

**KLASIFIKASI POTENSI SARANG JENTIK NYAMUK PADA
CITRA LINGKUNGAN MENGGUNAKAN *CONVOLUTIONAL
NEURAL NETWORK (CNN) DAN DENSENET121***

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH :

SHENDY CANDRA SUKMA BUANA

32602100114

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

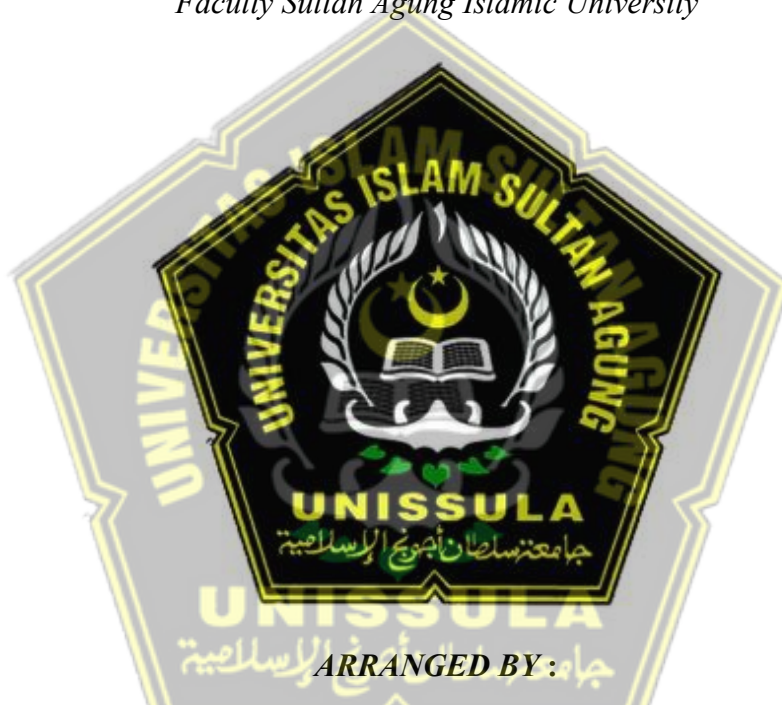
UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2026

FINAL PROJECT
CLASSIFICATION OF POTENTIAL MOSQUITO LARVA NESTS
IN ENVIRONMENTAL IMAGES USING CONVOLUTIONAL
NEURAL NETWORK (CNN) AND DENSENET121

Proposed to complete the requirement to obtain a bachelor's degree (S-1) at
Informatics Engineering Department of Industrial Technology
Faculty Sultan Agung Islamic University



SHENDY CANDRA SUKMA BUANA

32602100114

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY

2026

LEMBAR PENGESAHAN
TUGAS AKHIR

KLASIFIKASI POTENSI SARANG JENTIK NYAMUK PADA CITRA
LINGKUNGAN MENGGUNAKAN CONVOLUTIONAL NEURAL
NETWORK (CNN) DAN DENSENET121

SHENDY CANDRA SUKMA BUANA
NIM 32602100114

Telah dipertahankan di depan tim penguji proposal tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 22 Desember 2025

TIM PENGUJI UJIAN SARJANA:

Ghufron, ST, M.Kom
NIK. 210 622 056
(Ketua Penguji)

30-1-2026

Moch. Taufik, ST., MIT
NIK. 210 604 034
(Anggota Penguji)

30-1-2026

Sam Farisa Chaerul Haviana, ST,
M.Kom
NIK. 210 615 046
(Pembimbing)

30-1-2026

Semarang, 30 JANUARI 2026

Mengetahui,
Kaprodik Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIK. 210 604 034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Shendy Candra Sukma Buana
NIM : 32602100114
Judul Tugas Akhir : Klasifikasi Potensi Sarang Jentik Nyamuk Pada Citra Lingkungan Menggunakan *Convolutional Neural Network (CNN)* dan *DenseNet121*

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila dikemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 19 FEBRUARY 2026

Yang Menyatakan,



Shendy Candra Sukma Buana

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Shendy Candra Sukma Buana
NIM : 32602100114
Judul Tugas Akhir : Klasifikasi Potensi Sarang Jentik Nyamuk Pada Citra Lingkungan Menggunakan *Convolutional Neural Network (CNN)* dan *DenseNet121*
Program Studi : Teknik Informatika
Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Klasifikasi Potensi Sarang Jentik Nyamuk Pada Citra Lingkungan Menggunakan *Convolutional Neural Network (CNN)* Dan *DenseNet121*.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung

Semarang, 19 FEBRUARI 2026

Yang Menyatakan,



Shendy Candra Sukma Buana

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan Rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Klasifikasi Potensi Sarang Jentik Nyamuk Pada Citra Lingkungan Menggunakan *Convolutional Neural Network* (CNN) dan *DenseNet121*” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Allah SWT, atas segala rahmat dan karunia-Nya.
2. Rektor UNISSULA Bapak Prof Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
3. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
4. Dosen pembimbing penulis Bapak Sam Farisa Chaerul Haviana, ST, M.Kom yang telah memberikan arahan, bimbingan, dan saran yang sangat berarti dalam proses penelitian hingga penulisan laporan ini.
5. Dosen-dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah membekali penulis dengan ilmu dan pengetahuan selama masa studi.
6. Orang tua penulis, Bapak Efendi Ariyanto dan Ibu Tuning Dwi Prasasti yang selalu memberikan doa, dukungan dan motivasi selama proses penyelesaian tugas akhir ini.
7. Teman-teman dan rekan seperjuangan, yang telah memberikan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir ini.
8. Kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semarang, 19 Februari 2026



Shendy Candra Sukma Buana

DAFTAR ISI

TUGAS AKHIR	i
LEMBAR PENGESAHAN TUGAS AKHIR.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
ABSTRAK	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	8
2.2.1 Lingkungan Berpotensi Sarang Jentik Nyamuk.....	8
2.2.2 Pengolahan Citra (Image Processing)	9
2.3 Konvolusi.....	12
2.3.1 Teori Konvolusi	12
2.3.2 Proses Konvolusi pada Citra Digital.....	14
2.3.3 <i>Convolutional Neural Network</i> (CNN).....	17
2.3.4 <i>DenseNet121</i>	Error! Bookmark not defined.
BAB III METODE PENELITIAN	21
3.1 Metodologi Penelitian.....	21
3.1.1 Alur Penelitian	21

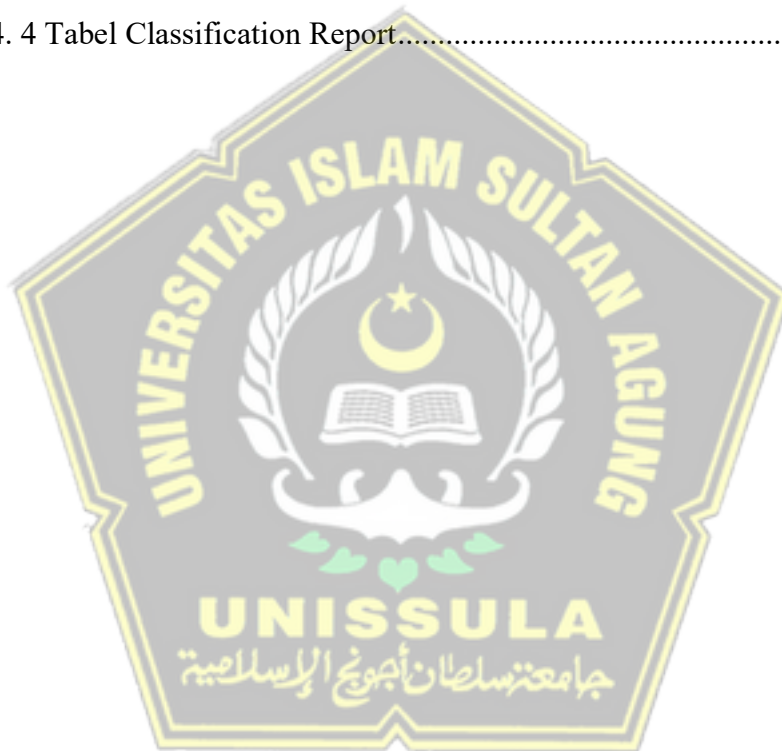
3.1.2	Pengumpulan Dataset:.....	23
3.1.3	Perancangan Model.....	24
3.1.4	Arsitektur <i>Convolutional Neural Network</i> (CNN) <i>DenseNet121</i>	28
3.1.5	Evaluasi Model.....	38
3.2	Analisa kebutuhan.....	40
3.3	Implementasi sistem.....	44
3.4	Rancangan <i>User Interface</i>	45
3.4.1	Halaman Antarmuka Utama.....	45
3.4.2	Halaman Hasil dari Sistem.....	46
BAB IV HASIL DAN ANALISIS PENELITIAN		49
4.1	Hasil	49
4.1.1	Pengumpulan Data	49
4.2	Pengujian Model	51
4.2.1	Pengolahan Awal Citra <i>Dataset</i>	51
4.2.2	Tahap Pembelajaran Model CNN berbasis <i>DenseNet121</i>	52
4.3	Evaluasi Model.....	57
4.4	Implementasi Model.....	60
4.4.1	Fitur Utama	60
4.4.2	Hasil Implementasi.....	61
4.5	Pembahasan Implementasi	65
BAB V KESIMPULAN DAN SARAN		67
5.1	Kesimpulan	67
5.2	Saran.....	67
DAFTAR PUSTAKA		68

DAFTAR GAMBAR

Gambar 2. 1 Lingkungan berpotensi nyamuk	8
Gambar 2. 2 Ilustrasi Image Processing	11
Gambar 2. 3 Matriks Konvolusi.....	12
Gambar 2. 4 Contoh operasi konvolusi.....	15
Gambar 2. 5 Hasil Operasi Konvolusi Pertama	15
Gambar 2. 6 Hasil Operasi Konvolusi Kedua	16
Gambar 2. 7 Hasil Operasi Konvolusi Ketiga.....	16
Gambar 2. 8 Hasil Operasi Konvolusi Keempat.....	17
Gambar 2. 9 Arsitektur CNN	18
Gambar 2. 10 Arsitektur <i>DenseNet</i> 121	19
Gambar 3. 1 WorkFlow Model.....	26
Gambar 3. 2 Alur Implementasi Sistem.....	44
Gambar 3. 3 Halaman Awal Sistem User Interface	46
Gambar 3. 4 Halaman Hasil Sistem	47
Gambar 3. 5 Halaman Hasil Sistem Menggunakan Grad-Cam	47
Gambar 4. 1 Arsitektur Sistem <i>DenseNet</i> 121	29
Gambar 4. 2 Contoh Gambar Setelah Augmentasi	52
Gambar 4. 3 Hasil Prediksi Data Test.....	56
Gambar 4. 4 Confusion Matrix	59
Gambar 4. 5 Tampilan Home Streamlit	61
Gambar 4. 6 Tampilan saat citra di input ke sistem.....	62
Gambar 4. 7 Tampilan hasil klasifikasi setelah upload Citra.....	64

DAFTAR TABEL

Tabel 2. 1 Bentuk efek Kernel	13
Tabel 2. 2 Tabel hasil konvolusi	17
Tabel 3. 1 Contoh Tabel Confusion Matrix	39
Tabel 4. 1 Struktur DenseNet121	37
Tabel 4. 2 Sample dataset citra lingkungan potensi nyamuk	50
Tabel 4. 3 Hyperparameter Model	52
Tabel 4. 4 Tabel Classification Report	58



ABSTRAK

Lingkungan yang memiliki genangan air sering menjadi tempat berkembang biaknya nyamuk sehingga berpotensi menimbulkan penyakit menular. Penelitian ini bertujuan mengembangkan sistem klasifikasi otomatis untuk mendeteksi potensi sarang jentik nyamuk berbasis citra lingkungan menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *DenseNet121*. *Dataset* diperoleh melalui dokumentasi lapangan dengan dua kategori, yaitu berpotensi dan tidak berpotensi sebagai sarang jentik nyamuk. Tahapan penelitian meliputi *preprocessing* citra yaitu *resize*, normalisasi, dan augmentasi, perancangan model CNN, serta penerapan *DenseNet121* dengan *transfer learning*. Hasil pengujian menunjukkan bahwa model mampu mencapai tingkat akurasi sebesar 98%, *precision* 95%, *recall* 100%, dan *F1-Score* 98%, yang membuktikan kehandalan model dalam mengenali pola visual lingkungan. Sistem juga telah diimplementasikan ke dalam antarmuka berbasis *Streamlit*, sehingga pengguna dapat mengunggah citra dan memperoleh hasil klasifikasi secara *real-time* melalui *browser*. Penelitian ini berhasil memenuhi tujuan untuk menghadirkan alat bantu berbasis kecerdasan buatan yang praktis, akurat, dan mudah diakses. Namun, pengembangan selanjutnya diperlukan melalui perluasan variasi *dataset*, peningkatan kondisi pencahayaan dan objek, serta eksplorasi model deteksi berbasis YOLO agar sistem lebih adaptif terhadap kondisi nyata di lapangan.

ABSTRACT

Environments with stagnant water often become breeding grounds for mosquitoes, which can lead to the spread of vector-borne diseases. This study aims to develop an automatic classification system to detect potential mosquito breeding sites based on environmental images using a Convolutional Neural Network (CNN) with the DenseNet121 architecture. The dataset was collected through field documentation and categorized into two classes: potential and non-potential breeding sites. The research stages include image preprocessing (resizing, normalization, and augmentation), CNN model design, and the application of DenseNet121 through transfer learning. The experimental results show that the model achieved an accuracy of 98%, precision of 95%, recall of 100%, and an F1-Score of 98%, demonstrating the reliability of the model in recognizing visual patterns of the environment. Furthermore, the system was successfully implemented into a Streamlit-based interface, enabling users to upload images and obtain classification results in real-time through a web browser. This research achieved its objective of providing a practical, accurate, and accessible AI-based tool to support early detection of mosquito breeding sites. Nevertheless, further development is recommended by expanding dataset variations, improving lighting and object diversity, and exploring YOLO-based detection models to enhance adaptability to real-world conditions.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Lingkungan tempat tinggal Masyarakat khususnya di wilayah Semarang yang padat penduduk dan Kawasan perkotaan maupun pedesaan kerap memiliki berbagai titik yang berpotensi menjadi tempat berkembangbiaknya nyamuk. Kondisi seperti genangan air di selokan, pot bunga yang tidak dikuras, kolam hias, serta area lembap yang tidak terpantau dapat menjadi tempat ideal bagi nyamuk untuk bertelur. Masalah utama bukan hanya pada keberadaan nyamuk itu sendiri, melainkan pada kesulitan dalam mengidentifikasi secara dini di lingkungan yang memiliki karakteristik mendukung berkembangbiakan nyamuk. (Ilyas dkk., 2025)

Seiring perkembangan teknologi khususnya dalam bidang pengolahan citra, kini tersedia peluang untuk melakukan klasifikasi otomatis terhadap potensi sarang jentik nyamuk yang berdasarkan citra visual dari lingkungan. Salah satu pendekatan yang efektif untuk ini adalah melalui metode *Convolutional Neural Network* (CNN) model pembelajaran *deep learning* yang unggul dalam mengenali pola dan fitur visual dalam gambar. CNN dapat mengekstraksi berbagai informasi dari citra lingkungan, seperti keberadaan air, tekstur tanah, dedaunan, atau benda-benda yang berpotensi menampung air. Salah satu arsitektur CNN yang dikenal memiliki performa tinggi adalah *DenseNet121* yang memanfaatkan koneksi secara padat untuk menghasilkan fitur yang lebih akurat (A. B. Saputra dkk., 2024). Kombinasi CNN dengan Teknik augmentasi data juga terbukti efektif dalam meningkatkan generalisasi model terhadap gambar yang bervariasi. (Toyib dkk., 2024)

Umumnya pendekatan ini dapat mendukung pemetaan wilayah-wilayah yang rawan menjadi habitat nyamuk secara cepat dan akurat, terutama di lingkungan masyarakat, kampus, atau fasilitas umum. Hal ini dapat menjadi bagian dari inisiatif digitalisasi pemantauan lingkungan dan pemberdayaan

masyarakat dalam menjaga kebersihan lingkungan secara mandiri melalui pendekatan berbasis citra.

Penelitian ini berfokus pada pengembangan sistem klasifikasi otomatis terhadap potensi sarang jentik nyamuk berdasarkan citra lingkungan. Dengan memanfaatkan CNN dengan arsitektur *DenseNet121* serta augmentasi data untuk meningkatkan akurasi model. Hasil dari penelitian ini diharapkan dapat mendukung proses pemantauan lingkungan secara digital dan efisien.

1.2 Perumusan Masalah

Berdasarkan latar belakang, rumusan masalah dalam penelitian ini adalah Bagaimana membangun model klasifikasi citra lingkungan untuk mendeteksi potensi sarang jentik nyamuk menggunakan CNN dengan arsitektur *DenseNet121*?

1.3 Pembatasan Masalah

Pembatasan masalah bertujuan untuk menghindari adanya kegiatan di luar sasaran, sehingga dalam pembuatan penelitian ini perlu ditentukan suatu batasan masalah sebagai berikut:

1. Penelitian ini hanya berfokus pada klasifikasi citra lingkungan yang menjadi dua kelas, yaitu berpotensi sarang jentik nyamuk dan tidak berpotensi sarang jentik nyamuk tanpa melakukan penandaan area atau segmentasi objek secara spesifik.
2. Data yang digunakan bersumber dari dokumentasi foto lapangan dengan kriteria visual yang ditentukan berdasarkan pengamatan umum, tanpa melibatkan pakar entomologi (ilmu yang mempelajari tentang serangga).
3. Penelitian ini menggunakan arsitektur *DenseNet121* dalam kerangka CNN sebagai model yang diuji, tanpa melakukan perbandingan dengan model lain.
4. Teknik augmentasi data yang digunakan terbatas pada transformasi sederhana seperti rotasi, flip, zoom, dan perubahan pencahayaan, tanpa menggunakan augmentasi berbasis GAN.

1.4 Tujuan

Penelitian ini bertujuan untuk membangun dan melatih model deteksi menggunakan CNN untuk mengklasifikasikan potensi sarang jentik nyamuk berdasarkan citra lingkungan.

1.5 Manfaat

Diharapkan hasil dari penelitian ini dapat memberikan manfaat bagi:

1. Kontribusi lingkungan: Memberikan kontribusi dalam Upaya menjaga Kesehatan lingkungan dengan menyediakan sistem klasifikasi otomatis yang akurat dan efisien untuk mendeteksi potensi sarang jentik nyamuk, sehingga Tindakan pencegahan dapat dilakukan secara cepat dan tepat sasaran.
2. Peneliti dan Akademisi: Menjadi referensi bagi peneliti dan akademisi di bidang Kesehatan lingkungan, biologi, dan teknologi informasi, serta sebagai bahan kajian untuk penelitian lanjutan terkait aplikasi *Deep Learning* dalam klasifikasi citra lingkungan
3. Praktisi dan Lembaga Pengelola Kesehatan Masyarakat: Membantu praktisi dan Lembaga yang bergerak di bidang pengendalian penyakit dan pengelolaan lingkungan dalam menerapkan teknologi pemantauan yang lebih baik, sehingga dapat memantau dan mengendalikan potensi sarang jentik nyamuk secara lebih efektif.
4. Masyarakat umum: Meningkatkan pemahaman Masyarakat mengenai pentingnya menjaga kebersihan lingkungan dan menghilangkan tempat-tempat yang berpotensi menjadi sarang nyamuk, serta mendorong keterlibatan aktif dalam program pencegahan penyakit berbasis lingkungan.

1.6 Sistematika Penulisan

Sistematika yang akan digunakan oleh penulis dalam pembuatan proposal tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada BAB I ini berisi latar belakang, perumusan masalah, pembatasan masalah, tujuan dan sistematika penulisan yang menjelaskan isi dari masing-masing sub-bab dalam proposal.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II membahas Tinjauan Pustaka yang berisi teori-teori dasar dan penelitian-penelitian terdahulu yang relevan dengan topik. Dalam bab ini dijelaskan konsep mengenai dasar habitat nyamuk secara umum, karakteristik citra lingkungan, pengolahan citra digital, serta teori mengenai *Convolutional Neural Network* (CNN) dan arsitektur *DenseNet* yang menjadi kasus utama dalam metode penelitian.

BAB III : METODE PENELITIAN

Pada BAB III berisi metodologi penelitian yang menjelaskan secara rinci langkah-langkah yang dilakukan dalam penelitian ini dan evaluasi performa dan efisiensi dalam klasifikasi.

BAB IV : HASIL DAN PEMBAHASAN

Pada BAB IV menyajikan hasil dari pelatihan dan pengujian model yang telah dibuat, serta analisis dan interpretasi terhadap hasil tersebut. Dalam bab ini ditampilkan visual hasil klasifikasi, grafik performa model selama proses pelatihan dan pembahasan terhadap akurasi model dalam membedakan lingkungan yang berpotensi dan tidak berpotensi menjadi sarang jentik nyamuk.

BAB V : KESIMPULAN DAN SARAN

Pada BAB V berisi Kesimpulan dari hasil penelitian yang telah dilakukan serta saran-saran yang dapat diberikan untuk pengembangan penelitian lebih lanjut.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai pemanfaatan pengolahan citra digital untuk pengklasifikasian objek lingkungan dengan memanfaatkan algoritma *Convolutional Neural Network* (CNN) telah banyak dilakukan, terutama dalam konteks pengawasan lingkungan berbasis teknologi. Penelitian sebelumnya menunjukkan bahwa penggunaan *Convolutional Neural Network* (CNN) yang dilengkapi augmentasi data mampu meningkatkan performa klasifikasi citra tulisan tangan. Dataset yang digunakan dalam pelatihan model dari penelitian ini terdiri dari 1.120 gambar untuk pengujian dan 280 gambar untuk validasi dan pelatihan model dilakukan sebanyak 300 epoch dan menunjukkan akurasi sebesar 96.07% dan *loss* sebesar 11.62%. Penelitian ini menunjukkan bahwa augmentasi data memainkan peran penting dalam memperkaya variasi data pelatihan dan mengurangi overfitting, serta dapat diterapkan pada konteks visual lain seperti klasifikasi kondisi lingkungan. (Toyib dkk., 2024)

Penelitian sebelumnya mengembangkan sistem klasifikasi citra terumbu karang dengan membandingkan dua arsitektur CNN modern. Dataset yang digunakan berasal dari platform Kaggle yang terdiri dari dua kelas yaitu *Bleached* dan *Unbleached*. Dataset kemudian dilatih melalui beberapa tahap yaitu perubahan citra, pembagian data, pelatihan model, dan pengujian model. Evaluasi model menggunakan *matrix accuracy*, *precision*, *recall*, F1-score dan *confusion matrix*. Hasilnya menunjukkan bahwa akurasi *DenseNet121* mencapai 88.06% dengan nilai *loss* sebesar 36.91%. Penelitian ini membuktikan bahwa *DenseNet121* dapat digunakan secara efektif untuk klasifikasi citra lingkungan dengan kompleksitas tinggi, terutama dalam mengidentifikasi detail visual halus seperti tekstur atau pula permukaan objek. (Utomo & Mulyono, 2025)

Penelitian sebelumnya mengembangkan sistem klasifikasi citra rontgen untuk mendeteksi pneumonia. Penggunaan optimizer seperti Adam, Adamax, Adadelta diuji pada masing masing konfigurasi selama 50 epoch. Dengan data yang berbeda dari data pelatihan, model ini mencapai akurasi sebesar 96% selama fase pengujian. Penelitian ini menunjukkan bahwa CNN dengan arsitektur *DenseNet201* efektif dalam memahami fitur tekstur dan struktur dalam citra medis, yang secara prinsip serupa dengan klasifikasi kondisi lingkungan berbasis visual.(Hermawan dan Muhammad, 2024)

Penelitian sebelumnya menggunakan *DenseNet121* untuk membedakan jenis batu ginjal berdasarkan tekstur citra CT. Data yang digunakan dan diolah dalam penelitian ini yaitu 1.101 data *training*, 110 data validasi, 110 data *testing* data tersebut untuk masing masing kelas. Model CNN dengan arsitektur *DenseNet121* yang dikembangkan mencapai akurasi 98.18%, *precision* 96.36%, *recall* 100, dan F1-Score 98.14%. Hasil ini menunjukkan bahwa *DenseNet121* memiliki kemampuan representasi fitur yang kuat meskipun pada gambar grayscale dan dapat diterapkan pada domain klasifikasi citra non-medikal seperti lingkungan alami. (Pramesti, 2024)

Penelitian sebelumnya yang menunjukkan penggunaan *DenseNet121* dalam konteks validasi keaslian citra. Dengan teknik watermarking LSB dan CNN sebagai alat deteksi fitur citra palsu, sistem mampu mencapai akurasi sebesar 91% dengan *precision* 92%, *recall* 91%, dan F1-score 91%. Meskipun fokus utama adalah autentikasi, penelitian ini menunjukkan bahwa *DenseNet121* efektif dalam mengenali detail minor pada struktur visual. (Jiwani dan Bagus, 2025)

Penelitian sebelumnya membangun sistem klasifikasi berbasis aplikasi mobile yang mampu membedakan berbagai jenis penyakit cacar berdasarkan foto. Dengan dataset sebanyak 1000 gambar dan augmentasi berupa rotasi, flipping, serta peningkatan kontras, model *DenseNet121* yang digunakan menghasilkan akurasi pelatihan sebesar 95.41%, akurasi validasi sebesar 97.20%, *loss* pelatihan 13.55%, dan *loss* validasi sebesar 8.48,

menjadikannya cocok untuk implementasi praktis dalam lingkungan nyata yang bervariasi. (Falah dan Mulyono, 2024)

Penelitian sebelumnya menunjukkan bahwa arsitektur *DenseNet121* mampu mengklasifikasikan jenis hijab secara akurat berdasarkan fitur tekstur dan warna pada citra produk. Dengan dataset hijab citra berukuran 640x225 piksel, model yang digunakan mencapai akurasi klasifikasi sebesar 89%, *precision* sebesar 89%, *recall* sebesar 89%, dan F1-score sebesar 87%. Hal ini menunjukkan bahwa *DenseNet121* dapat digunakan untuk mengidentifikasi perbedaan visual halus pada objek yang memiliki tampilan seragam. (D. L. R. Putri dan Supatman, 2025)

Penelitian sebelumnya menerapkan arsitektur CNN ini untuk mendeteksi penyakit pada tanaman berdasarkan gejala visual daun. Dataset yang digunakan terdiri dari ribuan gambar daun kopi yang dikumpulkan dari lahan pertanian di Sulawesi Selatan. Dengan teknik augmentasi seperti *rotasi*, *flipping*, dan *zoom*. Arsitektur *DenseNet121* berhasil mencapai 97%. Penelitian ini membuktikan potensinya dalam klasifikasi berbasis tekstur alami seperti daun. (Solehudin dkk., 2025)

Penelitian sebelumnya yang juga menunjukkan bahwa *DenseNet121* mampu mengenali pola penyakit yang menyerang daun tomat. Dataset terdiri dari 2.283 citra daun tomat, yang terdiri dari 3 jenis kelas prediksi diantaranya yaitu, penyakit *leaf mold*, daun tomat sehat, dan penyakit tomat lainnya yang dimasukkan kedalam kelas prediksi penyakit lainnya. Hasil penelitian ini diuji dengan 3 model uji coba, dengan nilai akurasi, *precision*, dan *recall* yang dihasilkan sebesar 92.6%, 93,3%, dan 93%. (Awalia dan Primajaya, 2022)

Penelitian sebelumnya membandingkan dua arsitektur CNN terkemuka dalam klasifikasi tumor otak menggunakan dataset MRI. Hasil evaluasi menunjukkan akurasi mencapai 96.31% ini membuktikan potensinya dalam klasifikasi berbasis tekstur alami seperti daun (F. A. P. R. Putri dkk., 2025)

2.2 Dasar Teori

2.2.1 Lingkungan Berpotensi Sarang Jentik Nyamuk

Lingkungan fisik yang sangat memengaruhi keberadaan jentik nyamuk dan kepadatan populasi nyamuk dewasa. Lingkungan seperti genangan air, empang, rawa, tambak, dan aliran sungai yang tidak mengalir merupakan tempat yang umum dijadikan lokasi perindukan oleh jentik nyamuk. Kondisi seperti air sungai yang bercampur dengan air laut membentuk genangan air payau yang mendukung pertumbuhan nyamuk. Selain itu, lingkungan biologis seperti keberadaan tumbuhan air dan hewan tertentu juga mempengaruhi kepadatan larva ditempat perindukan. (Sinum, 2022)



Gambar 2. 1 Lingkungan berpotensi nyamuk (https://www.detik.com/properti/tips-dan-panduan/d-7296867/penting-5-cara-jitu-basmi-jentik-nyamuk-di-rumah-penyebab-dbd#google_vignette)

Pada gambar 2.1 lingkungan dengan kelembapan tinggi serta adanya genangan air baik alami maupun buatan merupakan tempat yang sangat potensial menjadi sarang jentik nyamuk.

Menurut Haidy Massa (2025), habitat utama jentik nyamuk pada umumnya ditemukan pada berbagai wadah buatan yang dapat menampung air dalam jangka waktu tertentu. Wadah-wadah ini menjadi tempat ideal bagi

nyamuk untuk bertelur karena kondisi air yang relatif tenang dan seringkali tidak terganggu oleh aktifitas manusia. Selain itu, wadah kecil yang kerap diabaikan seperti ember, pot tanaman, dan ban bekas juga terbukti menjadi tempat berkembang biak jentik nyamuk yang signifikan. (Haidy Massa dkk., 2025)

Pendekatan konvensional berbasis survey lapangan dan pengamatan visual dinilai kurang efisien karena bersifat subjektif, membutuhkan waktu lama, serta memerlukan keahlian khusus untuk mengidentifikasi keberadaan jentik pada wadah penampungan air. Oleh karena itu, metode berbasis citra digital dan kecerdasan buatan mulai dikembangkan sebagai Solusi modern untuk mendeteksi habitat potensial nyamuk secara lebih cepat dan akurat. Menurut Nainggolan (2023), dengan memanfaatkan kemampuan *Convolutional Neural Network* (CNN) dalam mengekstraksi fitur visual secara otomatis, sistem klasifikasi dapat mendeteksi keberadaan jentik diberbagai jenis wadah. Hal ini menunjukkan bahwa integrasi teknologi *Deep Learning* dengan data citra lingkungan berpotensi jentik nyamuk menjadi strategi efektif untuk mendukung pengendalian vector secara presisi. (Nainggolan dkk., 2023)

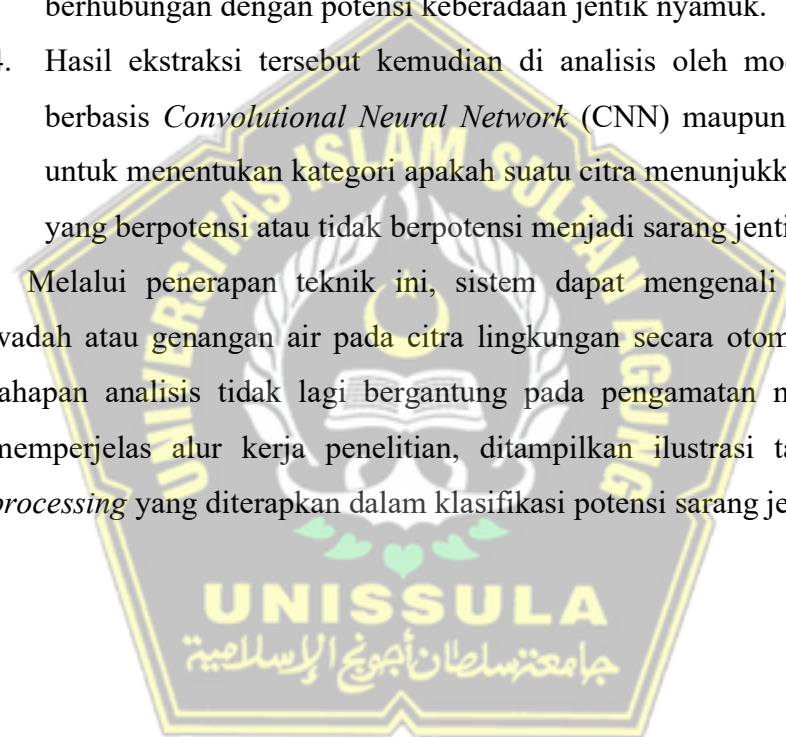
Pemahaman terhadap kondisi lingkungan ini menjadi sebagai dasar untuk pengembangan sistem klasifikasi visual berbasis citra digital guna mendeteksi potensi sarang nyamuk secara otomatis.

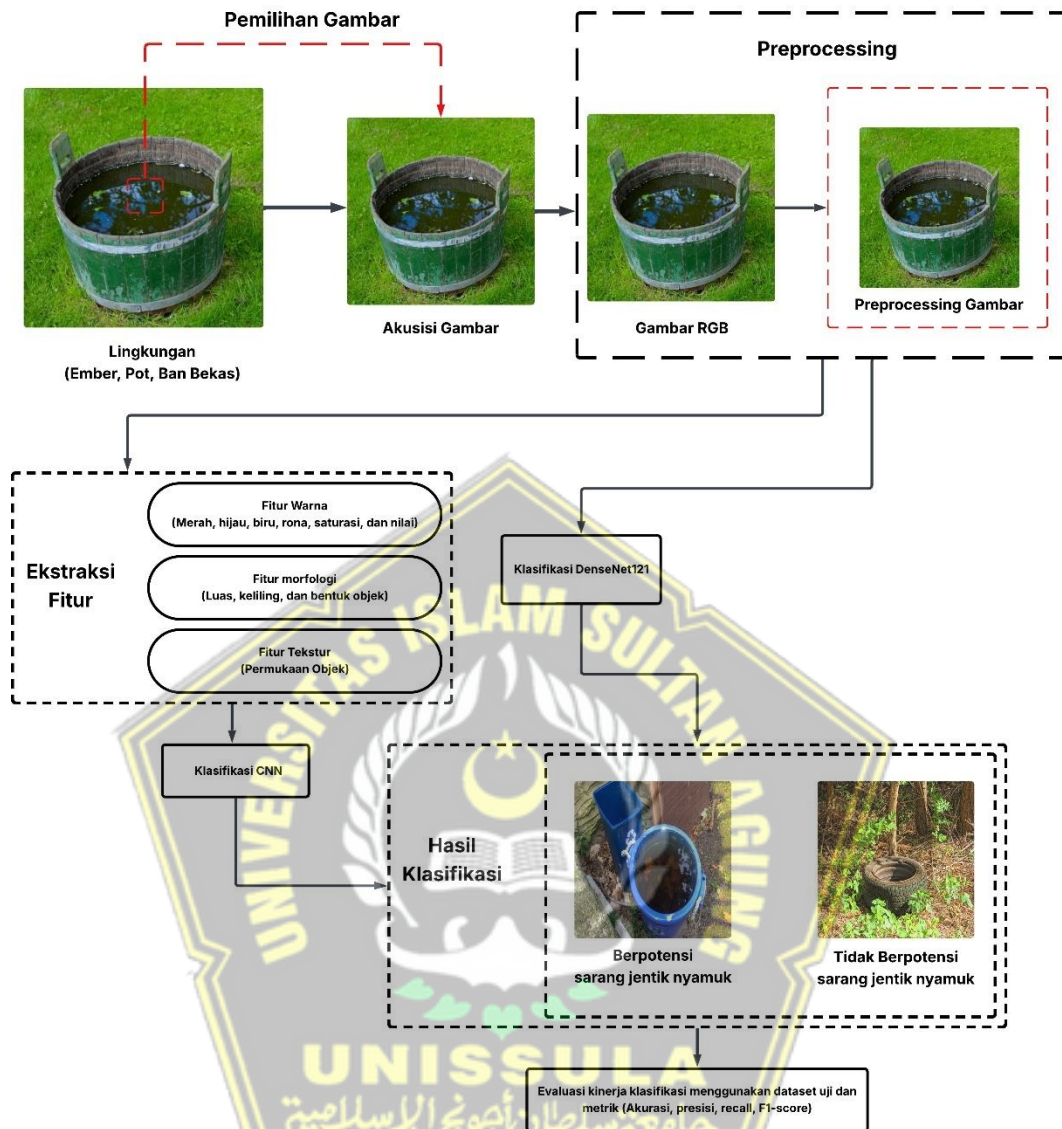
2.2.2 Pengolahan Citra (Image Processing)

Pengolahan citra digital merupakan tahap awal dalam sistem klasifikasi berbasis Deep Learning karena berfungsi menyiapkan data agar sesuai dengan karakteristik model yang digunakan, terutama pada klasifikasi potensi sarang jentik nyamuk. Dengan teknik pengolahan citra, penelitian ini dapat menganalisis lingkungan untuk mengklasifikasikan potensi sarang jentik nyamuk seperti warna, bentuk ataupun tekstur. Dalam proses pengolahan citra ada beberapa tahapan, yaitu:

1. Pengambilan citra lingkungan dilakukan menggunakan handphone untuk memotret kondisi sekitar yang berpotensi dan tidak berpotensi menjadi sarang jentik nyamuk.
2. Tahap selanjutnya adalah *preprocessing*, yaitu pembersihan dan penyesuaian citra agar fokus terhadap objek utama berupa wadah atau genangan air.
3. Dari citra yang telah diproses, dilakukan ekstraksi fitur visual yang relevan seperti bentuk, tekstur, dan pola permukaan wadah yang berhubungan dengan potensi keberadaan jentik nyamuk.
4. Hasil ekstraksi tersebut kemudian di analisis oleh model klasifikasi berbasis *Convolutional Neural Network (CNN)* maupun *DenseNet121* untuk menentukan kategori apakah suatu citra menunjukkan lingkungan yang berpotensi atau tidak berpotensi menjadi sarang jentik nyamuk

Melalui penerapan teknik ini, sistem dapat mengenali karakteristik wadah atau genangan air pada citra lingkungan secara otomatis, sehingga tahapan analisis tidak lagi bergantung pada pengamatan manual. Untuk memperjelas alur kerja penelitian, ditampilkan ilustrasi tahapan *image processing* yang diterapkan dalam klasifikasi potensi sarang jentik nyamuk.





Gambar 2. 2 Ilustrasi *Image Processing* (Taha dkk., 2022)

Pada gambar 2.2 menunjukkan ilustrasi alur *image processing* untuk mengklasifikasikan potensi sarang jentik nyamuk berdasarkan citra lingkungan. Proses diawali dengan pengambilan gambar wadah penampung air, seperti ember, pot bunga, maupun ban bekas. Selanjutnya citra yang diperoleh melalui tahap *preprocessing* untuk menyesuaikan ukuran, normalisasi, serta pembersihan latar belakang agar fokus pada objek utama. Setelah itu, dilakukan ekstraksi fitur dari citra wadah yang meliputi:

1. Fitur warna (merah, hijau, biru, rona, saturasi, dan nilai),
2. Fitur morfologis (luas, keliling, dan bentuk objek),

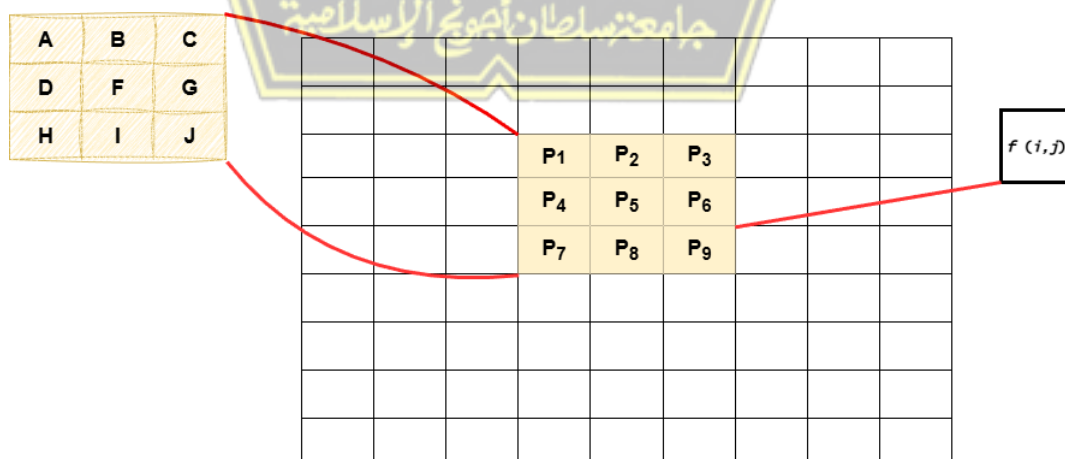
3. Fitur tekstur (karakteristik permukaan objek).

Fitur-fitur tersebut kemudian dianalisis menggunakan pendekatan klasifikasi berbasis deep learning, yaitu model *Convolutional Neural Network* (CNN) dan *DenseNet*121. Hasil klasifikasi digunakan untuk menentukan kategori citra, apakah berpotensi atau tidak berpotensi sebagai sarang jentik nyamuk. Tahap akhir adalah evaluasi kinerja model dengan menggunakan metrik umum antara lain *Akurasi*, *Presisi*, *Recall*, dan *F1-Score*, guna menilai performa sistem secara keseluruhan.

2.3 Konvolusi

2.3.1 Teori Konvolusi

Dalam pengolahan citra digital, konvolusi merupakan salah satu metode yang digunakan untuk meningkatkan kualitas citra. Implementasi konvolusi pada citra digital dilakukan dalam bentuk dua dimensi. Penerapan teknik ini memiliki tujuan, seperti deteksi garis (*ridge detection*), mempertajam citra (*sharpening*), dan mengaburkan citra (*blurring*). Setiap proses *filtering* melibatkan pergeseran serta operasi perkalian titik antara data input dan nilai *kernel*. Lapisan konvolusi secara signifikan meningkatkan kompleksitas model melalui optimalisasi *outputnya*. Berikut adalah ilustrasi dari proses konvolusi antara matriks citra digital dengan matriks kernel:



Gambar 2. 3 Matriks Konvolusi

Keterangan:

$$f(i, j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9$$

$f(i, j)$ = hasil citra yang sudah di konvolusi

$A - I$ = matriks *kernel*

$P_1 P_9$ = fungsi *kernel*

Proses konvolusi pada matriks citra $m \times n$ menggunakan *kernel* 3×3 ditunjukkan dalam persamaan 6. Langkah pertama mengekstrak sub-matriks berukuran 3×3 dari matriks citra dan mengubahnya ke dalam bentuk vektor berukuran $3^2 = 9$ elemen. Kedua transformasi ini berpedoman pada persamaan 1 dan 2. Setelah transformasi selesai, kedua vektor tersebut dikalikan secara titik. Berikut penjelasan mengenai komponen yang terlibat dalam proses konvolusi, yaitu kernel, stride, dan feature map:

1. *Kernel*

Kernel merupakan matriks numerik berukuran kecil yang diaplikasikan dalam proses konvolusi citra. *Kernel* dengan dimensi dan pola angka yang bervariasi akan menghasilkan *output* yang berbeda-beda dalam operasi konvolusi. Secara umum, *kernel* memiliki dimensi $m \times n$, ukuran yang paling sering digunakan adalah 3×3 . Terdapat berbagai variasi bentuk matriks kernel 3×3 yang disesuaikan dengan efek yang diinginkan. Bentuk-bentuk matriks tersebut dapat diamati pada contoh tabel berikut:

Tabel 2. 1 Bentuk efek *Kernel*

<i>Operation</i>	<i>Kernel</i>
<i>Ridge Detection</i>	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
<i>Sharpen</i>	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
<i>Blurring</i>	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Tabel 2.1 menunjukkan beberapa contoh kernel dua dimensi yang umum digunakan dalam proses konvolusi pada pengolahan citra digital.

Setiap kernel memiliki pola nilai berbeda yang menghasilkan efek tertentu ketika diterapkan pada citra.

2. *Stride*

Stride merupakan *parameter* yang mengatur seberapa jauh *filter* bergeser saat melakukan proses konvolusi. Apabila nilai *stride* ditetapkan sebesar 1, maka filter akan bergerak satu piksel setiap kali bergeser secara horizontal maupun vertikal. Nilai *stride* yang kecil memungkinkan jaringan menangkap lebih banyak detail dari citra karena pergeseran *filter* lebih rapat, namun hal ini juga meningkatkan beban komputasi. Sebaliknya, *stride* yang lebih besar membuat proses konvolusi lebih efisien, tetapi dapat mengurangi detail informasi yang diperoleh. Oleh karena itu, pemilihan nilai *stride* yang tepat sangat penting karena tidak semua konfigurasi *stride* menghasilkan performa yang optimal.

3. *Feature map*

Feature Map adalah hasil *output* dari proses konvolusi, yaitu operasi perkalian antara nilai piksel input dengan filter tertentu. Melalui *Feature Map*, model dapat menganalisis dan mengenali pola-pola penting pada bagian-bagian tertentu dari citra. *Visualisasi Feature Map* dilakukan untuk mengetahui fitur apa saja yang berhasil terdeteksi dan dipertahankan oleh jaringan selama proses ekstraksi fitur. Secara umum, semakin dalam letak suatu layer dalam jaringan, maka fitur yang diekstraksi akan semakin abstrak dan bersifat lebih umum. (Affifah dkk., 2022)

2.3.2 Proses Konvolusi pada Citra Digital

Operasi konvolusi dilakukan dengan cara menggeser *kernel* pada citra secara bertahap di setiap posisi piksel, kemudian menyimpan hasil perhitungannya ke dalam matriks keluaran baru. Contoh proses konvolusi dengan *kernel* berukuran 3×3 dan menggunakan nilai *stride* 2 adalah sebagai berikut: (MathWorks, 2025)

Citra $f(x,y)$ berukuran 5×5 serta *kernel* berukuran 3×3 dengan *stride* sebanyak 2 masing-masing adalah:

		<i>Vektor Citra</i>								
$f(x,y)=$	3	3	4	6	3	dan	$g(x,y)=$	<i>Kernel</i>		
	5	5	6	6	2			0	-1	0
	6	5	5	5	2			-1	5	-1
	5	7	6	6	4			0	-1	0
	2	4	2	3	3					

Gambar 2. 4 Contoh operasi konvolusi

Operasi konvolusi antara citra $f(x,y)$ dengan kernel $g(x,y)$, yaitu $f(x,y) * g(x,y)$ dapat diilustrasikan sebagai berikut:

Sub-matriks pertama diletakkan pada posisi sudut kiri atas citra, kemudian dilakukan perhitungan untuk memperoleh nilai piksel hasil konvolusi pada koordinat (1,1):

3	3	4	6	3	→				
5	5	6	6	2		6			
6	5	5	5	2					
5	7	6	6	4					
2	4	2	3	3					

Gambar 2. 5 Hasil Operasi Konvolusi Pertama

Nilai ini dihitung dengan cara berikut:

$$(0 \times 3) + (-1 \times 3) + (0 \times 4) + (-1 \times 5) + (5 \times 5) + (-1 \times 6) \\ + (0 \times 6) + (-1 \times 5) + (0 \times 5) = 6$$

Sub-matriks kemudian digeser dua piksel ke arah kanan sesuai dengan nilai *stride*, dan dilakukan kembali perhitungan untuk mendapatkan nilai piksel hasil konvolusi pada posisi (1,3):

3	3	4	6	3					
5	5	6	6	2		6	11		
6	5	5	5	2					
5	7	6	6	4					
2	4	2	3	3					

Gambar 2. 6 Hasil Operasi Konvolusi Kedua

Nilai ini dihitung dengan cara berikut:

$$(0 \times 4) + (-1 \times 6) + (0 \times 3) + (-1 \times 6) + (5 \times 6) + (-1 \times 2) \\ + (0 \times 5) + (-1 \times 5) + (0 \times 2) = 11$$

Sub-matriks kemudian digeser dua piksel ke bawah dan lakukan kembali perhitungan untuk mendapatkan nilai piksel hasil konvolusi pada posisi (3,1):

3	3	4	6	3					
5	5	6	6	2		6	11		
6	5	5	5	2		15			
5	7	6	6	4					
2	4	2	3	3					

Gambar 2. 7 Hasil Operasi Konvolusi Ketiga

Nilai ini dihitung dengan cara berikut:

$$(0 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 5) + (5 \times 7) + (-1 \times 6) \\ + (0 \times 2) + (-1 \times 4) + (0 \times 2) = 15$$

Sub-matriks kemudian digeser dua piksel ke arah kanan sesuai dengan nilai *stride*, lalu mulai lagi melakukan konvolusi dari sisi kiri citra. Setiap konvolusi, geser sub-matriks dua piksel ke kanan (3,3):

3	3	4	6	3					
5	5	6	6	2		6	11		
6	5	5	5	2		15	12		
5	7	6	6	4					
2	4	2	3	3					

Gambar 2. 8 Hasil Operasi Konvolusi Keempat

Nilai ini dihitung dengan cara berikut:

$$(0 \times 5) + (-1 \times 5) + (0 \times 2) + (-1 \times 6) + (5 \times 6) + (-1 \times 4) \\ + (0 \times 2) + (-1 \times 3) + (0 \times 3) = 12$$

Berikut merupakan hasil dari matriks yang telah di konvolusi:

Tabel 2. 2 Tabel hasil konvolusi

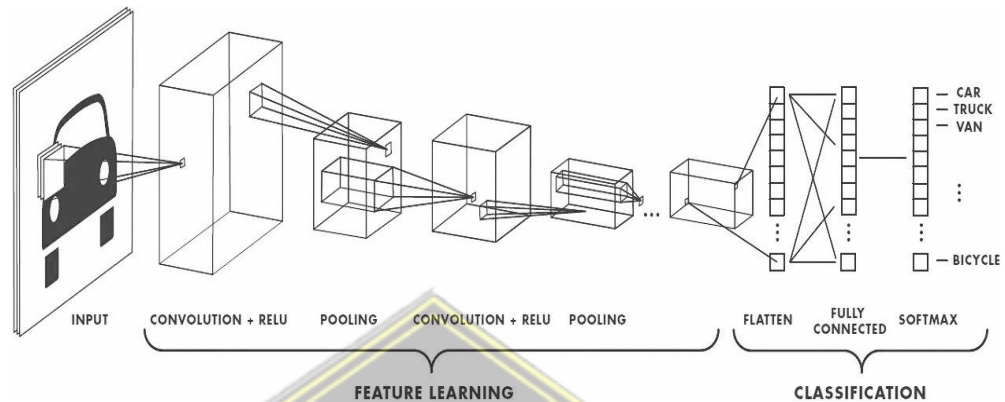
3	3	4	6	3
5	6	11	6	2
6	15	12	5	2
5	7	6	6	4
2	4	2	3	3

Tabel 2.2 menunjukkan hasil konvolusi citra 5×5 menggunakan *kernel* 3×3 dengan *stride* 2 sehingga menghasilkan *feature map* berukuran 2×2. Kernel pertama ditempatkan pada sudut kiri atas dan menghasilkan nilai 6 pada posisi (1,1). Setelah digeser dua piksel ke kanan, diperoleh nilai 11 pada posisi (1,2). Pergerakan kernel dua piksel ke bawah menghasilkan nilai 15 pada posisi (2,1), dan pergeseran terakhir ke kanan menghasilkan nilai 12 pada posisi (2,2). Dengan demikian, Tabel 2.2 memperlihatkan empat hasil konvolusi sesuai urutan pergerakan *kernel*.

2.3.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur jaringan saraf tiruan yang dirancang khusus untuk menangani masalah pengolahan citra. CNN terbukti sangat efektif dalam tugas klasifikasi, deteksi objek, serta pengenalan pola visual, terutama karena kemampuannya dalam

mengekstraksi fitur spasial dari gambar input. Model ini bekerja dengan cara menyerupai cara kerja otak manusia dalam memproses visual sehingga dapat memahami detail gambar dengan baik.



Gambar 2. 9 Arsitektur CNN (Anhar, Anhar Putra, 2023)

Pada gambar 2.9 Arsitektur CNN umumnya terdiri dari dua tahap utama, yaitu *feature learning* dan *classification*. Dalam *feature learning* melibatkan beberapa komponen penting, yaitu:

1. *Convolutional Layer*: lapisan inti dalam CNN yang bertugas mengekstraksi fitur lokal dari gambar input menggunakan *convolution operation*.
2. *Activation Function (ReLU)*: fungsi aktivitas yang digunakan setelah konvolusi untuk memperkenalkan non-linearitas. ReLU (*Rectified Linear Unit*) merupakan fungsi yang umum digunakan karena sederhana dan efektif.
3. *Pooling Layer*: digunakan untuk mereduksi dimensi fitur sambil mempertahankan informasi penting. *Pooling* yang sering digunakan adalah *max pooling* dan *average pooling*.

Dalam klasifikasi terdiri dari beberapa proses lanjutan, yaitu:

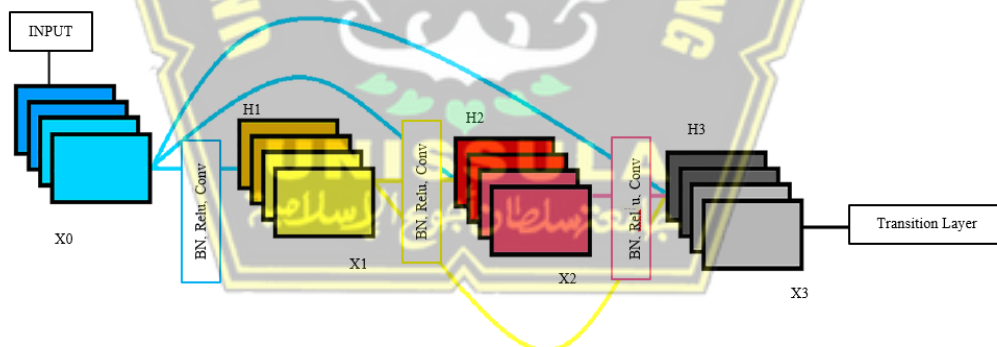
1. *Flatten Layer*: mengubah hasil dari fitur map menjadi satu vector agar bisa diproses oleh layer klasifikasi.
2. *Fully Connected Layer*: mirip dengan jaringan saraf biasa (ANN) yang menyambungkan semua neuron dari layer sebelumnya.

3. *Softmax*: digunakan untuk mengubah *output* menjadi nilai probabilitas dari masing-masing kelas dan memilih kelas dengan probabilitas tertinggi sebagai prediksi akhir. (Anhar, Anhar Putra, 2023)

2.3.4 DenseNet121

DenseNet121 merupakan salah satu arsitektur dalam keluarga *Densely Connected Convolutional Networks (DenseNet)* yang dirancang untuk meningkatkan efisiensi dan akurasi pada jaringan saraf dalam pengolahan citra. *DenseNet* dikembangkan untuk mengatasi kelemahan arsitektur CNN konvensional yang sering mengalami permasalahan seperti *vanishing gradient* dan duplikasi informasi fitur, sehingga menghasilkan informasi dan gradien yang lebih efisien.

Dalam arsitektur *DenseNet121* setiap layer menerima semua *output* dari layer sebelumnya sebagai inputnya. Pendekatan ini memungkinkan jaringan untuk mempelajari fitur yang lebih kompleks dengan jumlah parameter yang lebih sedikit.



Gambar 2. 10 Arsitektur *DenseNet121* (A. D. Saputra dkk., 2023)

Pada gambar 2.10, *DenseNet121* sendiri terdiri dari total 121 layer yang dibentuk dari susunan:

1. 5 layer awal (*Convolution* dan *Pooling*)
2. 3 transition layer
3. 2 dense block (yang terdiri dari kombinasi 1x1 dan 3x3 *convolution*)

4. Serta 1 layer klasifikasi akhir.

Jika di jumlahkan: $5 + (6 + 12 + 24 + 16) \times 2 = 121$. Dengan penjelasan bahwa 5 merupakan *Convolution* dan *Pooling layer*. 3 merupakan *transition layer* (6,12,24). 2 merupakan *DenseBlock* (1x1 dan 3x3 *convolution*). 1 merupakan layer klasifikasi. (A. D. Saputra dkk., 2023)



BAB III

METODE PENELITIAN

3.1 Metodologi Penelitian

Pada bagian ini memaparkan metodologi yang akan digunakan pada penelitian ini. Metode yang digunakan adalah *Convolutional Neural Network* (CNN) dengan arsitektur *DenseNet121* dan tujuan dari penelitian ini adalah membangun sistem klasifikasi yang mampu mengidentifikasi potensi sarang jentik nyamuk berdasarkan citra lingkungan secara otomatis dan akurat.

3.1.1 Alur Penelitian

Proses penelitian dimulai dengan akuisisi citra lingkungan, dilanjutkan dengan *preprocessing citra*, perancangan model *Convolutional Neural Network* (CNN) dan *DenseNet121*, pelatihan model menggunakan *dataset* yang telah dipisahkan, evaluasi kinerja menggunakan metrik tertentu, serta analisis hasil klasifikasi. Tahapan penelitian ini meliputi:

1. Akuisisi data citra lingkungan

Tahap awal Adalah pengumpulan data berupa citra lingkungan yang berpotensi menjadi sarang jentik nyamuk. Citra diperoleh dengan cara dokumentasi langsung menggunakan kamera handphone pada objek wadah air, seperti ember, pot bunga, ban bekas dan lingkungan sekitar yang berpotensi dan tidak berpotensi sarang jentik nyamuk. Proses akuisisi mempertimbangkan variasi kondisi pencahayaan, sudut pengambilan gambar, serta latar belakang lingkungan agar data yang diperoleh lebih representative dan mampu menggambarkan kondisi nyata di lapangan

2. *Preprocessing* Citra

Citra yang telah diperoleh kemudian melalui tahapan *preprocessing* untuk memastikan data memiliki kualitas dan format yang seragam. Langkah-langkah *preprocessing* meliputi

Resize, yaitu mengubah resolusi citra menjadi 640x640 piksel agar sesuai dengan input layer model.

b. Normalisasi, yaitu mengubah nilai piksel ke dalam rentang [0-1] untuk mempercepat proses komputasi dan menstabilkan pelatihan model.

c. Augmentasi, yaitu memperbanyak variasi data dengan melakukan transformasi seperti *flipping* horizontal maupun vertikal, *zoom*, dan penyesuaian kecerahan. Augmentasi bertujuan meningkatkan kemampuan generalisasi model serta mengurangi resiko *overfitting*.

3. Perancangan model CNN (*Convolutional Neural Network*)

Pada tahap ini dilakukan perancangan arsitektur *Convolutional Neural Network* (CNN) yang digunakan sebagai salah satu model klasifikasi. CNN (*Convolutional Neural Network*) dirancang dengan beberapa lapisan, antara lain convolutional layer, pooling layer, fully connected layer, dropout, serta output layer dengan fungsi aktivasi softmax. Tujuan dari CNN (*Convolutional Neural Network*) ini adalah mengekstraksi ciri visual dari citra lingkungan dan mengklasifikasikannya ke dalam dua kategori, yaitu berpotensi atau tidak berpotensi.

4. Implementasi *DenseNet121* dengan *transfer learning*

Selain CNN, penelitian ini juga menggunakan arsitektur *DenseNet121* yang telah terbukti efektif dalam ekstraksi fitur citra. Model ini diimplementasikan dengan pendekatan *transfer learning*, dimana bobot awal yang diperoleh dari dataset *ImageNet* dimanfaatkan kembali, lalu dilakukan *fine-tuning* pada lapisan akhir agar sesuai dengan dataset penelitian. *DenseNet121* dipilih karena memiliki konektivitas

antar layer yang padat, sehingga mampu menghasilkan representasi fitur yang lebih efisien.

5. Pelatihan dan validasi model

Dataset yang telah diproses kemudian dibagi menjadi data latih dan data uji. Data latih digunakan untuk memperbarui bobot model, sedangkan data uji digunakan untuk memantau kinerja model selama proses pelatihan agar tidak terjadi *overfitting*. Proses pelatihan dilakukan selama sejumlah epoch dengan parameter yang telah ditentukan, seperti *learning rate*, *optimizer*, dan *batch size*.

6. Evaluasi kinerja model

Setelah model selesai dilatih, tahap selanjutnya adalah evaluasi menggunakan data uji yang sebelumnya dipisahkan dari *dataset*. Evaluasi dilakukan dengan menggunakan metrik kuantitatif, antara lain:

- a. *Accuracy*, untuk mengukur tingkat ketepatan prediksi model.
- b. *Precision*, untuk melihat seberapa tepat model dalam memprediksi kelas positif.
- c. *Recall*, untuk mengukur kemampuan model dalam mengenali seluruh data positif.
- d. *F1-Score*, sebagai rata-rata antara *precision* dan *recall*.
- e. *Confusion matrix*, untuk memberikan Gambaran distribusi prediksi benar dan salah pada masing-masing kelas.

3.1.2 Pengumpulan Dataset:

Pengambilan citra dilakukan di berbagai Lokasi yang memiliki potensi berbeda dalam menampung jentik nyamuk. Dataset yang digunakan dalam penelitian ini terbagi menjadi dua kelas seperti area dengan genangan air, saluran air tersumbat, pot bunga, dan tempat sampah terbuka. Sebagai pembanding, gambar dari lingkungan yang tidak memiliki potensi menjadi sarang jentik, seperti tanah terbuka, pot tanaman hias, dan lapangan juga dikumpulkan.

Seluruh citra dalam dataset ini disimpan dalam format JPG karena memiliki ukuran relatif kecil dan kompatibilitas tinggi dengan *deep learning* seperti *TensorFlow* dan *Keras*. Format ini juga mendukung efisiensi dalam proses pemuatan data dan pelatihan model. Gambar dikumpulkan dan disimpan dengan resolusi 640x640 piksel untuk mempertahankan detail visual yang relevan seperti tekstur tanah, genangan air, dan benda di lingkungan sekitar. Ukuran ini cukup besar untuk menjaga kualitas informasi visual, namun tetap efisien untuk diproses. Untuk kelas nya sendiri terbagi menjadi dua, yaitu:

a. Kelas Berpotensi sarang jentik nyamuk

Citra yang termasuk kedalam kelas ini umumnya menunjukkan keberadaan genangan air yang tidak mengalir di area terbuka, seperti di dalam pot kosong, selokan tersumbat, dan genangan air yang tidak terpantau. Area ini bersifat lembap dan tidak mengalami perawatan rutin yang merupakan kondisi ideal bagi nyamuk untuk berkembang biak.

b. Kelas Tidak Berpotensi sarang jentik nyamuk

Citra dalam kategori ini menampilkan area lingkungan yang bersih, kering atau tidak memiliki wadah air terbuka. Misalnya pot bunga yang berisi tanaman dengan tanah padat, permukaan tanpa genangan. Data kemudian dibagi menjadi 2 subset, yaitu:

- a. *Training set* 80% dari total *dataset*.
- b. *Testing set* 20% dari total *dataset*.

Pembagian ini bertujuan untuk menghindari bias dan memastikan bahwa data uji tidak pernah digunakan selama proses pelatihan model.

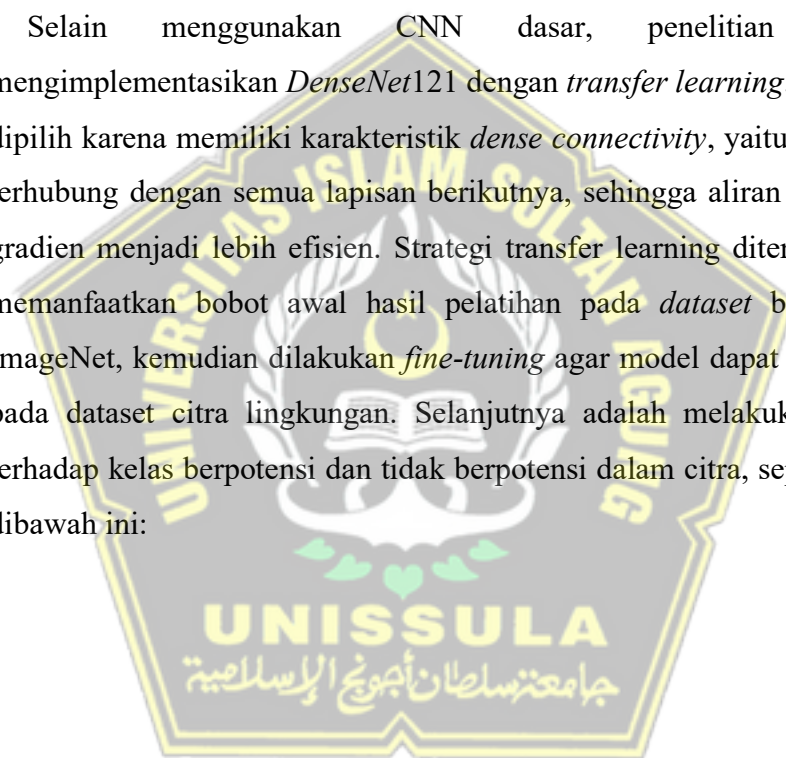
3.1.3 Perancangan Model

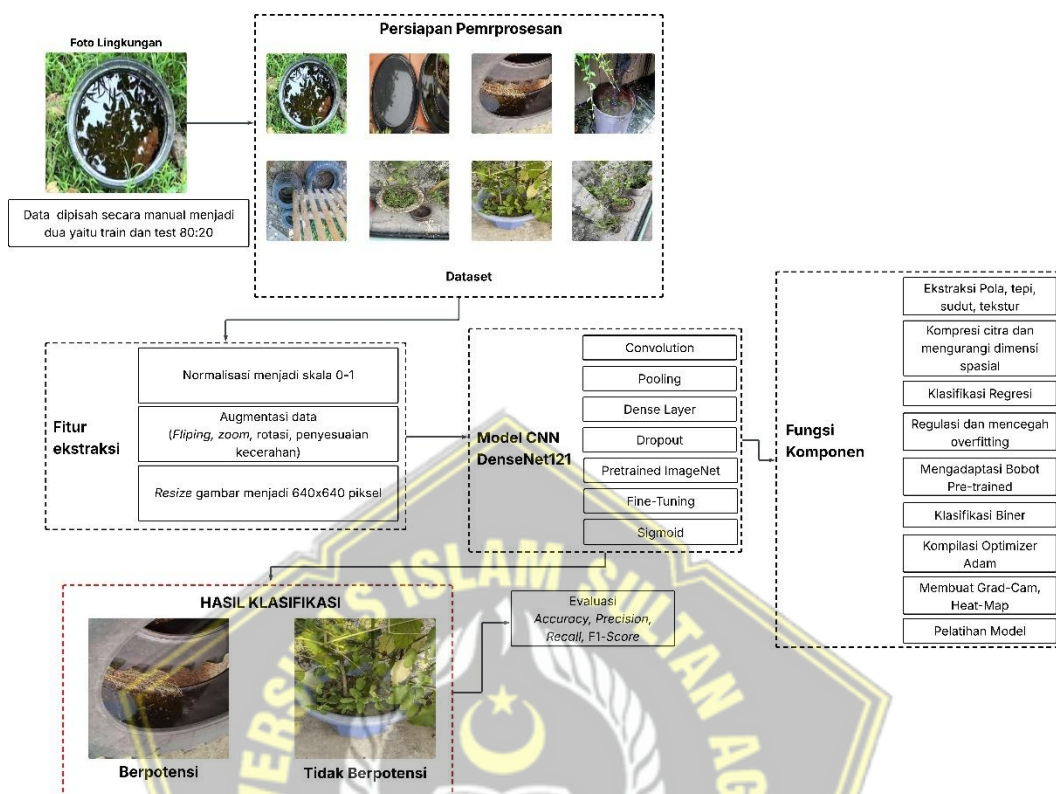
Dalam penelitian ini, dirancang sebuah pendekatan berbasis *deep learning* untuk mendeteksi keberadaan potensi sarang jentik nyamuk melalui citra lingkungan. Perancangan model dilakukan dengan memanfaatkan *Convolutional Neural Network* (CNN) yang berfungsi dalam melakukan ekstraksi fitur citra secara otomatis. CNN dipilih karena kemampuannya dalam

mengenali pola visual kompleks tanpa memerlukan segmentasi atau teknik pemrosesan manual yang rumit.

Proses ekstraksi fitur dilakukan melalui tahapan propagasi maju (*forward propagation*) pada lapisan konvolusi dan *pooling*, sehingga menghasilkan representasi citra yang lebih ringkas namun tetap memuat informasi penting. Representasi tersebut diproses lebih lanjut oleh lapisan klasifikasi untuk menentukan kategori citra yang mengandung potensi sarang jentik atau tidak mengandung potensi sarang jentik.

Selain menggunakan CNN dasar, penelitian ini juga mengimplementasikan *DenseNet121* dengan *transfer learning*. *DenseNet121* dipilih karena memiliki karakteristik *dense connectivity*, yaitu setiap lapisan terhubung dengan semua lapisan berikutnya, sehingga aliran informasi dan gradien menjadi lebih efisien. Strategi transfer learning diterapkan dengan memanfaatkan bobot awal hasil pelatihan pada *dataset* besar misalnya ImageNet, kemudian dilakukan *fine-tuning* agar model dapat menyesuaikan pada *dataset* citra lingkungan. Selanjutnya adalah melakukan klasifikasi terhadap kelas berpotensi dan tidak berpotensi dalam citra, seperti *workflow* dibawah ini:





Gambar 3. 1 WorkFlow Model

Pada gambar 3.1 ditunjukkan alur kerja sistem klasifikasi tempat berpotensi menjadi sarang jentik nyamuk. Sistem ini terdiri dari beberapa tahap utama, yaitu pengambilan data, *preprocessing*, ekstraksi fitur menggunakan CNN berarsitektur *DenseNet121*.

1. Pengambilan data (Foto lingkungan)

Proses dimulai dengan pengumpulan citra lingkungan yang berpotensi maupun tidak berpotensi menjadi tempat berkembang biaknya jentik nyamuk. Citra diperoleh dengan memotret berbagai wadah seperti ember, pot, dan ban bekas. Setiap citra diberi label sesuai kondisi sebenarnya, yaitu berpotensi atau tidak berpotensi.

2. *Preprocessing*

Tahap ini merupakan proses penting untuk mempersiapkan data gambar sebagai input ke dalam model. Adapun tahapapan *preprocessing* meliputi:

a. Normalisasi piksel

Nilai piksel citra dinormalisasikan ke rentang [0-1] dengan cara membagi setiap piksel RGB dengan 255, sehingga proses komputasi menjadi lebih stabil dan efisien.

b. Augmentasi data

Dilakukan dengan operasi *flipping*, *zoom*, rotasi dan penyesuaian kecerahan untuk memperluas variasi data dan mengurangi resiko *overfitting*.

c. *Resize* gambar

Seluruh gambar diubah ukurannya menjadi 640x640 piksel agar seragam dan sesuai dengan input layer model.

3. Ekstraksi fitur dan klasifikasi

Tahap berikutnya adalah melakukan ekstraksi fitur dari citra lingkungan yang telah melalui proses preprocessing, kemudian melanjutkannya ke tahap klasifikasi. Pada penelitian ini digunakan model berbasis *Convolutional Neural Network* (CNN) dengan arsitektur *DenseNet*¹²¹ yang dimodifikasi menggunakan pendekatan transfer learning. Proses ekstraksi fitur dilakukan melalui beberapa lapisan utama, yaitu:

a. *Convolution Layer*, berfungsi untuk mengekstraksi pola spasial dasar dari citra seperti tepi, sudut, tekstur, dan membuat grad-cam atau heat-map.

b. *Pooling Layer*, bertugas mengompresi citra dengan mengurangi dimensi spasial namun tetap mempertahankan informasi penting.

c. *Dense Layer*, digunakan untuk melakukan pemetaan non-linear dari fitur yang telah diekstraksi menuju kelas target.

d. *Dropout*, berfungsi sebagai regulasi agar model tidak mengalami *overfitting* dengan cara menonaktifkan neuron secara acak saat pelatihan.

- e. *Pretrained ImageNet*, yaitu bobot awal dari *DenseNet121* yang digunakan sebagai *feature extractor* agar model dapat memanfaatkan representasi visual yang sudah matang.
- f. *Fine-Tuning*, yaitu penyesuaian kembali pada lapisan akhir *DenseNet121* agar selaras dengan *dataset* penelitian.
- g. *Sigmoid Layer*, digunakan pada tahap akhir untuk menghasilkan probabilitas klasifikasi biner berpotensi atau tidak berpotensi.

4. Hasil klasifikasi

Output dari model berupa kategori Berpotensi atau Tidak berpotensi. Contohnya, wadah berisi air tergenang dan tidak tertutup akan diprediksi sebagai berpotensi, sedangkan wadah kering atau tertutup rapat diprediksi sebagai tidak berpotensi.

5. Evaluasi model

Tahap terakhir adalah mengevaluasi kinerja model menggunakan beberapa metrik, yaitu

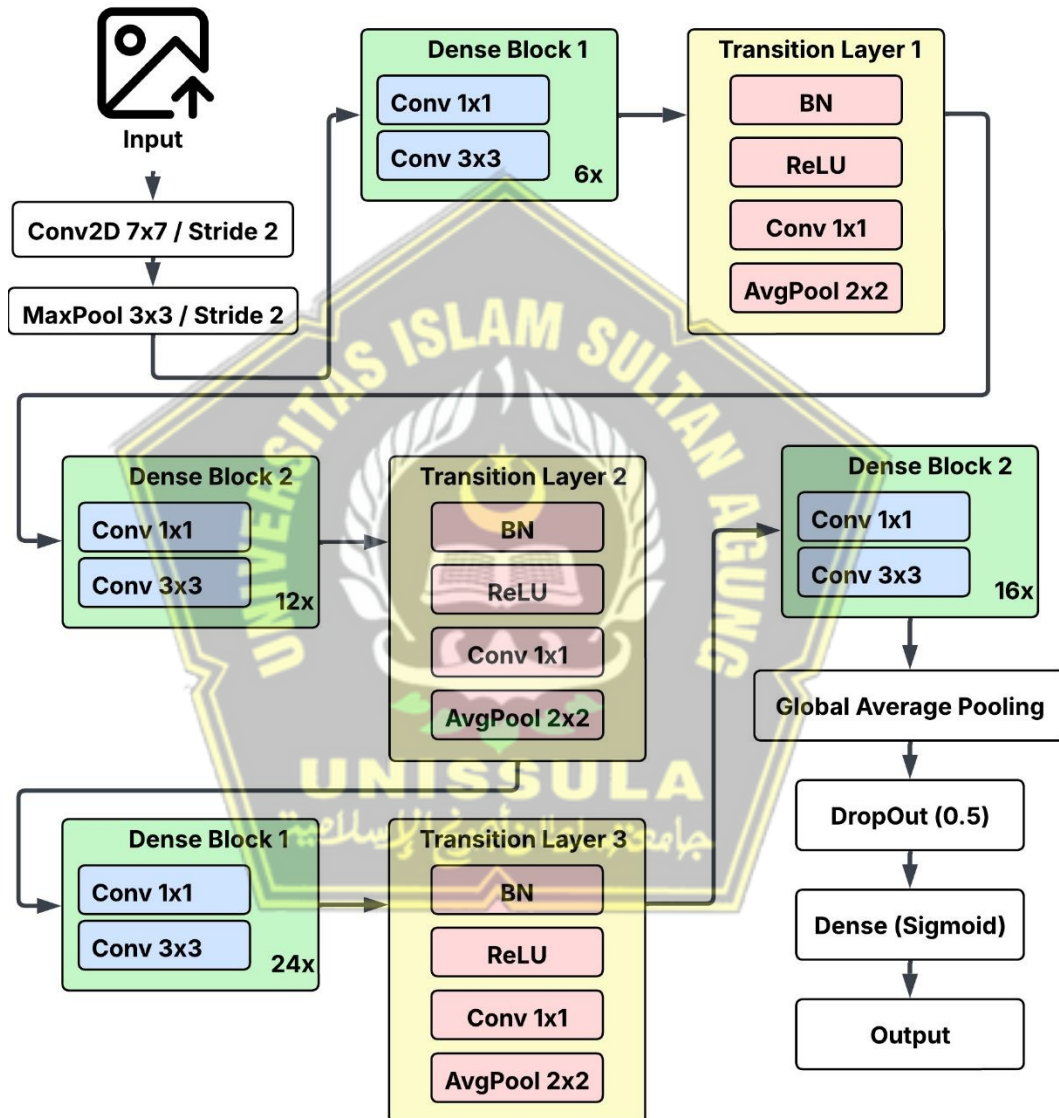
- 1) *Confusion matrix*, yang menggambarkan jumlah prediksi benar dan salah pada masing-masing kelas.
- 2) *Accuracy*, yang mengukur proporsi prediksi benar dari seluruh data uji.
- 3) *Precision*, yang menunjukkan tingkat ketepatan model dalam memprediksi kelas positif.
- 4) *Recall*, yang mengukur sejauh mana model mampu mendeteksi semua data positif.
- 5) *F1-Score*, yang merupakan rata-rata dari *precision* dan *recall*, sehingga memberikan penilaian yang lebih seimbang.

3.1.4 Arsitektur *Convolutional Neural Network* (CNN) *DenseNet121*

Model yang digunakan pada penelitian ini merupakan *Convolutional Neural Network* (CNN) yang dibangun dengan memanfaatkan arsitektur *DenseNet121* sebagai *feature extractor* utama melalui pendekatan *transfer learning*.

*DenseNet*₁₂₁ termasuk dalam keluarga CNN modern yang memiliki karakteristik koneksi padat antar-layer (*dense connectivity*), di mana setiap layer menerima input dari seluruh layer sebelumnya dalam blok yang sama (*dense block*).

Penjelasan masing-masing komponen adalah sebagai berikut.



Gambar 4. 1 Arsitektur Sistem *DenseNet*₁₂₁

Pada Gambar 4.1 menampilkan arsitektur dari model *DenseNet*₁₂₁ yang digunakan sebagai base model untuk ekstraksi fitur citra. Lapisan-Lapisannya dijelaskan sebagai berikut:

1. *Convolution Layer (Conv2D 7x7, Stride 2)*

Lapisan pertama pada *DenseNet121* adalah Conv2D dengan ukuran *kernel 7x7* dan *Stride 2*. Lapisan ini bertugas untuk menangkap fitur dasar citra seperti tepi, garis, dan perbedaan warna.

- a. Ukuran *kernel (7x7)* : angka *7x7* menunjukkan ukuran *kernel* yang digunakan dalam proses konvolusi, jadi maksudnya 7 tinggi kernel dan lebar 7 kernel yang artinya, kernel akan membaca potongan gambar berukuran 7 piksel x 7 piksel setiap kali melakukan perhitungan konvolusi.
- b. *Stride (2)* : membuat kernel bergeser dua piksel setiap langkah, sehingga ukuran *feature map* mengecil dua kali lipat. Fungsinya mengurangi ukuran *output (feature map)* karena kernel melangkah lebih jauh setiap kali geser, menghemat komputasi karena lebih sedikit posisi yang dihitung, menjadi bentuk *downsampling* awal sebelum *pooling* atau menyederhanakan data tanpa hilang fitur penting.

Tujuannya menyaring fitur paling dasar dari citra lingkungan seperti tepi wadah, bentuk genangan atau batas objek. Adapun rumus konvolusi sebagai berikut:

2. *MaxPooling Layer (3x3, Stride 2)*

Setelah melalui *convolution layer* awal, hasil keluaran diteruskan ke lapisan *MaxPooling* dengan ukuran kernel *3x3* dan stride 2. Lapisan ini berfungsi untuk mengurangi ukuran spasial seperti dimensi tinggi dan lebar dari *feature map* yang dihasilkan oleh lapisan konvolusi sebelumnya, sambil tetap mempertahankan fitur yang paling penting.

- a. Ukuran *kernel (3x3)* : angka *3x3* menunjukkan bahwa setiap kali proses pooling dilakukan, sistem akan melihat area citra berukuran 3 x 3 piksel, jadi hanya nilai maksimum yang dipilih untuk diteruskan ke lapisan berikutnya. Hal ini membuat model hanya mempertahankan fitur yang paling dominan dari setiap area kecil.

- b. *Stride* (2) : Nilai *stride* 2 menunjukkan bahwa *pooling* bergeser dua piksel setiap langkah, sehingga ukuran *output* menjadi lebih kecil dibandingkan *input* nya.

Fungsi utama lapisan *MaxPooling*, yaitu mengurangi ukuran data untuk menekan jumlah parameter dan mempercepat komputasi, mempertahankan fitur paling dominan sehingga model lebih fokus pada pola penting, meningkatkan ketahanan model terhadap pergeseran posisi objek. Tujuannya menyaring hasil konvolusi agar hanya fitur penting seperti bentuk, batas genangan air, dan tekstur dominan lingkungan yang diteruskan ke tahap berikutnya. Dengan demikian, proses selanjutnya dapat lebih fokus pada fitur esensial tanpa kehilangan konteks citra.

3. *Convolution layer* (1x1 dan 3x3)

Setelah proses ekstraksi fitur awal dan reduksi dimensi melalui *MaxPooling*, jaringan *DenseNet121* melanjutkan prosesnya ke lapisan konvolusi berikutnya, yaitu *Conv1x1* dan *Conv3x3*. Kedua lapisan ini bekerja secara berpasangan di dalam *Dense Block*, dan memiliki fungsi yang berbeda namun saling melengkapi. Adapun penjelasan *Conv1x1* dan *Conv3x3* sebagai berikut:

a. Lapisan *Conv1x1* (*Bottleneck Layer*)

Lapisan *Conv1x1* disebut sebagai *bottleneck layer* karena berfungsi mengurangi jumlah *channel* dari *feature map* tanpa mengubah ukuran spasialnya. Ukuran 1x1 berarti filter hanya membaca satu piksel per posisi, namun mengkombinasikan nilai dari semua *channel* pada posisi tersebut.

Fungsi utama dari lapisan *Conv1x1*, yaitu mengurangi beban komputasi sebelum masuk ke lapisan *Conv* berikutnya, mengontrol jumlah fitur yang akan diteruskan ke lapisan *Conv3x3*, mempertahankan informasi penting sambil membuat jaringan lebih efisien. Tujuannya menekan kompleksitas jaringan dan

mempercepat proses pelatihan tanpa kehilangan informasi penting dari fitur yang sudah di ekstraksi.

b. Lapisan Conv3x3

Setelah melewati bottleneck layer, fitur yang sudah disederhanakan diteruskan ke lapisan Conv3x3. Lapisan ini menggunakan filter berukuran 3 piksel x 3 piksel yang bergerak ke seluruh area *feature map* untuk mendeteksi pola visual yang lebih rinci seperti tekstur, tepi halus, atau detail kecil dari objek.

Fungsi utama dari Conv3x3, yaitu mengekstraksi pola lanjutan dari hasil Conv1x1, menangkap detail kecil seperti bentuk genangan air, bayangan, atau permukaan wadah, meningkatkan kemampuan jaringan dalam mengenali fitur kompleks dari citra lingkungan. Tujuannya lapisan Conv3x3 digunakan untuk memperdalam ekstraksi fitur sehingga model mampu mengenali karakteristik penting pada citra lingkungan, misalnya perbedaan antara genangan air dan permukaan tanah atau bentuk wadah yang berpotensi menjadi tempat berkembang biak nyamuk.

4. *Batch Normalization Layer* (BN)

Setelah proses konvolusi, hasil fitur yang dihasilkan sering memiliki distribusi nilai yang tidak stabil atau terlalu bervariasi. Untuk mengatasi hal ini digunakan lapisan *batch normalization* (BN) yang berfungsi menormalkan nilai aktivasi dari setiap *batch data* agar distribusinya tetap stabil selama proses pelatihan. Lapisan ini bekerja dengan menghitung rata-rata dan standar deviasi dari aktivasi setiap *batch*, kemudian menormalkan nilainya agar berada pada skala yang seragam. Proses normalisasi ini membantu jaringan belajar lebih cepat, menjaga kestabilan data antar lapisan dan mengurangi resiko *overfitting*.

a. Normalisasi nilai aktivasi

Pada setiap *batch* pelatihan dilakukan proses normalisasi menggunakan rumus sebagai berikut:

$$A_i = \frac{A_i - \mu_{Batch}}{\sqrt{\sigma_{Batch}^2 + \epsilon}}$$

Keterangan:

- A_i = nilai aktivasi
- μ_{Batch} = rata-rata batch
- σ_{Batch}^2 = variasi batch
- ϵ = konstanta kecil agar tidak terjadi pembagian nol

b. Transformasi hasil normalisasi

Setelah dinormalisasi, nilai tersebut diubah Kembali dengan dua parameter yang dapat dilatih, yaitu:

$$y_i = \gamma A_i + \beta$$

Keterangan:

- γ = factor skala
- β = factor pergeseran

Fungsi utama nya, yaitu menstabilkan nilai aktivvasi agar tidak terlalu besar atau kecil, mempercepat proses konvergensi saat pelatihan, mengurangi ketergantungan terhadap inisialisasi bobot, membantu mencegah *overfitting*. Tujuan nya *batch normalization* adalah menjaga agar data antar lapisan memiliki distribusi yang stabil sehingga jaringan CNN dapat belajar secara konsisten dan efisien.

5. *Activation ReLU layer (Reactified Linear Unit)*

Setelah dilakukan normalisasi, hasilnya diteruskan ke *activation ReLU layer (Reactified Linear Unit)*. Fungsi aktivasi ini memberikan sifat non-linearitas pada jaringan, sehingga model dapat mempelajari pola yang lebih kompleks dari data citra. ReLU bekerja dengan aturan sederhana ,yaitu:

$$f(x) = \max(0, x)$$

Keterangan:

- Jika nilai x lebih besar dari 0, maka hasilnya tetap x
 - Jika nilai x kurang dari atau sama dengan 0, maka hasilnya menjadi 0
- ReLU digunakan setelah setiap lapisan konvolusi untuk memastikan hanya fitur penting yang diteruskan ke lapisan berikutnya. Proses ini juga membantu jaringan belajar hubungan yang *non-linear* antara piksel citra, sehingga hasil klasifikasi menjadi lebih akurat. Fungsi utama dari *batch normalization*, yaitu menambahkan *non-linear* ke dalam model CNN, menghindari masalah *vanishing gradient* yang sering muncul pada fungsi aktivasi lain, mempercepat proses pelatihan karena perhitungannya sederhana. Tujuannya adalah meningkatkan kemampuan jaringan dalam mengenali pola kompleks seperti bentuk genangan air, permukaan wadah, atau batas objek yang menjadi ciri lingkungan berpotensi sarang jentik nyamuk.

6. *Average Pooling Layer (2x2)*

Setelah proses konvolusi dan aktivasi selesai, jaringan ini melakukan proses *Average Pooling* dengan ukuran *kernel* 2×2 . Lapisan ini digunakan pada bagian *Transition Layer* antar *Dense Block* dengan tujuan untuk mengurangi informasi penting dari fitur sebelumnya.

- a. Ukuran *kernel* (2×2) : angka 2×2 menunjukkan bahwa *pooling* dilakukan pada area citra berukuran 2 piksel. Berbeda dengan *MaxPooling* yang mengambil nilai tertinggi, *Average Pooling* menghitung rata-rata nilai dari keempat piksel dalam area tersebut.
- b. Peran dalam jaringan : Dengan melakukan perhitungan rata-rata, *Average pooling* mempertahankan informasi keseluruhan dari setiap area, bukan hanya nilai tertinggi saja. Hal ini membantu jaringan lebih stabil terhadap variasi pencahayaan atau *noise* kecil pada citra.

Fungsi utama *average pooling*, yaitu mengurangi ukuran dimensi (*downsampling*) dari *feature map*, mengurangi beban komputasi jaringan, menyederhanakan representasi fitur tanpa kehilangan informasi penting.

Tujuan *average pooling* adalah menjaga informasi global dari fitur yang telah diekstraksi dan memperkecil ukuran data agar proses pelatihan efisien.

7. *Global average pooling layer*

Setelah seluruh *Dense Block* selesai, hasil fitur terakhir diringkas melalui *Global average pooling* (GAP). Lapisan ini mengubah *feature map* dua dimensi menjadi vector satu dimensi dengan cara menghitung rata-rata dari setiap *channel* fitur secara keseluruhan. Cara kerja GAP, yaitu setiap *channel* pada *feature map* dihitung rata-ratanya menggunakan rumus sebagai berikut:

$$Y_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_c(i, j)$$

Keterangan:

- H = tinggi *feature map*
- W = lebar *feature map*
- $X_c(i, j)$ = nilai pada *channel* ke- c .

Peran GAP dapat menghapus lapisan *fully connected tradisional* sehingga jumlah parameter jauh lebih sedikit. GAP dapat menjaga hubungan spasial antar fitur tanpa menambah kompleksitas model, hasilnya berupa satu nilai rata-rata dari setiap *channel* yang kemudian dikirim ke lapisan *Dense*. Fungsi utama dari GAP sendiri adalah mereduksi *feature map* menjadi satu vektor per *channel*, mencegah *overfitting* karena tidak menambah bobot baru, dan mempertahankan makna spasial fitur. Tujuannya adalah menyederhanakan hasil ekstraksi fitur agar bisa langsung digunakan untuk proses klasifikasi tanpa menambah parameter besar seperti pada lapisan *fully connected* biasa.

8. *Dropout layer (0.5)*

Sebelum hasil dari GAP ke lapisan *Dense*, dilakukan proses *Dropout* dengan nilai 0.5. Teknik ini digunakan sebagai metode regularisasi untuk mencegah model terlalu menghafal data pelatihan. Cara kerja *Dropout*, yaitu saat proses pelatihan, *dropout* secara acak akan menonaktifkan sebagian *neuron* sekitar 50% dari lapisan sebelumnya pada setiap iterasi. *Neuron* yang tidak aktif tidak ikut dalam proses perhitungan dan pembaruan bobot pada saat itu. Peran *neuron* dalam jaringan yang aktif berbeda-beda di setiap iterasi, model belajar menjadi lebih fleksibel dan generalisasi lebih baik terhadap data baru. Fungsi utama dari *dropout*, yaitu mengurangi resiko *overfitting* pada data pelatihan, membuat model lebih tidak bergantung pada *neuron* tertentu, dan meningkatkan kemampuan generalisasi terhadap data uji. Tujuannya menjaga agar model tidak menghafal pola tertentu dari data latih dan tetap bisa bekerja baik pada data baru yang belum pernah dilihat.

9. *Dense (Sigmoid) dan Output*

Tahapan terakhir dari jaringan adalah lapisan *Dense (Fully connected)* dengan fungsi aktivasi *sigmoid*. Lapisan ini menghasilkan keluaran akhir berupa probabilitas yang menentukan apakah citra termasuk kategori berpotensi sarang jentik nyamuk atau tidak berpotensi. Fungsi aktivasi *sigmoid* dapat mengubah nilai input menjadi rentang antara 0 dan 1, adapun rumus dari aktivasi *sigmoid* sebagai berikut:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Jika hasil mendekati 1, maka citra diklasifikasikan sebagai berpotensi sarang jentik nyamuk, sedangkan nilai mendekati 0 berarti tidak berpotensi. Peran *dense* dengan satu *neuron output* digunakan karena masalah klasifikasi ini bersifat biner dua kelas. Fungsi utama *sigmoid*, yaitu mengubah hasil ekstraksi fitur menjadi probabilitas keluaran,

menentukan kelas akhir 0 sebagai tidak berpotensi dan 1 sebagai berpotensi sarang jentik nyamuk, dan menghubungkan seluruh fitur hasil ekstraksi menjadi keputusan akhir. Tujuannya menghasilkan nilai akhir yang mewakili Tingkat keyakinan model terhadap kategori citra yang diuji.

10. Struktur *DenseNet121*

Struktur *DenseNet121* dapat dijelaskan pada tabel berikut:

Tabel 4.1 Struktur *DenseNet121*

Komponen	Jumlah <i>Dense Layer</i>	Deskripsi
Dense Block 1	6	Mengekstraksi fitur dasar dari citra.
Transition Layer 1	-	Terdiri dari Batch Normalization, ReLU, 1×1 Convolution, dan Average Pooling 2×2 untuk mengurangi ukuran spasial dan <i>channel</i> fitur.
Dense Block 2	12	Mengekstraksi fitur menengah untuk pola visual yang lebih kompleks.
Transition Layer 2	-	Menghubungkan blok kedua dan ketiga serta melakukan kompresi dimensi fitur.
Dense Block 3	24	Mengekstraksi fitur tingkat tinggi (<i>high-</i>

		<i>level features</i>) dari hasil sebelumnya.
Transition Layer 3	-	Menghubungkan blok ketiga dengan blok keempat melalui proses normalisasi dan pooling.
Dense Block 4	16	Menghasilkan fitur akhir yang paling representatif terhadap citra masukan.

3.1.5 Evaluasi Model

Evaluasi terhadap performa model dilakukan untuk mengukur seberapa baik sistem dalam mengklasifikasikan citra lingkungan yang berpotensi maupun tidak berpotensi menjadi tempat sarang jentik nyamuk. Pada penelitian ini, proses evaluasi dilakukan menggunakan beberapa metrik, yaitu *accuracy*, *precision*, *recall*, *F1-Score*, *Confusion matrix*.

Accuracy digunakan untuk mengukur persentase jumlah prediksi yang benar terhadap keseluruhan data uji. Nilai *accuracy* dihitung dengan rumus sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP (*True Positive*) adalah jumlah citra berpotensi yang diprediksi dengan benar, TN (*True Negative*) adalah jumlah citra tidak berpotensi yang diprediksi dengan benar, FP (*False Positive*) adalah jumlah citra tidak berpotensi yang diprediksi salah sebagai berpotensi, dan FN (*False Negative*) adalah jumlah citra berpotensi yang diprediksi salah sebagai tidak berpotensi.

Selain *accuracy*, *F1-Score* juga digunakan untuk memberikan penilaian performa model yang lebih seimbang, khususnya pada *dataset* dengan

distribusi kelas yang tidak merata. *F1-Score* mempertimbangkan keseimbangan antara *presicion* dan *recall*, dengan rumus sebagai berikut:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Dengan:

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

Yang berarti tingkat ketepatan model dalam mengklasifikasikan data positif, dan

$$Recall = \frac{TP}{TP + FN}$$

Yang berarti kemampuan model dalam menemukan seluruh data yang seharusnya termasuk kedalam kelas positif.

Evaluasi dilakukan menggunakan data uji yang tidak dilibatkan dalam proses pelatihan, sehingga hasil evaluasi ini dapat memberikan gambaran umum mengenai kemampuan generalisasi model terhadap data baru.

Selain metrik tersebut, evaluasi performa model juga dilengkapi dengan *Confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang menyajikan distribusi jumlah prediksi benar dan salah untuk setiap kelas. Pada tabel 3.1, baris mewakili kelas sebenarnya (*Actual class*), sedangkan kolom mewakili kelas hasil prediksi (*Predicted class*). Dengan menggunakan *confusion matrix*, peneliti dapat mengidentifikasi kelas mana yang paling sering salah prediksi oleh model, sekaligus mengevaluasi sejauh mana kemampuan model dalam membedakan tiap kelas secara lebih detail.

Tabel 3. 1 Contoh Tabel *Confusion Matrix*

Actual/Predicted	Tidak Berpotensi	Berpotensi
Berpotensi	3	9
Tidak Berpotensi	9	3

Tabel 3.1 menunjukkan hasil *confusion matrix* yang diperoleh dari proses klasifikasi menggunakan model yang diusulkan. Tabel ini menggambarkan prediksi benar dan salah untuk dua kelas utama, yaitu berpotensi dan tidak berpotensi

Berdasarkan tabel 3.1 terlihat bahwa dari total sampel kelas berpotensi sebanyak 3 sampel berhasil diprediksi dengan benar, sedangkan 9 sampel lainnya diprediksi sebagai tidak berpotensi. Sebaliknya, pada kelas Tidak berpotensi terdapat 3 sampel yang diprediksi benar, sementara 9 sampel lainnya salah diklasifikasikan sebagai Berpotensi.

Hasil ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan kedua kelas dengan baik. Tingginya jumlah kesalahan klasifikasi dapat disebabkan oleh kemiripan karakteristik visual pada citra lingkungan yang memiliki kondisi mirip, baik yang benar-benar berpotensi menjadi tempat jentik nyamuk maupun yang tidak berpotensi. Oleh karena itu, diperlukan strategi peningkatan kinerja, seperti penambahan jumlah data latih, penerapan teknik augmentasi yang lebih beragam, atau penggunaan arsitektur model yang lebih kompleks untuk meningkatkan akurasi klasifikasi.

3.2 Analisa kebutuhan

Pada tahap ini, penulis melakukan analisis terhadap kebutuhan yang diperlukan selama proses pembangunan sistem klasifikasi tempat berpotensi jentik nyamuk berbasis *Deep Learning*. Analisis kebutuhan ini mencakup Bahasa pemrograman, perangkat lunak, serta *library* dan *framework* pendukung yang digunakan untuk mengembangkan sistem.

1. Bahasa pemrograman

a. *Python*

Bahasa pemrograman utama yang digunakan adalah *python*. Pemilihan *python* didasarkan pada kelengkapan ekosistemnya yang mendukung penelitian ini di bidang *machine learning*, *deep learning*, serta pengolahan citra digital. *Python* juga memiliki banyak *library*

yang relevan, mudah diintegrasikan, dan didukung oleh komunitas riset yang luas.

2. *Software*

a. Google Colab

Dimanfaatkan sebagai lingkungan komputasi berbasis *cloud* untuk menulis dan menjalankan kode *python*. Google colab juga mendukung penggunaan GPU secara gratis, sehingga mempercepat proses pelatihan *Deep Learning*.

b. Google Drive

Digunakan sebagai media penyimpanan dataset citra lingkungan dan hasil pelatihan model yang terintegrasi langsung dengan Google Colab.

c. Draw.io

Dimanfaatkan untuk membuat diagram alur sistem, visualisasi arsitektur jaringan, serta rancangan tahapan pengolahan citra.

d. *Visual Studio* (VsCode)

Digunakan untuk menulis, mengorganisasi dan menjalankan program berbasis *streamlit* secara lokal.

e. Figma

Dimanfaatkan untuk mendesain tampilan antarmuka aplikasi yang interaktif sebelum ke dalam *streamlit*.

f. Canva

Digunakan untuk membuat desain ilustratif, seperti bagan arsitektur sistem, alur pemrosesan dan visualisasi hasil penelitian.

3. *Library dan Framework*

a. *Tensorflow dan Keras*

Tensorflow digunakan sebagai *framework* utama dalam membangun dan melatih model *deep learning*. *Library* ini menyediakan berbagai komponen untuk membangun arsitektur jaringan, melakukan komputasi numerik berbasis tensor, serta

mendukung eksekusi GPU. *Keras* yang merupakan *high-level* API dari *TensorFlow* digunakan untuk memudahkan perancangan model dengan sintaks yang sederhana dan intuitif. Dalam penelitian ini, *Keras* dimanfaatkan untuk membangun arsitektur *Convolutional Neural Network* (CNN) dan *DenseNet*¹²¹ dengan menggunakan lapisan *Convolution*, *Pooling*, *Dense Layer*, serta *Softmax* sebagai fungsi aktivasi output.

b. *Scikit-Learn* (sklearn)

Digunakan pada tahap evaluasi model. *Library* ini menyediakan berbagai fungsi statistic untuk mengukur performa, diantaranya *classification report*, *confusion matrix*, serta perhitungan *precision*, *recall*, *F1-Score*, dan *accuracy*. Selain itu, *scikit-learn* juga digunakan untuk melakukan *train-test split* dan *compute class weight* dalam mengatasi distribusi data yang tidak seimbang.

c. *OpenCV* (cv2)

OpenCV digunakan untuk pemrosesan citra digital, termasuk membaca dataset gambar, mengubah ukuran (*Resize*), mengatur format warna, serta melakukan augmentasi sederhana. *Library* ini mendukung berbagai operasi manipulasi citra yang mempercepat tahapan *preprocessing* sebelum data dimasukkan kedalam model.

d. *NumPy*

NumPy berperan sebagai *library* fundamental untuk komputasi numerik dalam *python*. Dalam penelitian ini, *NumPy* digunakan untuk memanipulasi array multidimensi, melakukan perhitungan matematis dasar, serta mendukung operasi tensor yang dibutuhkan dalam proses pembelajaran mesin.

e. *Matplotlib* dan *Seaborn*

Keduanya digunakan untuk visualisasi dan hasil penelitian. *Matplotlib* dimanfaatkan untuk menampilkan grafik hasil pelatihan, seperti kurva *accuracy* dan *loss*. Sedangkan *Seaborn* digunakan untuk

membuat visualisasi statistic yang lebih interaktif, seperti *confusion matrix* dengan representasi heatmap.

f. PIL (*Python Imaging Library*)

Library PIL digunakan untuk membuka, membaca, dan menampilkan gambar, terutama dalam antarmuka pengguna berbasis *streamlit*. Fungsi ini memungkinkan pengguna untuk mengunggah gambar kemudian diproses lebih lanjut oleh model klasifikasi.

g. *Streamlit*

Streamlit dipakai untuk membangun antarmuka pengguna berbasis web. Dengan *framework* ini, sistem yang dikembangkan dapat dijalankan secara interaktif, memungkinkan pengguna mengunggah citra lingkungan dan langsung mendapatkan hasil klasifikasi apakah suatu wadah berpotensi menjadi tempat berkembangbiak jentik nyamuk atau tidak.

h. Modul pendukung lain

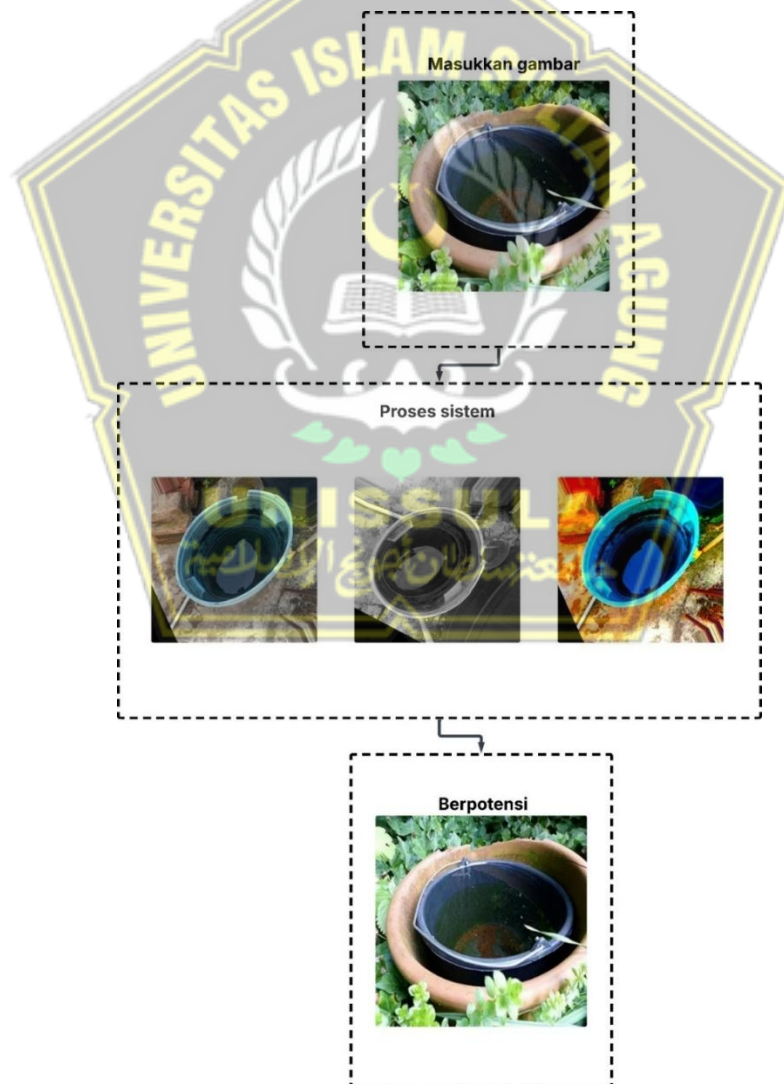
Selain *library* utama, penelitian ini juga menggunakan beberapa modul *python* pendukung, seperti:

- a) OS, digunakan untuk manajemen direktori dan file, seperti membaca *path dataset* atau menyimpan model hasil pelatihan
- b) Shutil, dimanfaatkan untuk menyalin, memindahkan atau menghapus file selama proses pengolahan dataset.
- c) *Random*, digunakan untuk melakukan pengacakan data, misalnya saat membagi *dataset* ke dalam kelompok latih dan uji agar distribusi lebih merata.
- d) *Warnings*, digunakan untuk mengabaikan peringatan (*Warning*) yang tidak berpengaruh terhadap hasil penelitian sehingga *output* kode lebih bersih.
- e) Glob, dipakai untuk membaca sekumpulan *file* gambar dalam satu *folder* dengan pola tertentu.

- f) *Time*, digunakan untuk mencatat waktu eksekusi proses pelatihan maupun *preprocessing*.

3.3 Implementasi sistem

Pada tahap ini dilakukan analisis untuk menentukan alur kerja implementasi sistem klasifikasi citra yang akan digunakan oleh pengguna. Alur ini disajikan dalam bentuk *workflow* untuk memperjelas tahapan interaksi antara pengguna dengan sistem, mulai dari proses input citra, pengolahan data oleh model, hingga keluaran hasil prediksi. Diagram alur implementasi sistem dapat dilihat pada Gambar 3.2



Gambar 3. 2 Alur Implementasi Sistem

Pada Gambar 3.2 ditunjukkan alur implementasi sistem deteksi tempat berpotensi menjadi sarang jentik nyamuk ketika digunakan oleh pengguna.

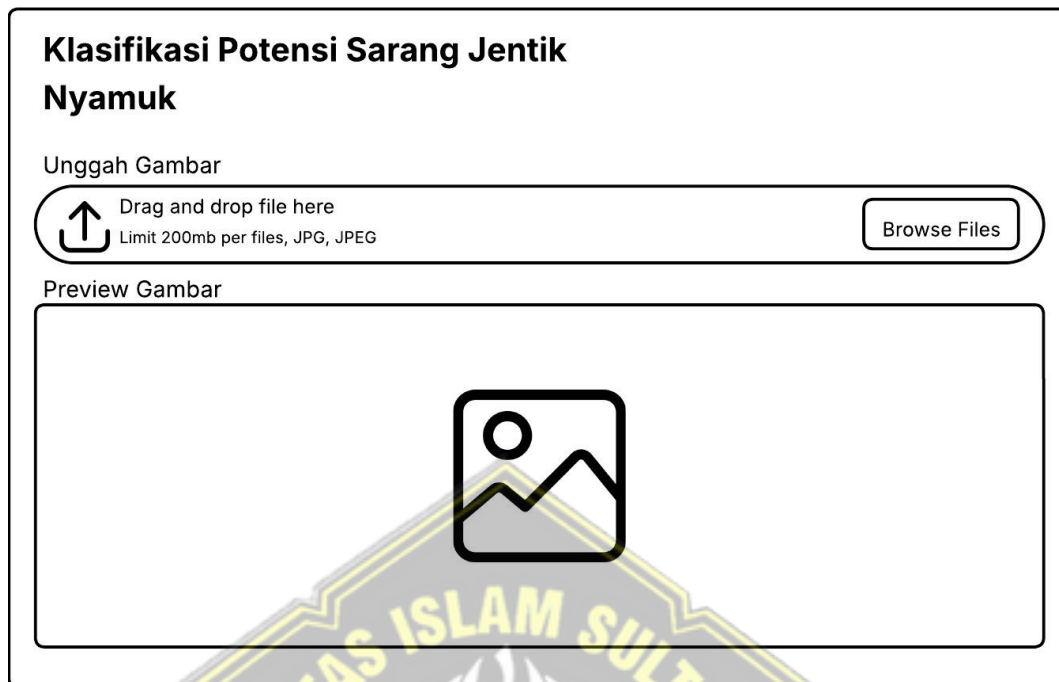
Alur tersebut terdiri dari beberapa tahapan, yaitu:

1. Pengguna membuka aplikasi sehingga akan muncul tampilan awal antarmuka berbasis web yang telah dirancang menggunakan *Streamlit*.
2. Pengguna mengunggah citra lingkungan yang ingin diperiksa. Proses unggah dapat dilakukan melalui tiga cara, yaitu *drag and drop file*, *browse file* dari perangkat.
3. Sistem melakukan pemrosesan citra. Tahap ini mencakup *preprocessing*, normalisasi, serta ekstraksi fitur visual dari objek dalam citra yang kemudian dianalisis oleh model *Deep Learning* untuk menentukan apakah objek tersebut berpotensi menjadi tempat berkembang biak jentik nyamuk.
4. Pengguna memperoleh hasil deteksi dalam bentuk keluaran pada antarmuka sistem, yaitu berupa informasi klasifikasi apakah citra yang dimasukkan termasuk kategori berpotensi menjadi sarang jentik nyamuk atau tidak berpotensi

3.4 Rancangan *User Interface*

3.4.1 Halaman Antarmuka Utama

Gambar berikut menampilkan rancangan awal dari antarmuka pada sistem klasifikasi tempat berpotensi jentik nyamuk. Rancangan ini memperlihatkan tampilan utama sistem yang difokuskan untuk mempermudah pengguna dalam melakukan unggah citra lingkungan, sebelum citra tersebut diproses lebih lanjut oleh model klasifikasi.



Gambar 3. 3 Halaman Awal Sistem *User Interface*

Pada gambar 3.3 ditampilkan rancangan antarmuka pengguna (*user interface*) pada halaman utama sistem klasifikasi potensi sarang jentik nyamuk. Halaman ini menjadi titik awal interaksi antara pengguna dengan sistem, di mana pengguna diarahkan untuk mengunggah citra lingkungan yang ingin diperiksa potensi keberadaannya sebagai sarang jentik nyamuk. Desain antarmuka dibuat sederhana sehingga dapat digunakan dengan mudah.

Pada bagian tengah halaman terdapat area unggah (*upload area*) yang dilengkapi instruksi *Drag and drop file here* bagi pengguna yang ingin langsung menyeret gambar ke area tersebut, atau tombol *Browse Files* yang dapat diklik untuk memilih *file* gambar secara *manual* dari perangkat yang digunakan.

3.4.2 Halaman Hasil dari Sistem

Gambar berikut merupakan rancangan awal tampilan antarmuka sistem klasifikasi sarang jentik nyamuk. Halaman ini menampilkan situasi setelah

citra lingkungan berhasil diunggah, di mana pengguna dapat melihat gambar yang siap dianalisis oleh sistem.

Hasil Prediksi:

Prediksi: **Berpotensi** Sarang Jentik Nyamuk

Kesimpulan : Berdasarkan analisis gambar, model mendeteksi adanya potensi sarang jentik nyamuk dengan probabilitas 0.9638.

Detail Probabilitas

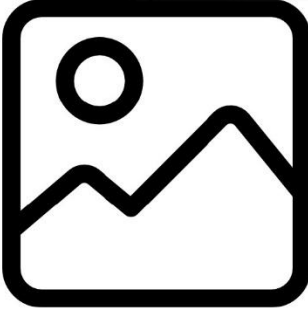
Berpotensi
96.38%

Tidak Berpotensi
3.62%

Gambar 3. 4 Halaman Hasil Sistem

Visualisasi Fokus Model (Grad-Cam)

Tampilkan HeatMap



Gambar 3. 5 Halaman Hasil Sistem Menggunakan Grad-Cam

Pada Gambar 3.4 dan 3.5 ditampilkan antarmuka sistem setelah citra lingkungan berhasil diunggah dan diproses oleh model klasifikasi. Antarmuka ini menyajikan hasil prediksi akhir dalam bentuk teks yang memuat informasi kelas hasil deteksi, yaitu Berpotensi Sarang Jentik Nyamuk, beserta tingkat probabilitas yang dihasilkan model. Selain itu, detail probabilitas ditampilkan dalam bentuk persentase untuk kedua kelas berpotensi dan tidak berpotensi, sehingga pengguna dapat memahami seberapa besar keyakinan sistem terhadap hasil klasifikasi yang diberikan. Tampilan ini dirancang dengan pendekatan sederhana namun informatif, guna memastikan pengguna dapat dengan mudah membaca serta menginterpretasikan hasil analisis tanpa hambatan teknis yang kompleks.



BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil

4.1.1 Pengumpulan Data

Pengumpulan dataset yaitu pengambilan citra yang dilakukan di berbagai lokasi yang memiliki potensi berbeda dalam menampung jentik nyamuk. Dataset yang digunakan dalam penelitian ini terbagi menjadi dua kelas seperti area dengan genangan air, saluran air tersumbat, pot bunga, dan tempat sampah terbuka. Sebagai pembandingan, gambar dari lingkungan yang tidak memiliki potensi menjadi sarang jentik, seperti tanah terbuka, pot tanaman hias, dan lapangan juga dikumpulkan.

Seluruh citra dalam dataset ini disimpan dalam format JPG karena memiliki ukuran relatif kecil dan kompatibilitas tinggi dengan deep learning seperti TensorFlow dan Keras. Format ini juga mendukung efisiensi dalam proses pemuatan data dan pelatihan model. Gambar dikumpulkan dan disimpan dengan resolusi 640x640 piksel untuk mempertahankan detail visual yang relevan seperti tekstur tanah, genangan air, dan benda di lingkungan sekitar. Ukuran ini cukup besar untuk menjaga kualitas informasi visual, namun tetap efisien untuk diproses. Untuk kelas nya sendiri terbagi menjadi dua, yaitu:

a. Kelas Berpotensi sarang jentik nyamuk

Citra yang termasuk kedalam kelas ini umumnya menunjukkan keberadaan genangan air yang tidak mengalir di area terbuka, seperti di dalam pot kosong, selokan tersumbat, dan genangan air yang tidak terpantau. Area ini bersifat lembap dan tidak mengalami perawatan rutin yang merupakan kondisi ideal bagi nyamuk untuk berkembang biak.

b. Kelas Tidak Berpotensi sarang jentik nyamuk

Citra dalam kategori ini menampilkan area lingkungan yang bersih, kering atau tidak memiliki wadah air terbuka. Misalnya pot bunga yang berisi tanaman dengan tanah padat, permukaan tanpa genangan atau saluran air yang mengalir deras.

Tabel 4. 2 Sample dataset citra lingkungan potensi nyamuk

Kelas	Sample Citra Lingkungan
<p>Berpotensi Sarang Jentik Nyamuk</p>	
<p>Tidak Berpotensi Sarang Jentik Nyamuk</p>	

Pada Tabel 4.1 ditunjukkan proses pengumpulan data citra lingkungan yang digunakan dalam penelitian ini. Dataset terdiri dari dua kategori utama, yaitu citra lingkungan yang berpotensi menjadi sarang jentik nyamuk dan citra lingkungan yang tidak berpotensi menjadi sarang jentik nyamuk. Setiap kategori disusun dalam direktori terpisah untuk memudahkan proses

pelatihan, dan evaluasi model. Adapun deskripsi dari masing-masing kategori adalah sebagai berikut:

1. Berpotensi Sarang Jentik Nyamuk

Kategori ini mencakup citra lingkungan yang memiliki kondisi mendukung perkembangbiakan nyamuk, seperti adanya genangan air, wadah berisi air terbuka, selokan tidak mengalir, maupun tempat penampungan air yang tidak tertutup. Ciri utama pada kelas ini adalah adanya potensi bagi air untuk tergenang dalam waktu lama sehingga memungkinkan larva nyamuk berkembang biak.

2. Tidak Berpotensi Sarang Jentik Nyamuk

Kategori ini berisi citra lingkungan yang tidak memiliki kondisi yang mendukung pertumbuhan jentik nyamuk. Contohnya adalah pot tanaman yang tidak menampung air, area kering, permukaan tanah atau semen yang tidak menahan air, serta wadah tertutup. Lingkungan pada kelas ini relatif aman dari risiko menjadi tempat perkembangbiakan nyamuk.

4.2 Pengujian Model

4.2.1 Pengolahan Awal Citra *Dataset*

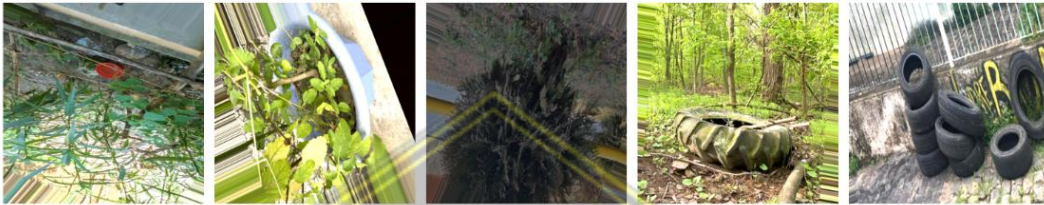
Gambar berikut menampilkan hasil pengolahan awal pada *dataset* citra lingkungan yang meliputi proses *resize* gambar menjadi 640×640 piksel, normalisasi piksel ke skala 0–1, augmentasi data seperti *flipping*, *zoom*, rotasi, dan penyesuaian kecerahan.

Contoh Gambar Setelah Augmentasi

Kelas: 1_berpotensi



Kelas: 0_tidak



Gambar 4. 2 Contoh Gambar Setelah Augmentasi

Pada Gambar 4.2 ditunjukkan contoh hasil pengolahan awal, di mana citra telah diseragamkan ukurannya, dinormalisasi, serta diperluas variasinya melalui augmentasi agar mencegah *overfitting* dan siap digunakan pada tahap pelatihan model klasifikasi.

4.2.2 Tahap Pembelajaran Model CNN berbasis *DenseNet121*

Tabel berikut menyajikan konfigurasi *hyperparameter* yang digunakan pada model CNN (*Convolutional Neural Network*) dengan arsitektur *DenseNet121*. *Hyperparameter* ini ditetapkan untuk mengoptimalkan proses pembelajaran, meliputi pengaturan *learning rate*, *optimizer*, fungsi aktivasi, jumlah *epoch*, ukuran *batch*, serta mekanisme *early stopping* untuk mencegah *overfitting*.

Tabel 4. 3 *Hyperparameter Model*

No.	<i>Hyperparameter</i>	Arti Nilai	Keterangan
1.	$Target_size = (640, 640, 3)$	Semua gambar diubah menjadi 640x640 piksel dengan 3 <i>channel</i> warna (RGB)	<i>DenseNet121</i> membutuhkan input resolusi tinggi supaya fitur detail bisa terdeteksi

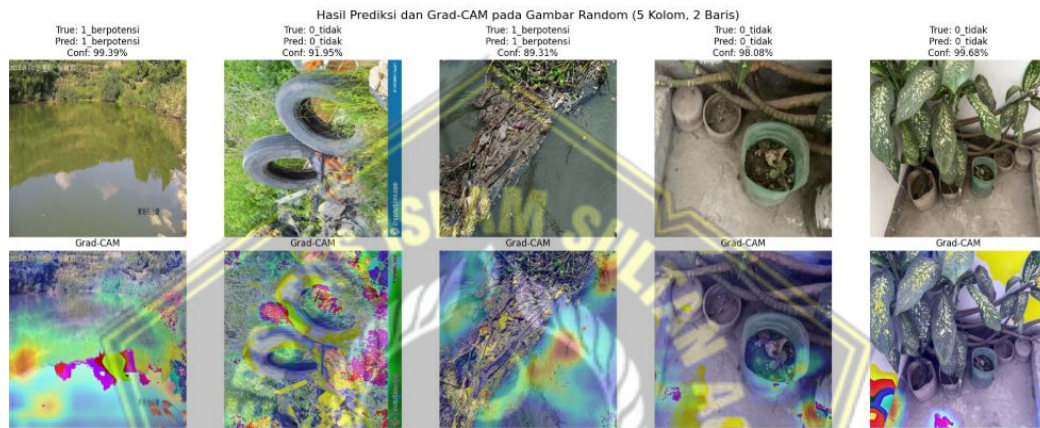
2.	<i>Batch_size = 16</i>	Model membaca 16 gambar sekaligus sebelum memperbarui bobotnya.	Ukuran ini dipilih karena seimbang antara efisiensi memori GPU dan kecepatan pelatihan
3.	<i>Class_mode = 'binary'</i>	Nilai ' <i>binary</i> ' berarti setiap gambar dikasih label 0 atau 1 mewakili dua kelas yang ada, yaitu berpotensi dan tidak berpotensi	Menentukan format label yang dihasilkan oleh generator data <i>biner</i> untuk 2 kelas
4.	<i>Optimizer = Adam</i>	<i>Adam (Adaptive Moment Estimation)</i> menggabungkan kecepatan dan kestabilan agar pelatihan lebih stabil dan cepat mencapai hasil optimal	Algoritma yang digunakan untuk memperbarui bobot model berdasarkan gradien loss function, bertujuan meminimalkan loss.
5.	<i>Learning_rate = 0.00001</i>	Nilai 0.00001 artinya model belajar dengan langkah yang sangat kecil dan stabil. Supaya tidak lompat saat mencari nilai terbaik	<i>Learning rate</i> kecil lebih cocok agar tidak merusak bobot awal yang sudah bagus.
6.	<i>Loss = Binary_crossentropy</i>	Menghitung jarak antara label asli	Karena klasifikasi hanya dua kelas, loss ini paling

		(0/1) dan hasil prediksi sigmoid	cocok digunakan dalam penelitian ini
7.	<i>Metrics = 'accuracy'</i>	Menghitung rasio antara jumlah prediksi benar dengan total sampel	Metrik akurasi paling mudah dipahami dan menggambarkan performa keseluruhan model
8.	<i>Activation (Output layer) = Sigmoid</i>	<i>Sigmoid</i> mengubah hasil perhitungan model menjadi probabilitas, nilai mendekati 1 berarti berpotensi, mendekati 0 berarti tidak berpotensi	<i>Sigmoid</i> paling cocok dibanding Softmax. <i>Sigmoid</i> sederhana, efisien, dan langsung bisa dikonversi ke dua kelas
9.	<i>Dropout (0.5)</i>	Nilai 0.5 artinya 50% neuron di nonaktifkan acak tiap langkah proses pelatihan	Nilai 0.5 sering digunakan karena dianggap sebagai keseimbangan yang pas antara mencegah overfitting dan tetap menjaga kemampuan model untuk belajar dengan baik.
10.	<i>Epoch = 100</i>	Model akan melihat seluruh dataset 100 kali untuk menyempurnakan hasilnya	100 <i>Epoch</i> dipilih agar model punya waktu belajar cukup lama untuk mencapai hasil maksimal dan menemukan pola fitur yang lebih kompleks pada lingkungan.

11.	<i>Callbacks Used = EarlyStopping, ModelCheckpoint</i>	<i>EarlyStopping</i> untuk menghentikan pelatihan otomatis jika model tidak membaik lagi. <i>ModelCheckpoint</i> untuk menyimpan model terbaik selama pelatihan	<i>EarlyStopping</i> dipilih karena dapat menghindari overfitting dan menghemat waktu. Jika validasi tidak membaik selama 10 <i>epoch</i> , pelatihan berhenti, dan bobot terbaik dikembalikan. <i>ModelCheckpoint</i> dipilih karena menjamin model terbaik tersimpan saat validasi akurasi paling tinggi
12.	<i>Patience</i>	10	Jumlah <i>epoch</i> tanpa peningkatan pada metrik monitor yang ditoleransi sebelum menghentikan pelatihan lebih awal.
13.	<i>Monitor</i>	' <i>val_loss</i> '	Metrik <i>loss</i> validasi yang dipantau untuk menentukan kapan menghentikan pelatihan atau menyimpan model.
14.	<i>Restore_best_weights</i>	<i>True</i>	Mengembalikan bobot model ke keadaan terbaik (<i>loss</i> validasi terendah) saat <i>Early Stopping</i> terpicu.
15.	<i>Save_best_only</i>	<i>True</i>	Hanya menyimpan model jika kinerja pada metrik

			monitor membaik dibandingkan sebelumnya.
--	--	--	--

Pada tabel 4.2 ini merangkum *hyperparameter* model dalam proses pelatihan yang digunakan dalam sistem. Setiap *hyperparameter* dalam tabel mencantumkan nilai yang digunakan dan penjelasan singkat mengenai fungsi atau tujuannya dalam membangun dan melatih model klasifikasi.



Gambar 4. 3 Hasil Prediksi *Data Test*

Pada Gambar 4.3 ditampilkan hasil visualisasi prediksi model CNN berbasis *DenseNet121* terhadap beberapa citra uji beserta interpretasi Grad-CAM. Setiap citra dilengkapi dengan informasi prediksi (Pred) yang dihasilkan model dan label sebenarnya (True). Kategori yang digunakan terdiri dari berpotensi dan tidak berpotensi sebagai sarang jentik nyamuk.

Selain menampilkan prediksi, disertakan pula heatmap Grad-CAM yang menunjukkan area citra yang paling berpengaruh dalam keputusan klasifikasi. Grad-CAM membantu memberikan interpretasi visual mengenai fokus perhatian model, misalnya pada area wadah berisi air atau permukaan lingkungan yang dianggap relevan sebagai potensi sarang nyamuk. Dari visualisasi ini terlihat bahwa sebagian besar prediksi sesuai dengan label sebenarnya, meskipun masih terdapat beberapa kesalahan akibat kesamaan pola visual atau kompleksitas latar belakang citra.

4.3 Evaluasi Model

Evaluasi model dilakukan dengan memanfaatkan empat metrik utama, yakni *accuracy*, *precision*, *recall*, dan *F1-Score*, sebagai indikator menyeluruh terhadap kinerja model. *Accuracy* digunakan untuk mengetahui proporsi prediksi yang benar dibandingkan dengan keseluruhan data uji. Walaupun metrik ini mampu memberikan gambaran umum performa, *accuracy* terkadang kurang mewakili kondisi sebenarnya jika distribusi kelas tidak seimbang.

Sebagai pelengkap, digunakan metrik *precision* yang menggambarkan kemampuan model dalam meminimalkan kesalahan pada klasifikasi positif. *Precision* dihitung dari perbandingan antara jumlah prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan. Dengan demikian, *precision* menekankan seberapa tepat model dalam menetapkan sebuah citra ke dalam kelas tertentu, yang sangat relevan apabila kesalahan positif membawa konsekuensi signifikan.

Metrik *recall* digunakan untuk menilai sejauh mana model berhasil mengenali seluruh sampel positif yang terdapat pada *dataset*. *Recall* yang tinggi berarti model tidak banyak melewatkan data yang seharusnya dikategorikan positif. Untuk memperoleh penilaian yang lebih seimbang, digunakan *F1-Score* yang merupakan rata-rata harmonis antara *precision* dan *recall*. Nilai *F1-Score* yang tinggi menunjukkan bahwa model tidak hanya akurat, tetapi juga konsisten dalam mengenali kelas target dengan tingkat kesalahan yang rendah.

Berdasarkan hasil pengujian, model menunjukkan performa yang memuaskan dengan *accuracy* 98%, *precision* 95%, *recall* 100%, dan *F1-Score* 98%. Selain evaluasi numerik, dilakukan pula analisis visual melalui perbandingan hasil prediksi dengan label sebenarnya pada sejumlah citra validasi. Dari hasil tersebut terlihat bahwa mayoritas citra berhasil diklasifikasikan secara tepat, meskipun masih terdapat sedikit kesalahan akibat kemiripan pola visual maupun kompleksitas latar belakang. Secara

keseluruhan, sistem ini dapat diandalkan untuk mengidentifikasi lingkungan yang berpotensi ataupun tidak berpotensi menjadi sarang jentik nyamuk.

1. *Classification Report*

Classification Report digunakan sebagai acuan untuk menilai kinerja model klasifikasi secara lebih detail. Penelitian ini menyajikan nilai *accuracy*, *precision*, *recall*, *F1-Score*, dan support pada masing-masing kelas, sekaligus menampilkan nilai akurasi keseluruhan. Dengan adanya penelitian ini, performa model dapat dianalisis tidak hanya dari sisi ketepatan prediksi total, tetapi juga dari keseimbangan kemampuan model dalam mengenali setiap kelas.

Tabel 4. 4 Tabel *Classification Report*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Accuracy</i>
Berpotensi	1.00	0.95	0.97	0.97
Tidak Berpotensi	0.95	1.00	0.98	0.97

Pada tabel 4.4 menampilkan hasil evaluasi model menggunakan *classification report*. Pada kelas Berpotensi, model memperoleh nilai *precision* sebesar 1.00, artinya seluruh prediksi positif yang diberikan model benar-benar sesuai dengan label aslinya. Nilai *recall* pada kelas ini sebesar 0.95, yang menunjukkan bahwa 95% dari seluruh data berpotensi berhasil terdeteksi dengan benar, sedangkan sisanya masih terlewat. Kombinasi *precision* dan *recall* menghasilkan *F1-Score* sebesar 0.97, yang menandakan kinerja model sangat baik pada kelas ini.

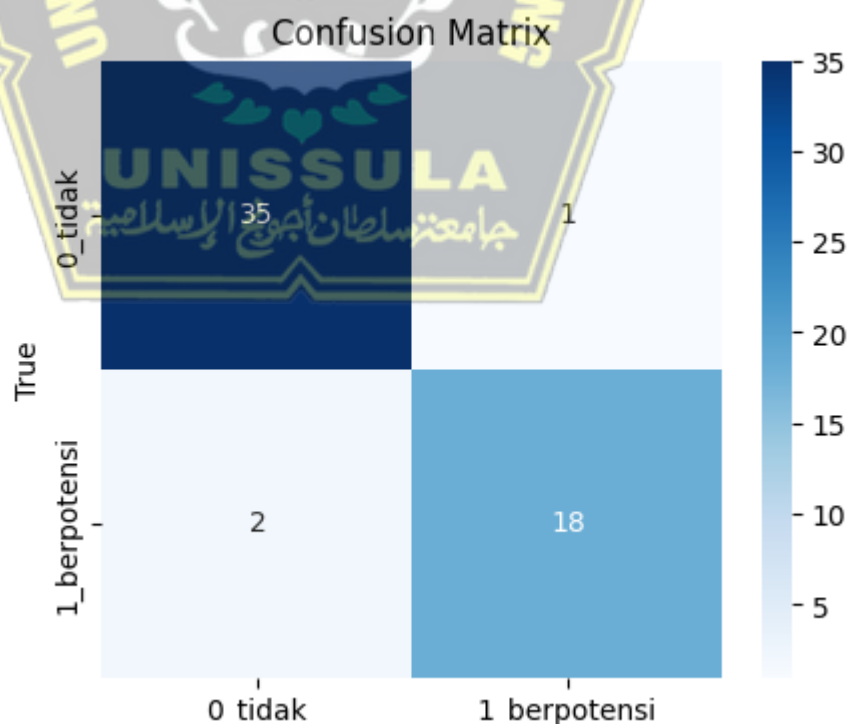
Untuk kelas Tidak Berpotensi, model mencapai *precision* 0.95, artinya terdapat sebagian kecil prediksi yang salah diklasifikasikan. Namun, *recall* pada kelas ini mencapai 1.00, yang berarti semua data yang benar-benar tidak berpotensi berhasil dikenali dengan baik oleh model. *F1-Score* untuk kelas ini adalah 0.98, menegaskan keseimbangan antara ketepatan dan kelengkapan deteksi.

Secara keseluruhan, model memperoleh akurasi sebesar 0.97 atau 97% dari total 40 data uji, yang berarti 97% prediksi sesuai dengan label sebenarnya. Nilai *macro average* dan *weighted average* masing-masing sebesar 0.97–0.98, menunjukkan bahwa performa model konsisten pada kedua kelas dan tidak berat sebelah terhadap distribusi data.

Hasil ini membuktikan bahwa model CNN berbasis *DenseNet121* mampu melakukan klasifikasi citra lingkungan dengan tingkat akurasi yang sangat tinggi, baik untuk kategori berpotensi maupun tidak berpotensi menjadi sarang jentik nyamuk.

2. *Confusion matrix*

Confusion matrix berfungsi sebagai alat evaluasi performa model klasifikasi dengan menampilkan jumlah prediksi yang benar (*True Positive* dan *true negative*) serta prediksi yang keliru (*False Positive* dan *false negative*) dalam bentuk matriks



Gambar 4. 4 *Confusion Matrix*

Confussion matrix pada gambar 4.4 menunjukkan hasil dari pengujian, berikut penjelasannya:

- a. Sebanyak 18 data dengan label sebenarnya berpotensi berhasil diprediksi dengan benar sebagai berpotensi (*True Positive*).
- b. Sebanyak 35 data dengan label sebenarnya tidak berpotensi berhasil diprediksi dengan benar sebagai tidak berpotensi (*True Negative*).
- c. Terdapat 2 data berpotensi yang salah diprediksi sebagai tidak berpotensi (*False Negative*).
- d. Terdapat 1 data tidak berpotensi yang salah prediksi sebagai berpotensi (*False Positive*).

4.4 Implementasi Model

4.4.1 Fitur Utama

Aplikasi antarmuka yang dibangun menggunakan Streamlit menyediakan sejumlah fitur utama yang dirancang untuk memudahkan pengguna dalam melakukan deteksi potensi sarang jentik nyamuk. Fitur-fitur tersebut meliputi:

1. Unggah citra lingkungan.

Pengguna dapat memilih dan mengunggah file gambar berformat .jpg, .jpeg, atau .png langsung dari perangkat. Aplikasi otomatis membaca *file* tersebut untuk kemudian diproses oleh model.

2. Pratinjau citra yang diunggah.

Setiap gambar yang dipilih ditampilkan kembali di halaman antarmuka agar pengguna dapat memastikan citra yang digunakan sudah sesuai sebelum dilakukan analisis lebih lanjut.

3. Prediksi potensi sarang jentik nyamuk.

Model CNN berbasis *DenseNet121* melakukan klasifikasi citra ke dalam dua kategori, yaitu berpotensi atau tidak berpotensi. Hasil prediksi ditampilkan langsung disertai tingkat probabilitasnya.

4. Tampilan metrik probabilitas.

Sistem menampilkan detail probabilitas masing-masing kelas dalam bentuk persentase, sehingga pengguna dapat mengetahui tingkat keyakinan model terhadap hasil prediksi.

5. Visualisasi Grad-CAM.

Aplikasi dilengkapi fitur interpretasi visual berupa *heatmap* Grad-CAM yang memperlihatkan area citra mana yang menjadi fokus utama model dalam menentukan prediksi. Area berwarna merah/oranye menunjukkan bagian yang paling berpengaruh terhadap keputusan klasifikasi.

4.4.2 Hasil Implementasi

1. Halaman Utama

Halaman utama aplikasi menampilkan judul sistem beserta penjelasan singkat mengenai tujuan penelitian. Aplikasi ini dikembangkan untuk mengidentifikasi potensi sarang jentik nyamuk pada citra lingkungan menggunakan *Convolutional Neural Network* (CNN) berbasis *DenseNet121*.



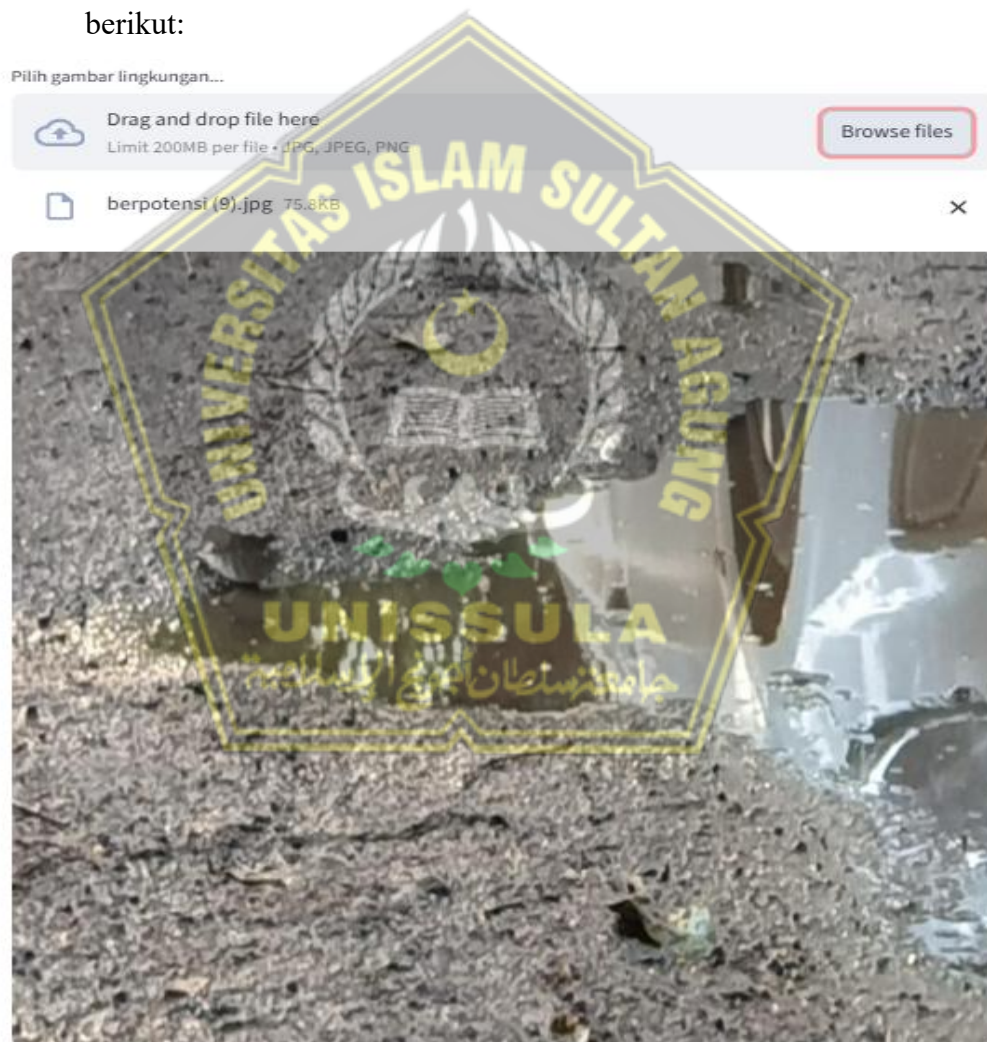
Gambar 4. 5 Tampilan *Home Streamlit*

Pada Gambar 4.6 ditampilkan halaman utama aplikasi dengan judul *Klasifikasi Potensi Sarang Jentik Nyamuk*. Halaman ini juga dilengkapi penjelasan singkat mengenai tujuan sistem, yaitu untuk mengidentifikasi

potensi sarang jentik nyamuk pada citra lingkungan menggunakan model *DenseNet121*.

2. Implementasi sebelum *upload* citra

Pada menu ini, pengguna dapat mengunggah citra lingkungan dalam format *.jpg*, *.jpeg*, maupun *.png*. Setelah file dipilih, sistem akan menampilkan pratinjau gambar di layar agar pengguna dapat memastikan citra yang digunakan sudah benar sebelum diproses lebih lanjut. Adapun tampilan antarmuka sebelum proses *upload* ditunjukkan pada gambar berikut:



Gambar 4. 6 Tampilan saat citra di input ke sistem

Pada Gambar 4.7 menampilkan tampilan *Streamlit* saat pengguna melakukan proses *upload* citra lingkungan pada sistem deteksi potensi

sarang jentik nyamuk. Tampilan ini menunjukkan halaman antarmuka yang memfasilitasi pengguna untuk mengunggah *file* gambar lingkungan yang akan dianalisis. Fitur utama pada halaman ini adalah komponen *File Uploader*, di mana pengguna dapat melakukan *drag and drop*, memilih *file* gambar secara manual menggunakan tombol *Browse Files*, atau mengambil gambar langsung dari perangkat. Sistem ini mendukung format *file* citra seperti JPG, JPEG, dan PNG, dengan batas ukuran maksimum 200MB per *file*, sehingga memberikan fleksibilitas dalam proses *input data*.

Secara keseluruhan, tampilan ini tidak hanya menekankan aspek fungsional sistem, tetapi juga menunjukkan kesiapan aplikasi dalam melakukan deteksi berbasis web, tanpa perlu instalasi perangkat tambahan. Antarmuka yang sederhana namun ramah pengguna ini menjadi bagian penting dari implementasi model CNN berbasis *DenseNet121* untuk klasifikasi lingkungan berpotensi maupun tidak berpotensi menjadi sarang jentik nyamuk.

3. Implementasi setelah *upload* citra

Setelah pengguna berhasil mengunggah citra lingkungan, sistem secara otomatis memproses gambar tersebut menggunakan model CNN berbasis *DenseNet121*. Proses ini melibatkan ekstraksi fitur citra yang kemudian diklasifikasikan ke dalam dua kategori, yaitu berpotensi atau tidak berpotensi sebagai sarang jentik nyamuk. Hasil prediksi ditampilkan dalam bentuk label klasifikasi yang informatif, disertai dengan nilai probabilitas pada masing-masing kelas yang memberikan gambaran tingkat keyakinan model terhadap keputusan yang dihasilkan. Adapun tampilan antarmuka setelah proses upload dapat dilihat pada gambar berikut:

Hasil Prediksi:

Prediksi: Berpotensi Sarang Jentik Nyamuk

Kesimpulan: Berdasarkan analisis gambar, model mendeteksi adanya potensi sarang jentik nyamuk dengan probabilitas 0.9104.

Detail Probabilitas

Berpotensi

91.04%

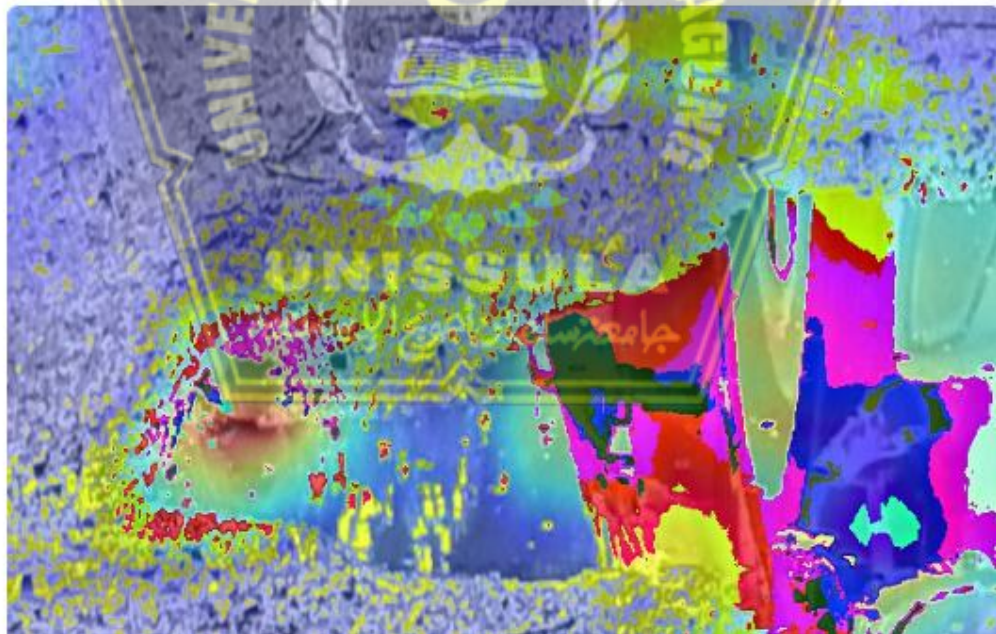
Tidak Berpotensi

8.96%

Visualisasi Fokus Model (Grad-CAM)

Heatmap di bawah ini menunjukkan area pada gambar yang paling penting bagi model untuk membuat prediksi. Area berwarna merah/oranye menunjukkan fokus model yang paling kuat.

Tampilkan Heatmap



Gambar 4. 7 Tampilan hasil klasifikasi setelah *upload Citra*

Pada Gambar 4.8 ditampilkan hasil *output* sistem setelah citra lingkungan diunggah. Model CNN berbasis *DenseNet121* memberikan prediksi Berpotensi dengan probabilitas sebesar 91.04%, sedangkan kelas Tidak Berpotensi memperoleh 8.96%. Selain label klasifikasi dan nilai

probabilitas, ditampilkan pula visualisasi Grad-CAM yang menunjukkan area citra yang menjadi fokus utama model dalam menentukan keputusan. Warna merah atau oranye pada *heatmap* menandakan bagian dengan pengaruh paling kuat terhadap prediksi, sehingga membantu pengguna memahami dasar pengambilan keputusan model sekaligus meningkatkan transparansi hasil klasifikasi.

4.5 Pembahasan Implementasi

Implementasi sistem deteksi potensi sarang jentik nyamuk menggunakan *Convolutional Neural Network* (CNN) berbasis *DenseNet121* telah berhasil direalisasikan secara menyeluruh. Sistem dirancang untuk memproses citra lingkungan secara otomatis, mulai dari tahap *preprocessing*, ekstraksi fitur, hingga menghasilkan prediksi kategori potensi sarang jentik nyamuk. Proses klasifikasi berjalan dengan baik, di mana arsitektur *DenseNet121* mampu mengekstraksi pola visual secara efisien serta menghubungkannya dengan hasil klasifikasi yang akurat.

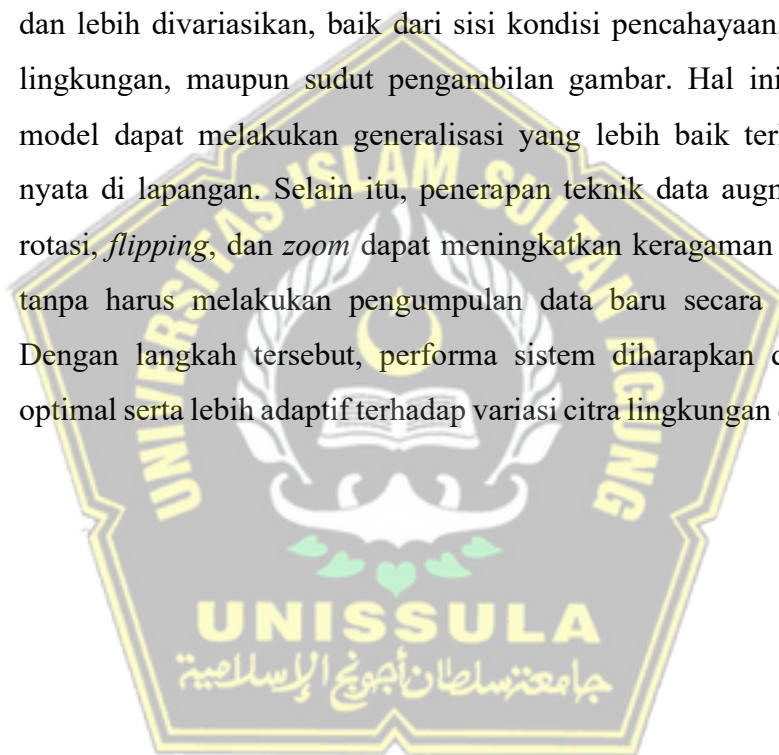
Kinerja model menunjukkan performa yang cukup baik, dengan capaian *accuracy* sebesar 97%, *precision* 100%, *recall* 95%, dan *F1-Score* 97%. Nilai metrik tersebut mengindikasikan bahwa model memiliki kemampuan generalisasi yang baik terhadap data uji, sehingga dapat diandalkan untuk mengenali citra lingkungan yang sebelumnya belum pernah diproses. Dengan demikian, sistem mampu mendeteksi potensi sarang jentik nyamuk dengan ketepatan yang tinggi dan konsistensi hasil yang stabil.

Dari sisi implementasi model, integrasi dengan Streamlit sebagai antarmuka berbasis web memberikan nilai tambah yang signifikan. Sistem memungkinkan pengguna untuk mengunggah citra lingkungan dan memperoleh hasil klasifikasi secara *real-time* melalui *browser*, tanpa memerlukan instalasi perangkat lunak tambahan. Antarmuka yang

dikembangkan bersifat ringan, interaktif, serta mudah digunakan, sehingga dapat diakses oleh pengguna dari berbagai latar belakang.

Secara keseluruhan, sistem yang dibangun telah memenuhi tujuan penelitian, yakni menyediakan alat bantu berbasis kecerdasan buatan yang dapat mendeteksi potensi sarang jentik nyamuk secara otomatis, akurat, dan praktis. Dengan adanya sistem ini, diharapkan dapat membantu masyarakat, tenaga kesehatan, maupun peneliti lapangan.

Sebagai pengembangan ke depan, disarankan agar *dataset* diperluas dan lebih divariasikan, baik dari sisi kondisi pencahayaan, variasi objek lingkungan, maupun sudut pengambilan gambar. Hal ini penting agar model dapat melakukan generalisasi yang lebih baik terhadap kondisi nyata di lapangan. Selain itu, penerapan teknik data augmentasi seperti rotasi, *flipping*, dan *zoom* dapat meningkatkan keragaman data pelatihan tanpa harus melakukan pengumpulan data baru secara besar-besaran. Dengan langkah tersebut, performa sistem diharapkan dapat semakin optimal serta lebih adaptif terhadap variasi citra lingkungan di dunia nyata.



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi potensi sarang jentik nyamuk berbasis *Convolutional Neural Network* (CNN) dengan arsitektur *DenseNet121* telah berhasil dibangun dan berfungsi dengan baik. Sistem ini mampu mengklasifikasikan citra lingkungan ke dalam dua kategori, yaitu berpotensi dan tidak berpotensi sebagai sarang jentik nyamuk melalui tahapan *preprocessing*, ekstraksi fitur, hingga klasifikasi secara otomatis. Hasil pengujian menunjukkan kinerja model yang cukup baik, dengan capaian *accuracy* sebesar 98%, *precision* 95%, *recall* 100%, dan *F1-Score* 98%, yang menandakan bahwa model memiliki kemampuan generalisasi yang baik terhadap data uji dan dapat diandalkan dalam mendeteksi kondisi lingkungan yang belum pernah dianalisis sebelumnya.

5.2 Saran

Walaupun sistem yang dibangun telah menunjukkan performa yang baik, masih terdapat beberapa aspek yang dapat dikembangkan lebih lanjut. Sistem saat ini belum mampu membedakan citra lingkungan luar ruangan dan dalam ruangan, sehingga kemampuan tersebut perlu ditambahkan agar hasil klasifikasi lebih informatif. Selain itu, pemanfaatan arsitektur lain seperti *YOLO (You Only Look Once)* disarankan untuk meningkatkan kecepatan dan akurasi deteksi secara *real-time*. Pengembangan selanjutnya juga perlu memperluas dan memperkaya variasi *dataset* agar model lebih adaptif terhadap kondisi lapangan, serta memanfaatkan augmentasi data untuk menambah keragaman data latih tanpa pengumpulan baru yang besar. Terakhir, sistem dapat diarahkan ke bentuk aplikasi *mobile* agar lebih praktis digunakan langsung di lapangan dengan dukungan kamera ponsel.

DAFTAR PUSTAKA

- Affifah, D. D., Permanasari, Y., Matematika, P., Matematika, F., Alam, P., & Bandung, U. I. (2022). *Teknik Konvolusi pada Deep Learning untuk Image Processing Pencermiran Terhadap Sumbu ? Pencermiran Terhadap Sumbu ?* 103–112.
- Alwy, S. M. (2025). Klasifikasi penyakit daun kopi arabika berbasis gambar menggunakan model Convolutional Neural Networks DenseNet121. *Klasifikasi penyakit daun kopi arabika berbasis gambar menggunakan model Convolutional Neural Networks DenseNet121*, 6(2), 881–890.
- Anhar, Anhar Putra, R. A. (2023). Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN). *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(2), 466. <https://doi.org/10.26760/elkomika.v11i2.466>
- Awalia, N., & Primajaya, A. (2022). Identifikasi Penyakit Leaf Mold Daun Tomat Menggunakan Model DenseNet-121. *Jurnal Ilmiah Ilmu Komputer*, 8(1), 49–54. <http://ejournal.fikom-unasman.ac.id>
- Falah, F., & Mulyono, S. (2024). *Implementasi Arsitektur Densely Connected Convolutional Networks (DenseNet121) Untuk Sistem Klasifikasi Penyakit Cacar*. 8(2), 253–258.
- Haidy Massa, M., Ould Lemrabott, M. A., Abdillahi Guedi, O., Briolant, S., & Ould Mohamed Salem Boukhary, A. (2025). Physical and Chemical Characteristics of *Aedes aegypti* Larval Habitats in Nouakchott, Mauritania. *Tropical Medicine and Infectious Disease*, 10(6), 1–11. <https://doi.org/10.3390/tropicalmed10060147>
- Hermawan, H. M., & Muhammad. (2024). *Sistem Pendeteksi Pneumonia Menggunakan Metode Convolutional Neural Network dengan Arsitektur DenseNet201*. Universitas Islam Sultan Agung Semarang.
- Ilyas, M., Setyaningrum, E., Mumtazah, D. F., & Busman, H. (2025). *Tempat Perindukan dan Kepadatan Larva Aedes spp . sebagai Indikator Risiko Penularan Demam Berdarah di Tempat Wisata Bandar Lampung Breeding*

Sites and Density of Aedes spp . Larvae as Indicators of Dengue Fever Transmission Risk in Bandar Lampung Tourist Attractions. 8, 39–50.
<https://doi.org/10.32528/bioma.v10i1.2701>

Jiwani, F. A., & Bagus, S. (2025). *Sistem Deteksi Pemalsuan Citra Menggunakan CNN DenseNet-121 dan Watermark Least Significant Bit (LSB)*. Universitas Islam Sultan Agung Semarang.

MathWorks. (2025). *Convolution*.
<https://www.mathworks.com/discovery/convolution.html>

Nainggolan, P. I., Efendi, S., Budiman, M. A., Lydia, M. S., Joshua, I., Bukit, D. S., Rahmat, R. F., Sinaga, L. A., & Subroto, E. (2023). Classification Of Aedes Mosquito Larva Using Convolutional Neural Networks And Extreme Learning Machine. *2023 7th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 79–83.

Pramesti, V. A. (2024). *Klasifikasi Penyakit Batu Ginjal Melalui Citra Ct Scan Dengan Penerapan Metode Convolutional Neural Network Arsitektur Densenet-121*. Universitas Islam Sultan Agung Semarang.

Putri, D. L. R., & Supatman. (2025). Klasifikasi Citra Jenis Hijab Menggunakan Densenet-121. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(1).

Putri, F. A. P. R., Tanjung, J. P., & Dharshini, N. P. (2025). Implementation of Transfer Learning on CNN using DenseNet121 and ResNet50 for Brain Tumor Classification. *Journal Of Informatics And Telecommunication Engineering*, 8(2), 325–334.

Saputra, A. B., Pamungkas, D. P., & Widodo, D. W. (2024). Rancangan Sistem Identifikasi Citra Penyakit dan Hama Bawang Merah Menggunakan Metode CNN model Densenet 201. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 8(2), 775–780.

Saputra, A. D., Hindarto, D., & Santoso, H. (2023). Disease Classification on Rice Leaves using DenseNet121, DenseNet169, DenseNet201. *Sinkron*, 8(1), 48–55. <https://doi.org/10.33395/sinkron.v8i1.11906>

Sinum, I. M. (2022). *Identifikasi dan analisis kepadatan nyamuk Anopheles sp. yang berpotensi sebagai vektor malaria berdasarkan lingkungan Kecamatan*

Teluk Pandan, Kabupaten Pesawaran.

- Solehudin, M. A., Gerhana, Y. A., & Taufik, I. (2025). *Klasifikasi Penyakit Daun Kopi Arabika Berbasis Gambar Menggunakan Model Convolutional Neural Networks DenseNet121*. *6(2)*, 881–891. <https://doi.org/10.47065/josh.v6i2.6407>
- Taha, M. F., Abdalla, A., Elmasry, G., Gouda, M., Zhou, L., Zhao, N., Liang, N., Niu, Z., Hassanein, A., Al-Rejaie, S., He, Y., & Qiu, Z. (2022). Using Deep Convolutional Neural Network for Image-Based Diagnosis of Nutrient Deficiencies in Plants Grown in Aquaponics. *Chemosensors*, *10(2)*, 1–23. <https://doi.org/10.3390/chemosensors10020045>
- Toyib, M., Pratama, T. D. K., & Aqil, I. (2024). Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data. *Algoritma: Jurnal Matematika, Ilmu pengetahuan Alam, Kebumihan dan Angkasa*, *2(3)*, 108–120.
- Utomo, A. Agung, & Mulyono, S. (2025). Perbandingan Mobilenetv2 Dan Densenet121 Pada Klasifikasi Terumbu Karang Menggunakan Convolutional Neural Network. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, *2(3 SE-Artikel)*, 930–941. <https://doi.org/10.70248/jrsit.v2i3.1844>

UNISSULA
جامعة سلطان أبجوج الإسلامية