkan hasil pelatihan model *NASNet Mobile* di atas menunjukkan bahwa penggunaan *on-the-fly data augmentation* memberikan hasil yang lebih baik daripada tanpa *augmentation*. Pada gambar 4.8 grafik *loss*, model dengan *data augmentation* menunjukkan kurva yang lebih stabil dan penurunan *validation loss* yang lebih konsisten, menandakan generalisasi yang lebih baik. Sedangkan pada gambar 4.6 model tanpa *augmentation* mengalami *overfitting*, terlihat dari *validation loss* yang berfluktuasi tajam dan *validation accuracy* yang tidak stabil. *Data augmentation* membantu menambah variasi data, sehingga model lebih siap menghadapi data baru tanpa perlu penambahan data latih sebenarnya. Dengan demikian, penggunaan *data augmentation* sangat direkomendasikan untuk meningkatkan stabilitas dan akurasi model, terutama pada data yang terbatas.

Selanjutnya diketahui hasil pengujian sistem menunjukkan akurasi yang sangat tinggi, membuktikan bahwa penggunaan CNN dengan arsitektur NASNet mampu meningkatkan performa klasifikasi secara signifikan. Temuan ini konsisten dengan beberapa penelitian terbaru yang telah dilakukan. Sebagai contoh, penelitian oleh Adedoja et al. pada tahun 2022 berhasil mencapai akurasi tinggi, yaitu 99,3%, dalam klasifikasi penyakit pada daun tumbuhan. Penelitian ini memanfaatkan keunggulan NASNet dalam mengenali pola kompleks pada citra daun yang telah diproses secara optimal. Selain itu, penelitian oleh Nassar et al. pada tahun 2024 juga mendemonstrasikan efektivitas NASNet Mobile dalam klasifikasi objek, dengan akurasi mencapai 98,3%. Pendekatan mereka melibatkan optimasi parameter model dan preprocessing data yang disesuaikan dengan karakteristik dataset yang digunakan. Penelitian ini semakin memperkuat bukti bahwa NASNet merupakan arsitektur yang andal dalam menangani berbagai tugas klasifikasi, asalkan proses penentuan parameter model, teknik augmentasi, dan *preprocessing data* dilakukan secara cermat dan tepat. Hal ini menjadikan NASNet sebagai pilihan yang sangat efektif dalam aplikasi klasifikasi berbasis CNN.

Hasil pengujian pada penelitian ini dapat dibandingkan dengan penelitian sebelumnya yang menggunakan data serupa, dengan melihat table confusion matrix berikut ini.



Gambar 4. 14 Perbandingan *confussion matrix*

Gambar 4.14 menunjukkan perbandingan hasil klasifikasi kondisi langit menggunakan dua pendekatan berbeda, yaitu *CNN NASNet Mobile* dengan akurasi

99,37% (penelitian ini) dan metode berbasis *Texton* dengan akurasi 95,2% (penelitian terdahulu). Berdasarkan *confusion matrix* yang ditampilkan, *CNN NASNet Mobile* memberikan hasil yang lebih unggul dengan prediksi yang hampir sempurna untuk semua kelas. Kelas *Patterned Clouds*, *Sky*, *Thick Dark Clouds*, dan *Thick White Clouds* diprediksi dengan benar sebesar 100%, sedangkan pada kelas *Veil Clouds* hanya terdapat satu kesalahan, yaitu satu sampel yang salah diklasifikasikan sebagai *Sky*. Sementara itu, metode berbasis *Texton* juga memberikan hasil yang cukup baik pada kelas *Sky*, *Patterned Clouds*, *Thick Dark Clouds*, dan *Thick White Clouds* dengan tingkat akurasi yang tinggi, tetapi menunjukkan kelemahan pada kelas *Veil Clouds*. Model ini hanya mampu memprediksi 78% sampel kelas *Veil Clouds* dengan benar, sementara 9% salah diklasifikasikan sebagai *Sky* dan 13% sebagai *Patterned Clouds*. Kekurangan utama dari metode *Texton* adalah ketidakmampuannya dalam menangkap fitur visual yang kompleks atau mirip pada kelas-kelas tertentu, sehingga sering terjadi kesalahan pada kelas yang memiliki pola visual halus, seperti *Veil Clouds*. Dengan demikian, dapat disimpulkan bahwa *CNN NASNet Mobile* lebih unggul dalam kasus ini karena kemampuannya mengekstraksi fitur secara mendalam dan menghasilkan akurasi yang lebih tinggi dibandingkan metode berbasis *Texton*.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan dari penelitian klasifikasi citra langit menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur *NASNet* ini adalah sebagai berikut:

1. Dalam penelitian ini, telah berhasil dikembangkan menggunakan model CNN dengan arsitektur *NASNet* yang mampu mengklasifikasikan citra langit siang hari dengan akurasi yang memuaskan hingga 99.37%. Model ini mampu melakukan klasifikasi bahkan dengan jumlah data yang cukup sedikit.
2. Pemanfaatan *on-the-fly data augmentations* sangat berpengaruh dalam mencegah model *overfitting*, karena augmentasi data menghasilkan variasi gambar baru pada saat pelatihan (runtime) dengan menerapkan transformasi acak pada gambar yang ada setiap kali proses pelatihan. Oleh karena itu, augmentasi data memperkaya variasi data yang dilihat oleh model tanpa secara fisik menambah ukuran dataset pada disk.

5.2. Saran

Saran untuk penelitian selanjutnya terkait klasifikasi citra langit menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur *NASNet*:

1. Pertimbangkan perluasan dataset dengan melakukan pengumpulan dan pemrosesan citra langit yang berlokasi di Indonesia, dengan fokus pada daerah tertentu yang representatif, guna memperkaya dataset yang ada. Penambahan data citra langit dari daerah tersebut dapat memberikan informasi yang lebih spesifik dan relevan, sehingga dapat meningkatkan akurasi model dalam mengklasifikasikan kondisi langit di wilayah Indonesia.
2. Perlu dilakukan Eksperimen dengan berbagai *hyperparameter* pada model *CNN*, seperti *learning rate*, jumlah *epoch*, atau ukuran *batch*, untuk mencari kombinasi yang menghasilkan performa terbaik.
3. Membuat sistem yang *realtime* agar dapat memperoleh hasil klasifikasi secara langsung sehingga dapat dimanfaatkan untuk pengambilan keputusan yang lebih cepat.

DAFTAR PUSTAKA

- Anggreni, R., & Adriat, R. (2018). *Analisis Pengaruh Tutupan Awan Terhadap Radiasi Matahari di Kota Pontianak*. 6(3), 214–219. www.esrl.noaa.gov
- Bhadauriya Agarwal, N., & Kumar Yadav, D. (2024). A Comprehensive Analysis of Classical Machine Learning and Modern Deep Learning Methodologies. *Article in International Journal of Engineering Research*. <https://doi.org/10.17577/IJERTV13IS050275>
- Cerqueira, V., Santos, M., Baghoussi, Y., & Soares, C. (2024). *On-the-fly Data Augmentation for Forecasting with Deep Learning*. <https://github.com/vcerqueira/ondat>
- Dagan, G., Koren, I., Kostinski, A., & Altaratz, O. (2018). Organization and Oscillations in Simulated Shallow Convective Clouds. *Journal of Advances in Modeling Earth Systems*, 10(9), 2287–2299. <https://doi.org/10.1029/2018MS001416>
- Dev, S., Lee, Y. H., & Winkler, S. (n.d.). Categorization Of Cloud Image Patches Using An Improved Texton-Based Approach.
- Dev, S., Member, S., Hui Lee, Y., Member, S., & Winkler, S. (2016). Color-based Segmentation of Sky/Cloud Images From Ground-based Cameras. In *IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing: Vol. Xx*.
- Goodson, G., & Kopknit, A. (n.d.). *THE TEN DIFFERENT TYPES OF CLOUDS*.
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. <http://arxiv.org/abs/2008.05756>
- Heymsfield, A. J., Krämer, M., Luebke, A., Brown, P., Cziczo, D. J., Franklin, C., Lawson, P., Lohmann, U., McFarquhar, G., Ulanowski, Z., & Van Tricht, K. (2017). Cirrus Clouds. *Meteorological Monographs*, 58, 2.1-2.26. <https://doi.org/10.1175/amsmonographs-d-16-0010.1>
- Hinkelman, L. M., Evans, K. F., Clothiaux, E. E., Ackerman, T. P., & Stackhouse, P. W. (2007). The effect of cumulus cloud field anisotropy on domain-averaged solar fluxes and atmospheric heating rates. *Journal of the Atmospheric Sciences*, 64(10), 3499–3520. <https://doi.org/10.1175/JAS4032.1>
- Huang, W., Wang, Y., & Chen, X. (2018). Cloud detection for high-resolution remote-sensing images of urban areas using colour and edge features based on dual-colour models. *International Journal of Remote Sensing*, 39(20), 6657–6675. <https://doi.org/10.1080/01431161.2018.1466069>
- Idris, N., Feresa, C., Foozy, M., & Shamala, P. (2020). A Generic Review of Web Technology: Django and Flask. In *International Journal of Advanced Computing Science and Engineering* (Vol. 2, Issue 1).
- Javidi, Bahram. (2002). *Image recognition and classification : algorithms, systems, and applications*. Marcel Dekker.
- Khatib Sulaiman Dalam No, J., Hamad Ali, A., & Mohsin Abdulazeez, A. (n.d.). Transfer Learning In Machine Learning: A Review Of Methods And Applications. *Indonesian Journal of Computer Science*.
- Liu, S., Duan, L., Zhang, Z., Cao, X., & Durrani, T. S. (2022). Ground-Based Remote Sensing Cloud Classification via Context Graph Attention Network. *IEEE Transactions on Geoscience and Remote Sensing*, 60. <https://doi.org/10.1109/TGRS.2021.3063255>
- Lv, Q., Zhang, S., & Wang, Y. (2022). Deep Learning Model of Image Classification Using Machine Learning. *Advances in Multimedia*, 2022. <https://doi.org/10.1155/2022/3351256>
- Moughyt, S., Jaouhari, Z. El, Kadmiri, O. El, Masmoudi, L., Kadmiri, Z. El, & Zaz, Y. (2015). *Cloud coverage estimation using ground based images and segmentation techniques Image Processing: 3D medical image watermarking (Digital watermark) View project Segmentation Method Based On Multi-Objective Optimization For Very High Spatial Resolution Satellite Images View project Cloud coverage*

- estimation using ground based images and segmentation techniques.*
<https://doi.org/10.13140/RG.2.1.4040.9767>
- Naufal, M. F., & Kusuma, S. F. (2022). Weather image classification using convolutional neural network with transfer learning. *AIP Conference Proceedings*, 2470. <https://doi.org/10.1063/5.0080195>
- O, A., Adedola, A. P., & Mapayi, T. (n.d.). *Intelligent Mobile Plant Disease Diagnostic System Using NASNet-Mobile Deep Learning*.
- Öztürk, Ş., & Akdemir, B. (2018). Effects of Histopathological Image Pre-processing on Convolutional Neural Networks. *Procedia Computer Science*, 132, 396–403. <https://doi.org/10.1016/j.procs.2018.05.166>
- Parker, J. (2011). *Algorithms for Image Processing and Computer Vision Second Edition*.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Sun, X., Liu, L., & Zhao, S. (2011). Whole sky infrared remote sensing of cloud. *Procedia Earth and Planetary Science*, 2(1), 278–283. <https://doi.org/10.1016/j.proeps.2011.09.044>
- Taravat, A., Del Frate, F., Cornaro, C., & Vergari, S. (2015). Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters*, 12(3), 666–670. <https://doi.org/10.1109/LGRS.2014.2356616>
- Tran-Trung, K., Duong Thi Hong, H., & Truong Hoang, V. (2022). Weather Forecast Based on Color Cloud Image Recognition under the Combination of Local Image Descriptor and Histogram Selection. *Electronics (Switzerland)*, 11(21). <https://doi.org/10.3390/electronics11213460>
- Walecha, W. (2021). Application of Flask with Python. *International Journal of Advances in Engineering and Management (IJAEM)*, 3(6), 1665. <https://doi.org/10.35629/5252-030616651669>
- Wang, Q., & Yuan, Y. (2014). Learning to resize image. *Neurocomputing*, 131, 357–367. <https://doi.org/10.1016/j.neucom.2013.10.007>
- Yabuki, M., Shiobara, M., Nishinaka, K., & Kuji, M. (2014). Development of a cloud detection method from whole-sky color images. *Polar Science*, 8(4), 315–326. <https://doi.org/10.1016/j.polar.2014.07.004>
- Yan-Fang, X., Min, H., & Hao-Xue, L. (2008). *Influence of Rotation Processing on Digital Image's Colors*.
- Yuan, F., Wen Lek, Y., & Lee, Y. hui. (n.d.). *Correlation between Cloud Optical Thickness and Solar Radiation*.
- Zhang, C., Zhuge, X., & Yu, F. (2019). Development of a high spatiotemporal resolution cloud-type classification approach using Himawari-8 and CloudSat. *International Journal of Remote Sensing*, 40(16), 6464–6481. <https://doi.org/10.1080/01431161.2019.1594438>
- Zoph, B., & Le, Q. V. (2016). *Neural Architecture Search with Reinforcement Learning*. <http://arxiv.org/abs/1611.01578>