ANALISIS PERFORMA *VISION TRANSFORMER* PADA KLASIFIKASI *FINE-GRAINED* CACAT CANGKANG TELUR

TUGAS AKHIR

Diajukan oleh:

Farras Nabil Rivanza

220705019

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH

2026 M / 1447 H

LEMBAR PERSETUJUAN

ANALISIS PERFORMA *VISION TRANSFORMER* PADA KLASIFIKASI *FINE-GRAINED* CACAT CANGKANG TELUR

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1) dalam Ilmu/Program Studi Teknologi Informasi

Oleh:

FARRAS NABIL RIVANZA

220705019

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Malahayati, M.T	Dr. Hendri Ahmadian, S.Si., M.I.M.
NIP. 198301272015032003	NIP. 198301042014031002

Mengetahui,

Ketua Program Studi Teknologi Informasi

Malahayati, M.T

NIP. 198301272015032003

LEMBAR PENGESAHAN

ANALISIS PERFORMA *VISION TRANSFORMER* PADA KLASIFIKASI *FINE-GRAINED* CACAT CANGKANG TELUR

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains Dan Teknologi UIN Ar-Raniry Banda Aceh Dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Kamis, 05 Februari 2026 M
17 Sya'ban 1447 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,

Malahayati, M.T
NIP. 198301272015032003

Sekretaris,

Dr. Hendri Ahmadian, S.Si., M.I.M.
NIP. 198301042014031002

Penguji I,

Raihan Islamadina, M.T
NIP. 198901312020122011

Penguji II,

Nizam Akbar, S.T., M.T
NUPTK. 2849779680130032

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	: Farras Nabil Rivanza
NIM	: 220705019
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Judul	: Analisis Performa *Vision Transformer* Pada Klasifikasi *Fine-Grained* Cacat Cangkang Telur

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 05 Februari 2026

Yang menyatakan

Farras Nabil Rivanza

ABSTRAK

Nama : Farras Nabil Rivanza

NIM : 220705019

Program Studi : Teknologi Informasi

Judul : Analisis Performa Vision Transformer Pada Klasifikasi Fine Grained Cacat Cangkang Telur

Tanggal : 05 Februari 2026

Jumlah Halaman : 72

Pembimbing 1 : Malahayati, M.T

Pembimbing 2 : Dr. Hendri Ahmadian, S.Si., M.I.M

Permasalahan kualitas dan keamanan pangan pada telur ayam sangat dipengaruhi oleh integritas cangkangnya. Proses *Quality Control* (QC) atau penyortiran manual memiliki kelemahan mendasar seperti tingkat subjektivitas yang tinggi, inkonsistensi akibat kelelahan mata, dan kapasitas sortir yang lambat. Masalah paling kritis dalam QC telur adalah deteksi cacat cangkang berupa retak halus (*hairline cracks*) yang berisiko tinggi memicu kontaminasi bakteri patogen ke dalam isi telur. Penelitian ini bertujuan untuk merancang serta menganalisis performa arsitektur *Vision Transformer* (ViT) dalam melakukan klasifikasi *fine-grained* cacat cangkang telur ke dalam tiga kelas, yaitu: Baik, Retak, dan Kotor. Metodologi yang digunakan mengadopsi kerangka kerja *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Pemodelan dibangun menggunakan teknik *Transfer Learning* dengan menerapkan arsitektur *pre-trained* ViT-B/16. Untuk meningkatkan ketangguhan (*robustness*) model, pelatihan dilakukan menggunakan *hybrid dataset* yang menggabungkan data sekunder dari repositori publik dan data primer hasil pengambilan langsung menggunakan kamera ponsel cerdas. Hasil evaluasi kuantitatif menunjukkan bahwa model ViT berhasil mencapai tingkat akurasi global sebesar 99%. Secara spesifik, model mencatatkan nilai *Recall* sempurna yaitu 1.00 (100%) untuk kelas 'Retak'. Tingginya nilai *Recall* ini sangat krusial dalam meminimalkan risiko lolosnya telur cacat (*False Negative*) ke tangan konsumen. Lebih lanjut, evaluasi menggunakan metode *Cross-Domain* mengonfirmasi bahwa model memiliki karakteristik *Fail-Safe* (Gagal di Sisi

Aman), di mana model terbukti mampu mendeteksi 100% telur retak pada domain data baru tanpa meloloskannya sebagai telur baik. Kesimpulannya, arsitektur *Vision Transformer* terbukti sangat efektif dan andal untuk mendeteksi anomali *fine-grained* pada cangkang telur, menjadikannya solusi yang sangat layak diimplementasikan dalam sistem QC otomatis untuk menjamin keamanan pangan.

Kata kunci: *Vision Transformer*, *Quality Control*, cacat cangkang telur, klasifikasi *fine-grained*, keamanan pangan, *deep learning*

ABSTRACT

Name : Farras Nabil Rivanza

Student ID : 220705019

Study Program : Information Technology

Title : Performance Analysis of Vision Transformer on Fine-Grained Classification of Eggshell Defects

Date : 05 February 2026

Number Of Page : 72

Supervisor 1 : Malahayati, M.T

Supervisor 2 : Dr. Hendri Ahmadian, S.Si., M.I.M

The quality and food safety of chicken eggs are highly dependent on the integrity of their shells. The manual Quality Control (QC) or sorting process has fundamental weaknesses such as a high degree of subjectivity, inconsistency due to human fatigue, and a slow sorting capacity. The most critical problem in egg QC is the detection of shell defects in the form of hairline cracks, which carry a high risk of triggering the contamination of pathogenic bacteria into the egg contents. This study aims to design and analyze the performance of the Vision Transformer (ViT) architecture in conducting fine-grained classification of eggshell defects into three classes: Good, Cracked, and Dirty. The methodology used adopts the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. The modeling was built using the Transfer Learning technique by applying the pre-trained ViT-B/16 architecture. To improve the robustness of the model, training was conducted using a hybrid dataset that combines secondary data from public repositories and primary data taken directly using a smartphone camera. The quantitative evaluation results show that the ViT model successfully achieved a global accuracy rate of 99%. Specifically, the model recorded a perfect Recall value of 1.00 (100%) for the 'Cracked' class. This high Recall value is very crucial in minimizing the risk of defective eggs (False Negative) reaching consumers. Furthermore, the evaluation using the Cross-Domain method confirmed that the model has a Fail-Safe characteristic, where the model was proven capable of detecting 100% of cracked eggs in a new data domain without letting them pass as good eggs. In conclusion,

the Vision Transformer architecture is proven to be highly effective and reliable for detecting fine-grained anomalies on eggshells, making it a very feasible solution to be implemented in an automated QC system to ensure food safety.

Keywords: *Vision Transformer*, *Quality Control*, eggshell defects, *fine-grained* classification, food safety, *deep learning*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Puji syukur kehadirat Allah SWT, yang atas limpahan rahmat, hidayah, dan karunia-Nya, Shalawat serta salam senantiasa tercurah kepada junjungan besar Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya yang setia hingga akhir zaman. Dengan limpahan rahmat dan kasih sayang Allah Yang Maha Pengasih lagi Maha Penyayang, penulis dapat menyelesaikan tugas akhir ini yang berjudul *"**Analisis Performa Vision Transformer Pada Klasifikasi Fine-Grained Cacat Cangkang Telur**"*. Karya ilmiah ini merupakan syarat guna menyelesaikan pendidikan Strata Satu (S-1) pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh.

Penulis menyampaikan rasa terima kasih yang tulus kepada semua pihak yang telah berkontribusi, baik secara langsung maupun tidak langsung, dalam proses pembelajaran, pemberian ilmu, dukungan moral, serta berbagai manfaat lainnya yang menjadi bagian penting hingga terselesaikannya skripsi ini. Oleh karena itu, pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Kedua orang tua tercinta, Ayahanda Saiful Bahri dan Ibunda Siti Dara Mardiana, serta seluruh keluarga besar yang telah memberikan doa yang tiada henti, dukungan moril, materil, dan kasih sayang yang tak terhingga selama penulis menempuh pendidikan. Hanya Allah yang dapat membalas kasih sayang mereka semua, semoga Allah selalu melimpahkan rahmatnya kepada mereka.
2. Bapak Prof. Dr. Ir. M. Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.
3. Ibu Malahayati, M.T. dan Bapak Khairan AR, M.Kom. Selaku Ketua dan Sekretaris Program Studi Teknologi Informasi yang telah memberikan arahan dalam menyelesaikan tugas akhir ini.
4. Bapak Mulkan Fadhli, S.T., M.T. selaku Dosen Pembimbing Akademik (PA) yang telah memberikan bimbingan dan arahan selama masa perkuliahan.

5. Ibu Malahayati M.T selaku dosen pembimbing 1 yang telah memberikan bimbingan, masukan, dan arahan selama penyusunan tugas akhir ini.
6. Bapak Dr. Hendri Ahmadian, S.Si., M.I.M selaku dosen pembimbing 2 yang telah memberikan bimbingan, masukan, dan arahan selama penyusunan tugas akhir ini.
7. Seluruh Dosen dan Staf Akademik Program Studi Teknologi Informasi yang telah memberikan ilmu, wawasan, dan pelayanan administrasi selama ini.
8. Seseorang dengan pemilik NIM 230705061 dan tanggal lahir 20 Agustus 2005, terima kasih telah pernah hadir dan menjadi salah satu halaman terindah dalam perjalanan ini. Terima kasih atas segala doa, dukungan, dan semangat yang diberikan secara tulus untuk Penulis. Kehadiranmu pernah menjadi alasan kuat bagi Penulis untuk terus melangkah hingga ke titik ini. Semoga semesta selalu menjagamu, terima kasih telah mengizinkan Penulis untuk mengenalmu.
9. HIMA - TI, organisasi yang telah menjadi wadah untuk Penulis berkembang dalam membentuk karakter dan pola pikir, yang secara tidak langsung turut membantu dalam penyelesaian studi ini. Terima kasih atas kesempatan, pengalaman organisasi, kebersamaan dan kekeluargaan, serta pelajaran berharga yang telah diberikan.
10. Sahabat dan rekan-rekan seperjuangan di Program Studi Teknologi Informasi Angkatan 2022, atas segala kebersamaan, semangat, dan dukungannya.
11. Yala Boys, Revolusi Warkop, dan anggota 3M terima kasih telah membersamai penulis dalam proses pembuatan skripsi ini, semoga kalian sukses selalu.
12. Dan yang terakhir semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah turut membantu dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna karena keterbatasan ilmu dan pengalaman. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif dari semua pihak demi perbaikan di masa mendatang.

Akhir kata, semoga tugas akhir ini dapat diterima, disetujui, dan penelitian yang akan dilaksanakan memberikan manfaat yang baik bagi penulis, almamater, maupun bagi pengembangan ilmu pengetahuan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Banda Aceh, 05 Februari 2026

Penulis,

Farras Nabil Rivanza

NIM. 220705019

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Telur ayam merupakan salah satu komoditas pangan dan sumber protein hewani utama yang dikonsumsi secara masif di Indonesia. Sebagai komoditas vital, jaminan kualitas dan keamanan pangan telur menjadi prioritas utama, baik bagi konsumen maupun produsen. Seiring dengan meningkatnya volume produksi di industri peternakan modern, proses *Quality Control* (QC) atau kontrol kualitas menjadi tantangan sentral. Proses QC ini tidak hanya menentukan nilai jual (*grading*) telur, tetapi juga, yang paling krusial, menjamin keamanan pangan (*food safety*) bagi konsumen (Abdullah et al., 2022).

Secara tradisional, proses sortir dan QC di peternakan masih banyak mengandalkan inspeksi visual manual oleh tenaga kerja. Metode ini memiliki berbagai kelemahan fundamental, di antaranya seperti subjektivitas yang tinggi, inkonsistensi akibat kelelahan mata (*human fatigue*), dan kapasitas sortir yang lambat yang tidak mampu mengimbangi skala produksi industri (Ghita & Miron, 2023).

Masalah paling kritis dalam QC telur adalah deteksi cacat cangkang, terutama retak halus (*hairline cracks*). Cacat yang sangat kecil ini sering terlewat oleh mata manusia namun sangat berbahaya. Secara ilmiah, integritas cangkang telur adalah penghalang fisik utama terhadap kontaminasi mikroba, dan telur retak terbukti memiliki risiko kontaminasi *Salmonella Enteritidis* yang signifikan (Freire et al., 2020). Retakan ini menjadi pintu masuk utama bagi bakteri patogen untuk mengkontaminasi isi telur dan menjadi risiko kesehatan serius bagi konsumen.

Otomatisasi proses QC menggunakan *Computer Vision* (CV) telah menjadi solusi yang banyak diteliti. Arsitektur standar seperti *Convolutional Neural Network* (CNN) telah umum digunakan untuk mendeteksi cacat telur (Wang et al., 2022) atau bahkan kotoran pada permukaan cangkang (Fan et al., 2022). Pendekatan lain seperti *object detection* (misalnya YOLOv5) juga telah diterapkan untuk deteksi retak secara *real-time* (Chen et al., 2023). Meskipun fungsional, arsitektur berbasis CNN yang bekerja dengan filter konvolusi lokal terkadang

memiliki keterbatasan dalam mendeteksi anomali yang sangat halus atau *fine-grained*, seperti membedakan retak halus dari pola atau tekstur alami cangkang (He et al., 2022).

Vision Transformer (ViT) sebuah arsitektur *Deep Learning* (DL) yang diperkenalkan oleh Dosovitskiy et al (2020) menawarkan pendekatan revolusioner. Dengan mengadopsi mekanisme *Self-Attention* dari pemrosesan bahasa alami (*Natural Language Processing* atau NLP), ViT menganalisis gambar secara holistik atau global. Hingga tahun 2025 ViT terus membuktikan keunggulannya dalam berbagai aplikasi presisi di bidang agrikultur (Murugavalli & Gopi, 2025). Untuk masalah deteksi cacat cangkang, ViT sangat menjanjikan karena kemampuannya dalam klasifikasi *fine-grained*, di mana mekanisme *attention* memungkinkan model untuk fokus secara presisi pada area anomali terkecil (He et al., 2022).

Keberhasilan ViT di domain *food quality control* lainnya telah divalidasi. Sebagai contoh, Ma et al. (2024) berhasil menerapkan ViT untuk deteksi cacat pada buah jeruk. Demikian pula, penelitian lain telah menunjukkan keunggulan ViT dalam mendeteksi cacat pada daging ayam broiler (Zheng et al., 2023) dan mengklasifikasikan kualitas biji kopi (Contreras et al., 2023). Tren ini menunjukkan potensi besar ViT untuk domain agrikultur dan QC pangan.

ViT telah terbukti sukses di domain QC pangan lain dan studi kasus QC telur didominasi oleh CNN, Oleh karena itu, pada penelitian ini akan dilakukan analisis performa arsitektur Vision Transformer (ViT) pada tugas klasifikasi *fine-grained* cacat cangkang telur dengan kategori: baik, retak, kotor sebagai fondasi untuk pengembangan *Quality Control System* yang otomatis dan akurat (Wang et al., 2022; Ma et al., 2024).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang arsitektur *deep learning* berbasis Vision Transformer (ViT) untuk melakukan klasifikasi *fine-grained* pada tiga kelas cacat cangkang telur (Baik, Retak, dan Kotor)?
2. Bagaimana performa model ViT dalam mendeteksi cacat *fine-grained*, terutama pada kelas 'Retak' (*hairline cracks*)?

1.3 Tujuan Penilitian

Sejalan dengan rumusan masalah yang telah ditetapkan, tujuan dari penelitian ini adalah:

1. Merancang dan membangun model klasifikasi berbasis arsitektur Vision Transformer (ViT) yang mampu mengidentifikasi citra telur ayam dalam kelas Baik, Retak, dan Kotor.
2. Menganalisis dan mengevaluasi performa model ViT secara kuantitatif menggunakan metrik standar, dengan fokus analisis utama pada metrik *Recall* untuk kelas 'Retak'.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat, baik secara teoretis maupun praktis:

1.4.1 Manfaat Teoretis (Akademis):

1. Memberikan kontribusi ilmiah pada bidang *deep learning* terapan, khususnya studi analisis performa arsitektur ViT untuk deteksi anomali dan *food quality control*.
2. Mengisi *research gap* mengenai penerapan ViT pada studi kasus klasifikasi *fine-grained* cacat cangkang telur.

1.4.2 Manfaat Praktis (Industri):

1. Menghasilkan sebuah *blueprint* model tervalidasi dan laporan analisis performa yang dapat diadopsi oleh industri peternakan lokal sebagai dasar pengembangan sistem *Quality Control* (QC) otomatis.
2. Membantu meningkatkan efisiensi proses sortir dan menjamin standar keamanan pangan bagi konsumen dengan mendeteksi cacat retak secara lebih akurat.

1.5 Batasan Masalah

Untuk memastikan penelitian tetap fokus, mendalam, dan dapat diselesaikan tepat waktu, maka penelitian ini dibatasi oleh ruang lingkup berikut:

1. Penelitian hanya berfokus pada kualitas eksternal cangkang telur ayam yang dibagi menjadi 3 (tiga) kelas klasifikasi: Baik (utuh, bersih), Retak (termasuk retak besar dan retak halus/rambut), dan Kotor (terkena noda).

2. Penelitian tidak menganalisis kualitas internal (kesegaran, Haugh Unit, warna kuning telur) ataupun *grading* ukuran (Besar, Sedang, Kecil).
3. Data citra yang digunakan adalah dataset gabungan (*hybrid*) yang terdiri dari data sekunder (publik dari *repository* seperti Roboflow, Kaggle dan Mendeley Data) dan data primer (koleksi peneliti).
4. Output (Keluaran) utama dari penelitian ini adalah model *deep learning* yang telah tervalidasi performanya dan laporan analisis performa model tersebut.
5. Penelitian ini tidak melakukan perbandingan performa dengan arsitektur lain (seperti CNN) dan tidak mencakup implementasi sistem aplikasi *real-time*, aplikasi web/seluler, atau integrasi perangkat keras (konveyor) jadi.

BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Tinjauan pustaka pada penelitian terdahulu merupakan langkah krusial untuk memetakan lanskap penelitian dan mengidentifikasi celah penelitian (*research gap*) yang spesifik. Pada bagian ini, dianalisis berbagai studi relevan yang diterbitkan dalam 5 (lima) tahun terakhir (2020-2025) yang berfokus pada dua pilar utama: (1) penerapan metodologi *Deep Learning* (CNN/YOLO) pada studi kasus QC telur, dan (2) penerapan metodologi ViT pada domain *fine-grained* dan QC pangan lainnya. Rangkuman dari penelitian-penelitian kunci yang menjadi fondasi justifikasi penelitian ini disajikan secara rinci dalam Tabel 2.1.

Tabel 2. 1 Tabel Penelitian Terdahulu

No.	Peneliti	Judul	Kesimpulan
1.	Rachman, A., Fauzi, A., & Wijonarko, B. (2026).	Klasifikasi Citra Rempah-Rempah di Indonesia Menggunakan Vision Transformer	Penelitian ini berhasil membangun sistem klasifikasi otomatis untuk 31 kelas rempah-rempah Indonesia. Model terbaik dicapai pada konfigurasi pelatihan 3 epoch yakni 97,2% Penggunaan metode *transfer learning* terbukti sangat efektif dan efisien. Model tidak perlu dilatih dari nol karena memanfaatkan bobot dari model ViT yang sudah dilatih sebelumnya pada dataset besar (ImageNet).

2.	Murugavalli, S., & Gopi, R. (2025)	Plant Leaf Disease Detection Using Vision Transformers For Precision Agriculture.	Penelitian ini membuktikan bahwa ViT terus menjadi metode unggulan (*state-of-the-art*) untuk tugas klasifikasi visual kompleks di bidang agrikultur, menunjukkan relevansi jangka panjang arsitektur ini dalam mendeteksi fitur penyakit yang halus
3.	Ma, C. et al. (2024)	Interpretable Citrus Fruit Quality Assessment Using Vision Transformers and Lightweight Large Language Models.	Penelitian terbaru ini membuktikan bahwa ViT sangat efektif untuk *quality control* (QC) produk pangan. Model ViT mereka berhasil mendeteksi cacat dan kebusukan pada permukaan buah jeruk dengan akurasi tinggi.
4.	Wang, Y. et al. (2023)	Egg Crack Detection Based on Improved EfficientNet-B4.	Penelitian ini juga berfokus pada deteksi retak telur. Hasilnya menunjukkan bahwa arsitektur CNN ringan (EfficientNet-B4) yang telah dimodifikasi mampu mencapai performa yang sangat baik untuk deteksi retak.
5.	Wang, J. et al. (2022)	Automatic Detection of	Penelitian ini berhasil membangun sistem untuk

		Eggshell Defects Based on Deep Learning.	mendeteksi cacat telur (termasuk retak). Hasilnya menunjukkan akurasi yang tinggi (di atas 98%) dengan menggunakan arsitektur CNN (secara spesifik VGG dan ResNet).
6.	He, J. et al. (2022)	TransFG: A Transformer Architecture for Fine-grained Recognition.	Penelitian ini membuktikan bahwa arsitektur berbasis ViT sangat superior untuk tugas Klasifikasi *Fine-Grained* (membedakan sub-kategori yang sangat mirip), karena mekanisme *attention*-nya dapat fokus pada detail pembeda yang halus.
7.	Dosovitskiy, A. et al. (2020)	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.	Penelitian ini adalah yang pertama memperkenalkan arsitektur *Vision Transformer* (ViT). Hasilnya membuktikan bahwa arsitektur berbasis *Transformer* murni dapat mengungguli performa CNN *state-of-the-art* pada *benchmark* klasifikasi gambar skala besar.

Tren penelitian terkini hingga tahun 2025, sebagaimana ditunjukkan oleh Murugavalli & Gopi (2025) semakin mempertegas relevansi arsitektur ViT untuk menangani masalah klasifikasi visual yang kompleks di sektor agrikultur.

Dari tinjauan pustaka di atas, sebuah *research gap* (celah penelitian) yang jelas dapat diidentifikasi:

1. Di satu sisi, Metodologi ViT telah terbukti sebagai *state-of-the-art* dan sangat unggul untuk tugas *fine-grained Quality Control* pada berbagai produk pangan (Jeruk, Kopi, Daging Ayam) (Ma et al., 2024; He et al., 2022).
2. Di sisi lain, Studi Kasus *Quality Control* Telur, meskipun telah diteliti (Wang et al., 2022), masih didominasi oleh metodologi lama (CNN/YOLO).

Belum ada studi mendalam yang berfokus menganalisis performa arsitektur *state-of-the-art* Vision Transformer (ViT) untuk studi kasus spesifik deteksi cacat *fine-grained* pada cangkang telur.

Penelitian ini bertujuan untuk mengisi celah tersebut. Penelitian ini akan menjadi salah satu studi pertama yang secara khusus menerapkan dan menganalisis seberapa baik performa ViT (ViT-B/16) dalam mengatasi masalah industrial yang penting ini, menggunakan justifikasi ilmiah dari paper *fine-grained* (He et al., 2022) dan *food safety* (Freire et al., 2020).

2.2 Landasan Teori

Bagian ini menguraikan konsep-konsep fundamental yang menjadi dasar dari perancangan model dalam penelitian ini, disusun secara hierarkis dari umum ke khusus.

2.2.1 Computer Vision dan Quality Control

Computer Vision (CV) adalah sebuah bidang ilmu komputer yang berfokus pada bagaimana komputer dapat "melihat" dan menginterpretasi data visual (gambar dan video) seperti halnya manusia. Tujuannya adalah untuk mengotomatisasi tugas-tugas yang dapat dilakukan oleh sistem visual manusia.

Dalam konteks industri, CV telah menjadi teknologi transformatif, terutama untuk *Quality Control* (QC) atau Kontrol Kualitas. Inspeksi visual manual oleh manusia, meskipun penting, memiliki keterbatasan signifikan: bersifat subjektif (standar berbeda antar individu), lambat, tidak konsisten, dan rentan terhadap kelelahan (*human fatigue*). Sistem QC berbasis CV menggantikan inspeksi manual ini dengan sistem otomatis yang mampu mendeteksi cacat produk, mengukur

dimensi, dan memvalidasi standar kualitas secara objektif, konsisten, dan dalam kecepatan tinggi (Ghita & Miron, 2023). Penelitian ini berfokus pada penerapan CV untuk QC cacat cangkang telur.

2.2.2 Telur Ayam

Telur ayam merupakan salah satu komoditas pangan sumber protein hewani yang paling penting dan dikonsumsi secara luas secara global. Dalam produksi skala industri, kualitas telur sangat bervariasi dan perlu distandarisasi melalui proses *Quality Control* (QC). Kualitas telur dibagi menjadi dua kategori yaitu, kualitas internal (kesegaran, ukuran kuning telur) dan kualitas eksternal (integritas cangkang).

Penelitian ini berfokus pada kualitas eksternal, yang merupakan garda terdepan keamanan pangan. Integritas struktural cangkang adalah komponen kunci; adanya cacat seperti retak halus (*hairline cracks*) atau noda kotoran secara signifikan meningkatkan risiko kontaminasi bakteri patogen. Secara spesifik, cangkang yang retak telah terbukti secara ilmiah menjadi jalur masuk utama bagi bakteri *Salmonella* untuk mengkontaminasi isi telur, yang merupakan risiko kesehatan publik yang serius (Freire et al., 2020). Secara tradisional, inspeksi cacat cangkang ini dilakukan secara manual (dikenal sebagai *candling*). Namun, metode ini tidak efisien untuk skala industri. Oleh karena itu, pengembangan metode non-destruktif berbasis *Computer Vision* untuk inspeksi kualitas telur (termasuk deteksi retak) telah menjadi bidang penelitian yang sangat aktif (Abdullah et al., 2022).

2.2.3 Klasifikasi Citra

Klasifikasi citra (*image classification*) adalah tugas fundamental dalam *Computer Vision*. Ini adalah proses di mana sebuah model komputasi menganalisis sebuah gambar dan memberikan *output* berupa satu label (kelas) dari sekumpulan kelas yang telah ditentukan sebelumnya (Kuznetsova et al., 2021).

Dalam konteks penelitian ini, tugasnya adalah klasifikasi *multi-kelas*. Model akan menerima satu gambar input (citra telur) dan harus memutuskan label mana yang paling tepat dari tiga pilihan kelas yang telah ditentukan: Baik, Retak, atau Kotor. Untuk mencapai ini, model harus dilatih menggunakan *dataset* berlabel, di mana ia belajar menemukan pola visual (fitur) yang membedakan satu kelas dari

kelas lainnya. Dalam kasus ini, model harus belajar bahwa pola visual "garis halus tidak beraturan" berkorelasi kuat dengan label Retak, sementara pola "permukaan bersih berbentuk oval" berkorelasi dengan label Baik. Penelitian ini secara spesifik menangani klasifikasi *fine-grained*, di mana perbedaan visual antar kelas (misal: Baik vs. Retak Halus) sangat minim dan sulit dibedakan.

2.2.4 Machine Learning

Machine Learning (ML) atau Pembelajaran Mesin adalah sub-bidang dari *Artificial Intelligence* (AI) yang berfokus pada pengembangan algoritma yang memungkinkan komputer untuk menemukan pola dan meningkatkan kinerjanya dari data tanpa diprogram secara eksplisit (Alpaydin, 2020). Berbeda dengan pemrograman tradisional yang memerlukan instruksi *rule-based* yang kaku, ML menggunakan model statistik untuk menemukan pola kompleks dalam data dan membuat prediksi atau keputusan berdasarkan pola tersebut.

Secara umum, ML tradisional (seperti *Support Vector Machines* atau *Random Forest*) seringkali memerlukan proses ekstraksi fitur manual (*manual feature engineering*), di mana seorang ahli harus menentukan fitur-fitur apa dari data (misalnya, warna rata-rata, tekstur, bentuk) yang penting untuk diberikan kepada model. Keberhasilan model sangat bergantung pada kualitas fitur yang dirancang secara manual ini.

Metode ML secara umum dibagi menjadi dua kategori utama berdasarkan cara model belajar dari data (Sarker, 2021):

1. *Supervised Learning* (Pembelajaran Terarah)

   Ini adalah metode yang akan digunakan dalam penelitian ini. Dalam *Supervised Learning*, model dilatih menggunakan dataset yang sudah berlabel. Data input (disebut X) disajikan bersama dengan output yang benar (disebut Y atau label). Tujuan model adalah mempelajari fungsi pemetaan F(X) = Y. Untuk tugas klasifikasi (seperti penelitian ini), output Y adalah label kategorikal (misal: Baik, Retak, Kotor). Keberhasilan model diukur secara langsung berdasarkan kemampuannya memprediksi label yang benar untuk data baru yang belum pernah dilihat (menggunakan Akurasi, Presisi, dan Recall).

2. *Unsupervised Learning* (Pembelajaran Tak Terarah)

Dalam *Unsupervised Learning*, model dilatih menggunakan dataset yang tidak memiliki label. Model hanya diberikan data input (X) dan tujuannya adalah untuk menemukan struktur atau pola tersembunyi dalam data itu sendiri. Tugas yang paling umum adalah *clustering* (pengelompokan), di mana algoritma (seperti *K-Means*) mencoba mengelompokkan data berdasarkan kemiripannya.

Penelitian ini secara sadar memilih *Supervised Learning* karena tujuannya adalah deteksi spesifik (mencari Retak), bukan eksplorasi data (*clustering*). Metode *supervised* memungkinkan evaluasi yang ketat terhadap metrik kinerja yang krusial untuk QC (seperti *Recall*), yang tidak mungkin dilakukan dengan metode *unsupervised*.

2.2.5 Deep Learning

Deep Learning (DL) adalah sub-bidang yang lebih spesifik dari *Machine Learning* yang menggunakan arsitektur *Artificial Neural Network* (ANN) dengan banyak lapisan (disebut "*deep*" atau "dalam") (Sarker, 2021). Keunggulan utama DL dibandingkan ML tradisional adalah kemampuannya melakukan ekstraksi fitur otomatis (*automatic feature extraction*) atau *representation learning*.

Pada data gambar, lapisan pertama DL (misalnya pada CNN) mungkin belajar mendeteksi fitur sederhana (tepi, sudut). Lapisan-lapisan berikutnya secara hierarkis akan menggabungkan fitur-fitur tersebut untuk mengenali fitur yang lebih kompleks (tekstur, pola, bentuk), hingga akhirnya mampu mengenali objek utuh (telur). Kemampuan ekstraksi fitur otomatis inilah yang membuat DL sangat dominan dalam tugas-tugas persepsi kompleks seperti *Computer Vision* (Sarker, 2021).

2.2.6 Transfer Learning

Dalam pengembangan model *Deep Learning* pada dataset dengan jumlah terbatas, melatih jaringan saraf tiruan dari awal (*training from scratch*) seringkali menghasilkan performa yang buruk akibat *overfitting*. Solusi standar untuk mengatasi masalah ini adalah *Transfer Learning*.

Menurut Zhuang et al. (2021), *Transfer Learning* adalah metodologi pembelajaran mesin di mana pengetahuan yang diperoleh dari satu domain masalah

(*source domain*) ditransfer untuk meningkatkan fungsi pembelajaran pada domain target yang berbeda namun terkait (target domain). Tujuannya adalah memanfaatkan fitur-fitur visual umum (seperti tepi, tekstur, dan bentuk) yang telah dipelajari model pada dataset raksasa (seperti *ImageNet*) agar tidak perlu dipelajari ulang dari nol.

Khusus untuk arsitektur *Vision Transformer* (ViT), penerapan *Transfer Learning* bersifat krusial. Khan et al. (2022) dalam survei komprehensifnya menjelaskan bahwa ViT memiliki bias induktif (*inductive bias*) yang lebih rendah dibandingkan CNN, yang artinya ViT membutuhkan jumlah data yang jauh lebih masif untuk belajar generalisasi. Oleh karena itu, pada dataset skala kecil hingga menengah, ViT yang menggunakan bobot *pre-trained* terbukti secara konsisten mengungguli ViT yang dilatih dari awal.

Strategi *Transfer Learning* yang diterapkan dalam penelitian ini adalah *Fine-Tuning.* Berbeda dengan sekadar ekstraksi fitur, *Fine-Tuning* melibatkan pelatihan ulang seluruh atau sebagian parameter model *pre-trained* dengan laju pembelajaran (*learning rate*) yang sangat rendah. Hal ini memungkinkan model untuk menyesuaikan representasi fitur global yang telah dipelajari sebelumnya agar lebih spesifik terhadap karakteristik dataset cangkang telur yang digunakan (Wang et al., 2022).

2.2.7 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah arsitektur *Deep Learning* yang menjadi standar emas (*gold standard*) untuk tugas *Computer Vision* selama dekade terakhir (Wang et al., 2022; Fan et al., 2022). Keunggulan arsitektur ini terletak pada dua konsep utama: *local receptive fields* (filter konvolusi) dan *parameter sharing*.

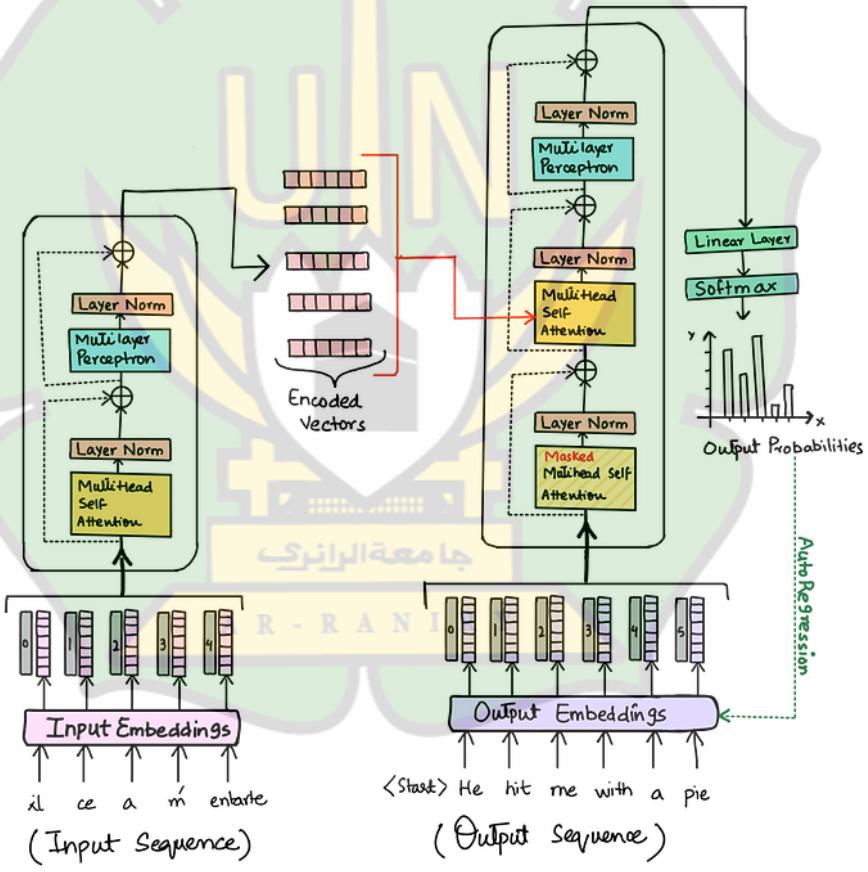
Filter konvolusi bekerja sebagai detektor fitur yang memindai gambar dan mengenali pola lokal (tepi, tekstur) dalam area kecil gambar. Karena *filter* yang sama digunakan di seluruh bagian gambar (*parameter sharing*), CNN sangat efisien secara komputasi. Arsitektur populer seperti ResNet, VGG, dan MobileNet adalah contoh CNN.

Namun, sifat alaminya yang fokus pada fitur lokal membuat CNN memiliki keterbatasan teoretis yaitu CNN secara inheren kesulitan dalam

memahami konteks global atau relasi spasial jangka panjang antar bagian gambar. Untuk tugas *fine-grained* (seperti mendeteksi retak halus), model mungkin kesulitan membedakan antara tekstur retakan dengan tekstur alami cangkang tanpa memahami konteks gambar secara utuh.

2.2.8 Arsitektur Transformer dan Self-Attention

Secara visual, arsitektur *Transformer* orisinal awalnya dirancang untuk tugas *Natural Language Processing* (NLP) seperti translasi dapat dilihat pada Gambar 2.1. Arsitektur ini terdiri dari dua komponen utama: tumpukan *Encoder* (di sebelah kiri), yang bertugas mencerna *input* dan membangun representasi konteks, dan tumpukan *Decoder* (di sebelah kanan), yang bertugas menghasilkan *output* sekuensial (Khan et al., 2022).



Gambar 2. 1 Arsitektur Transformer

Penting untuk dicatat bahwa arsitektur *Vision Transformer* (ViT) yang digunakan dalam penelitian ini hanya mengadopsi dan menggunakan tumpukan *Encoder* (sisi kiri) dari arsitektur asli ini untuk tugas klasifikasi citra.

Arsitektur *Transformer* awalnya dirancang untuk NLP guna mengatasi kelemahan arsitektur sekuensial seperti *Recurrent neural network* (RNN) atau *Long Short-Term Memory* (LSTM) yang kesulitan memproses dependensi jarak jauh (kata di awal dan akhir kalimat). Inti dari arsitektur *Transformer* adalah mekanisme *Self-Attention* (Khan et al., 2022).

Self-Attention adalah mekanisme yang memungkinkan model, ketika memproses satu *input* (misalnya, satu *patch* gambar), untuk melihat dan menimbang pentingnya semua *input* lain dalam sekuens tersebut. Ini memberikan kemampuan pemahaman konteks global.

Secara teknis, mekanisme *Self-Attention* bekerja dengan memproyeksikan setiap *input* (token atau *patch*) menjadi tiga vektor terpisah yang dapat dipelajari atau dilatih (Khan et al., 2022). Ketiga vektor ini adalah:

1. *Query* (Q): Sebuah representasi dari *input* saat ini yang digunakan untuk mengukur relevansi atau menanyakan informasi dari *input* lain di dalam sekuens.
2. *Key* (K): Sebuah representasi dari *input* yang bertindak sebagai kunci pencocokan atau identifikator. Vektor *Query* akan dicocokkan dengan semua vektor *Key* untuk menemukan kesamaan.
3. *Value* (V): Vektor ini membawa informasi semantik atau konten aktual dari *input*. Ini adalah informasi yang akan diambil dan diagregasi setelah relevansinya dihitung.

Proses *Self-Attention* kemudian berjalan dalam empat langkah matematis (Khan et al., 2022) yaitu:

1. Kalkulasi Skor (*Score*) yaitu, model menghitung skor relevansi dengan melakukan operasi *dot product* (perkalian matriks) antara vektor *Query* (Q) dari *input* saat ini dengan vektor *Key* (K) dari *semua input lain* (termasuk dirinya sendiri). Skor ini menentukan tingkat kecocokan atau relevansi antar *input*.

2. Pembagian (*Scaling*) yaitu, skor tersebut dibagi dengan akar kuadrat dari dimensi vektor *Key* untuk menjaga stabilitas gradien saat *training*.
3. Normalisasi (*Softmax*) yaitu, skor-skor yang telah dihitung kemudian dinormalisasi menggunakan fungsi *Softmax*. Tujuannya adalah untuk mengubah skor mentah menjadi distribusi probabilitas (nilai antara 0 dan 1) yang jika dijumlahkan hasilnya 1. Hasil ini disebut *Attention Weights* (Bobot Perhatian), yang secara efektif menentukan seberapa besar "porsi perhatian" yang harus diberikan pada setiap *input* lain.
4. Agregasi (*Weighted Sum*) yaitu, hasil dari *Softmax* yang disebut *Attention Weights* tersebut kemudian dikalikan dengan vektor *Value* (V) dari masing-masing *input*. *Input* yang dianggap sangat relevan (skor *softmax* tinggi) akan memiliki kontribusi *Value* yang besar. Vektor-vektor *Value* yang telah diboboti ini kemudian dijumlahkan untuk menghasilkan satu vektor *output*.

Vektor *output* ini adalah representasi baru dari *input* asli, yang kini telah diperkaya dengan konteks dari semua *input* lain yang relevan di seluruh sekuens. Kemampuan untuk memproses seluruh *input* secara paralel dan menangkap konteks global inilah yang membuat *Transformer* sangat superior (Khan et al., 2022).

Secara matematis, keempat langkah ini dapat diringkas ke dalam satu formula tunggal untuk *Scaled Dot-Product Attention*, yang diekspresikan sebagai berikut:

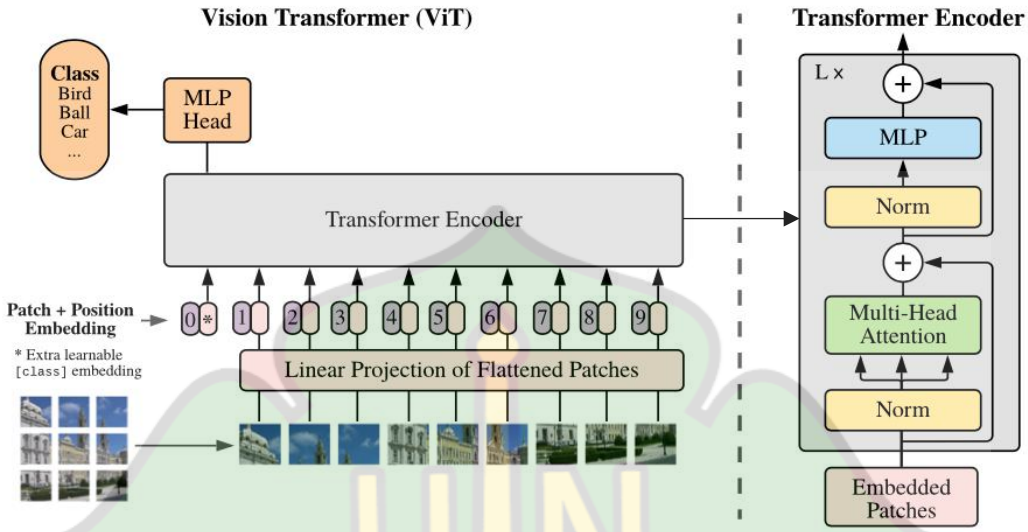
$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Di mana Q adalah matriks *Queries*, K adalah matriks *Keys*, V adalah matriks *Values*, dan d_k adalah dimensi dari vektor *Key* (yang digunakan untuk *scaling*). Formula inilah yang menjadi inti komputasi dalam setiap modul *Self-Attention*.

2.2.9 Vision Transformer

Vision Transformer (ViT), yang diperkenalkan oleh Dosovitskiy et al. (2020), adalah adaptasi arsitektur *Transformer* murni untuk tugas *Computer Vision*. Arsitektur ini secara fundamental menantang dominasi CNN dengan mengganti konvolusi lokal dengan mekanisme atensi global. Sebagaimana diilustrasikan pada gambar dibawah yaitu arsitektur ViT memproses citra dengan membaginya menjadi

patches (potongan) berurutan, yang kemudian diproyeksikan dan diproses oleh *Transformer Encoder* untuk menangkap hubungan global antar-bagian gambar sebelum dilakukan klasifikasi akhir (Dosovitskiy et al., 2020). Seperti yang ditunjukan pada gambar 2.3.



Gambar 2. 2 Alur Kerja ViT

Proses kerja ViT adalah sebagai berikut:

1. *Image Patching* (Pemotongan Gambar): Gambar input (misal: 224x224 piksel) dipecah menjadi serangkaian kotak-kotak kecil (*patches*) yang tidak tumpang tindih. Sesuai fokus penelitian ini, akan digunakan *patch size* 16x16 (model ViT-B/16). Ini mengubah gambar 224x224 menjadi 196 *patches* (sebuah grid 14x14).
2. *Patch Embedding* (Proyeksi Linear): Setiap *patch* (berdimensi 768) diratakan (*flattened*) menjadi vektor satu dimensi (768 dimensi). Vektor ini kemudian diproyeksikan secara linear ke dalam dimensi laten model (disebut *D-dimension*, misal 768). Proses ini disebut *Patch Embedding*.
3. Penambahan *Classification Token* atau Token Klasifikasi (CLS) dan *Positional Embedding*:
    a. *Classification Token* (*CLS Token*) yaitu sebuah token khusus yang dapat dipelajari (mirip *token* pada model BERT di NLP) ditambahkan di awal sekuens *patch embeddings*. Tujuan dari *token* ini adalah untuk bertindak sebagai agregator global. Token ini akan

mengumpulkan informasi representasi dari seluruh *patch* selama proses *training* dan *output*-nya akan digunakan untuk klasifikasi akhir.

b. *Positional Embedding* yaitu arsitektur *Transformer* pada dasarnya tidak memiliki pemahaman urutan (*permutation-invariant*). Ia tidak tahu apakah *patch* A berasal dari sudut kiri atas atau kanan bawah. Untuk mengatasi ini, *Positional Embedding* (vektor posisi yang dapat dipelajari) ditambahkan ke setiap *patch embedding* untuk menyuntikkan informasi spasial atau posisi asli dari setiap *patch*.

4. *Transformer Encoder:* Sekuens *patch embeddings* yang kini telah dilengkapi dengan informasi posisi (termasuk Token Klasifikasi) dimasukkan ke *Transformer Encoder* standar. *Encoder* ini terdiri dari beberapa *layer* yang ditumpuk, di mana setiap *layer* berisi modul *Multi-Head Self-Attention*, *Layer Normalization*, dan *MLP block*. Di sinilah setiap *patch* "melihat" dan mengevaluasi hubungannya dengan semua *patch* lain untuk membangun pemahaman konteks global.
5. *Multi-Layer Perceptron Head* (*Classifier*): Setelah sekuens diproses oleh *Encoder*, *output* yang hanya berasal dari posisi Token Klasifikasi yang diambil. Vektor *output* dari Token Klasifikasi ini (yang kini berisi representasi gambar teragregasi) kemudian dimasukkan ke MLP *head* untuk melakukan klasifikasi akhir.

Alur kerja di dalam satu *block* adalah *Output* dari lapisan sebelumnya pertama kali melalui *Layer Normalization (Norm)*, kemudian masuk ke *Multi Head Self Attention*. Hasilnya ditambahkan kembali ke *input* (residual connection). *Output* gabungan ini kemudian melalui *Layer Normalization* lagi, lalu masuk ke *Feed Forward Network* (MLP). Hasil akhirnya ditambahkan lagi ke *input* sebelumnya. Alur (Norm -> MSA -> Add -> Norm -> FFN -> Add) ini diulang sebanyak 12 kali.

Justifikasi Pemilihan ViT-B/16 yaitu didasari oleh keunggulan teoretis ViT yaitu kemampuannya untuk menangkap konteks global dan untuk tugas klasifikasi *fine-grained* (deteksi retak halus) ini sangat krusial (He et al., 2022). Hipotesisnya adalah, tidak seperti CNN yang mungkin mengabaikan retak halus sebagai *noise*

lokal, *Self-Attention* pada ViT dapat belajar bahwa *patch* yang berisi retakan memiliki relasi kontekstual yang kuat dengan *patch* lain di sekitarnya, sehingga memungkinkannya mengidentifikasi cacat *fine-grained* dengan lebih baik (He et al., 2022). Penggunaan *patch size* 16x16 (menghasilkan 196 *patches*) dipilih karena secara teoretis menawarkan resolusi spasial yang lebih tinggi untuk menangkap detail halus tersebut, dibandingkan dengan *patch size* yang lebih besar (misal: 32x32, yang hanya menghasilkan 49 *patches*).

2.2.10 Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) adalah arsitektur fundamental dari *Artificial Neural Network* (ANN) dan merupakan komponen inti dari banyak arsitektur *Deep Learning* (Sarker, 2021). MLP adalah model matematis yang terinspirasi dari neuron biologis, yang dirancang untuk mempelajari pola non-linear kompleks dalam data.

Secara struktural, MLP terdiri dari minimal tiga jenis lapisan:

1. *Input Layer* (Lapisan Input) yakni lapisan yang menerima data input mentah (vektor fitur). Dalam konteks ViT, ini bisa berupa *output* dari modul *Self-Attention* (di dalam *Encoder block*) atau *output* dari Token Klasifikasi (di *Classifier Head*).
2. *Hidden Layer* (Lapisan Tersembunyi) ini adalah lapisan komputasi utama. Setiap lapisan tersembunyi terdiri dari sejumlah *neuron*. Setiap *neuron* melakukan transformasi linear (perkalian bobot dan penambahan bias) pada *input* dari lapisan sebelumnya, yang kemudian diikuti oleh Fungsi Aktivasi non-linear (misalnya, ReLU, GELU, atau Sigmoid). Fungsi aktivasi inilah yang memungkinkan MLP untuk mempelajari pola yang kompleks.
3. *Output Layer* (Lapisan Output) yakni lapisan terakhir yang menghasilkan *output* akhir dari model.

Dalam arsitektur ViT, MLP memiliki dua peran spesifik:

1. MLP *Block* (di dalam Encoder): Berfungsi sebagai *feed-forward network* setelah modul *Self-Attention* untuk memproses lebih lanjut representasi data di setiap *layer Encoder*.

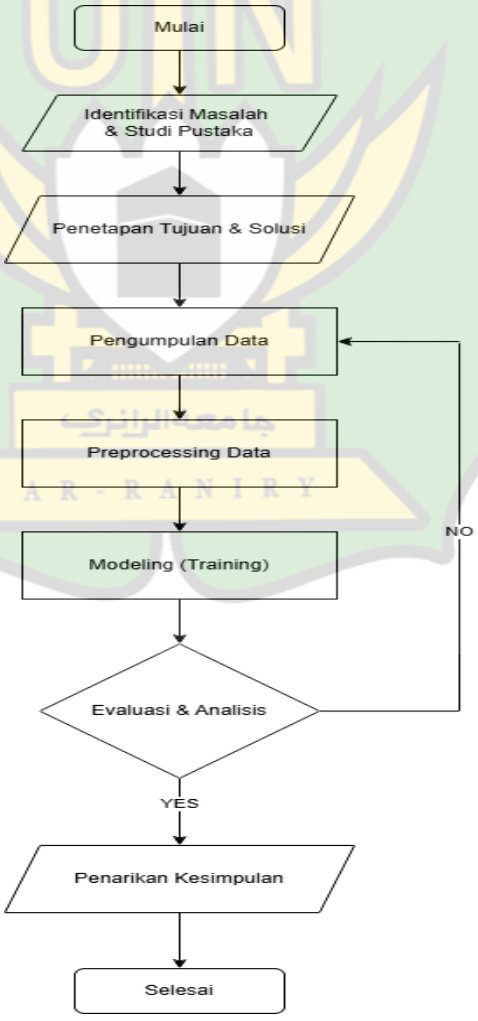
2. MLP *Head* (*Classifier*): Berfungsi sebagai kepala klasifikasi di akhir arsitektur. Komponen inilah yang mengambil representasi gambar final (vektor 768 dimensi dari Token Klasifikasi) dan memetakannya ke jumlah kelas yang diinginkan. Dalam penelitian ini, *MLP Head* akan mengubah vektor 768-dimensi menjadi 3 *output neuron* (Baik, Retak, Kotor) dan menggunakan fungsi aktivasi Softmax untuk menghasilkan probabilitas klasifikasi.

BAB III
METODE PENELITIAN

Bab ini menguraikan langkah-langkah sistematis yang akan dilakukan dalam penelitian. Metodologi ini mengambil pendekatan penelitian yaitu *Cross-Industry Standard Process for Data Mining* (CRISP-DM), tahapan rinci penelitian, serta alat dan bahan yang akan digunakan untuk mencapai tujuan penelitian.

3.1 Kerangka Kerja Penelitian

Penelitian ini mengadopsi kerangka kerja CRISP-DM, yakni sebuah metodologi standar industri yang paling banyak digunakan untuk proyek *data science* dan *machine learning* (Sarker, 2021). Adapun untuk kerangka kerja CRISP-DM yang digunakan sebagai acuan utama penelitan dijabarkan pada gambar 3.1.



Gambar 3. 1 Kerangka Kerja CRISP-DM

Alasan pemilihan CRISP-DM adalah karena sifatnya yang terstruktur, sistematis, dan iteratif (memungkinkan adanya proses kembali ke tahap sebelumnya jika diperlukan). penelitian ini akan mengikuti enam fase CRISP-DM yang disesuaikan dengan konteks penelitian skripsi:

1. *Business Understanding* (Pemahaman Masalah):
    a. *Proses:* Pada fase ini, dilakukan identifikasi masalah utama (QC telur manual tidak efisien, retak halus berbahaya bagi kesehatan pangan), penetapan tujuan proyek (menganalisis performa ViT untuk deteksi cacat *fine-grained*), dan penentuan kriteria sukses (tingkat *Recall* yang tinggi untuk kelas 'Retak').
    b. *Output:* Menghasilkan Bab 1 (Pendahuluan).
2. *Data Understanding* (Pemahaman Data):
    a. *Proses:* Melakukan studi pustaka untuk mencari *dataset* publik yang ada. Fase ini mengidentifikasi bahwa *dataset* publik yang ada (misal dari Kaggle/Mendeley) memiliki keterbatasan (misal: tidak ada kelas 'Kotor' yang jelas).
    b. *Output:* Menghasilkan keputusan untuk menggunakan strategi dataset gabungan (*hybrid*) guna meningkatkan ketangguhan (*robustness*) model.
3. *Data Preparation* (Persiapan Data):
    a. *Proses:* Ini adalah fase teknis utama yang mencakup diantaranya mulai dari akuisisi data sekunder (dari Kaggle/Mendeley) dan data primer (pengambilan foto baru), pembersihan dan penyeragaman label (baik, retak, kotor), transformasi data (*resizing* ke 224x224), dan pemisahan data (*splitting*) menjadi set latih, validasi, dan uji, serta augmentasi data pada set Latih.
    b. *Output:* Dataset final yang siap digunakan untuk *training* model.
4. *Modeling* (Pemodelan):
    a. *Proses:* Memilih arsitektur model (ViT-B/16) sesuai justifikasi teoretis. Menerapkan *Transfer Learning* menggunakan *library* Hugging Face

transformers untuk memuat bobot *pre-trained* dan menginisialisasi *head classifier* baru. Melakukan proses *training* dan *tuning*.

b. *Output:* Model ViT (model.safetensors atau .pth) yang telah dilatih.

5. *Evaluation* (Evaluasi):
   a. *Proses:* Menguji performa model yang telah dilatih menggunakan Data Uji (data yang belum pernah dilihat model). Menghitung metrik performa (Akurasi, Presisi, *Recall*, F1-Score) dan menganalisis *Confusion Matrix*.
   b. *Output:* Laporan analisis performa
6. *Deployment* (Penerapan):
   a. *Proses:* Sesuai Batasan Masalah, fase ini tidak diimplementasikan. Penelitian berhenti di Fase Evaluasi.
   b. *Output:* Rekomendasi untuk implementasi di masa depan

3.2 Pengumpulan Data

Untuk membangun model yang tangguh (*robust*) dan mampu melakukan generalisasi dengan baik terhadap variasi kondisi lapangan (pencahayaan, kamera, jenis telur), penelitian ini akan menggunakan pendekatan *hybrid dataset* (dataset gabungan).

3.2.1 Data Sekunder (*Baseline Dataset*)

Data sekunder digunakan sebagai basis data utama (*baseline*) untuk melatih model mengenali variasi fitur visual dasar pada cangkang telur. Sumber dataset publik diperoleh dari *repository* ilmiah terpercaya seperti Kaggle dan Mendeley Data.

Pada tahap realisasi, sebanyak 2.319 citra berhasil dikumpulkan. Proses seleksi dilakukan dengan meninjau kualitas citra dan menyeragamkan label menjadi tiga kelas target (Baik, Retak, Kotor). Jumlah data sekunder yang masif ini difokuskan untuk memaksimalkan proses pembelajaran fitur pada arsitektur ViT.

3.2.2 Data Primer (*Local Augmentation Dataset*).

Data primer dikumpulkan untuk melengkapi dataset sekunder dan berfungsi vital sebagai data uji validitas lingkungan nyata (*real-world validation*).

Pengambilan data ini bertujuan untuk mengatasi kesenjangan domain (*domain gap*) antara citra internet dengan citra kamera *smartphone* yang akan digunakan pengguna.

Sebanyak 150 citra digital dikumpulkan secara spesifik menggunakan kamera iPhone 12 dengan pengaturan pencahayaan natural. Meskipun jumlahnya lebih sedikit dibandingkan data sekunder, jumlah ini telah memenuhi syarat sebagai sampel representatif untuk menguji ketahanan (*robustness*) model terhadap variasi pengambilan gambar lokal, khususnya pada fitur retak halus (*fine-grained*) dan noda kotoran yang spesifik.

Proses akuisisi citra dilakukan menggunakan perangkat optik kamera utama iPhone 12. Pemilihan perangkat ini didasarkan pada justifikasi teknis bahwa spesifikasi sensor kamera tersebut memenuhi standar minimum untuk ekstraksi fitur fine-grained, dengan rincian sebagai berikut:

1. Resolusi Sensor dan Densitas Piksel

   Kamera utama iPhone 12 memiliki resolusi 12 Megapiksel (4032x3024). Resolusi ini jauh melampaui kebutuhan *input* model ViT (224x224). Secara teoritis, proses *downsampling* dari resolusi tinggi ke rendah menggunakan interpolasi akan mempertahankan fitur struktural halus (seperti garis retak) dengan lebih tajam dibandingkan pengambilan citra langsung dengan resolusi rendah, sehingga meminimalkan *aliasing artifact* (Kuznetsova et al., 2021).
2. Sensitivitas Cahaya (*Aperture*)

   Lensa utama memiliki bukaan (aperture) sebesar f/1.6. Bukaan lebar ini memungkinkan sensor menangkap cahaya maksimal, yang secara signifikan mengurangi noise (bintik) pada gambar. Tingkat noise yang rendah sangat krusial karena noise digital seringkali dapat disalahartikan oleh model sebagai tekstur retakan palsu atau noda kotoran kecil.
3. Konsistensi Fokus

   Fitur Phase Detection Autofocus (PDAF) menjamin fokus terkunci pada permukaan cangkang, memastikan retak rambut dan noda kotoran terekam dengan tajam.

Prosedur pengambilan data dilakukan secara terkontrol menggunakan *tripod* dengan jarak tetap 15-20 cm dan latar belakang gelap (*matte black*) untuk memudahkan segmentasi visual objek.

3.3 Pemrosesan dan Penggabungan Data

Pada tahap ini data dari berbagai sumber harus melewati beberapa proses untuk yang harus sesuai dengan prosedur, antara lain:

1. Pembersihan & Penyeragaman Label

   Seluruh data (primer dan sekunder) akan ditinjau. Label diseragamkan menjadi tiga kelas kategorikal yakni 0: Baik, 1: Retak, 2: Kotor.

2. Transformasi & Normalisasi Seragam

   Seluruh gambar akan melalui *pipeline preprocessing* yang identik yakni ambar akan di-*resize* ke ukuran *input* standar ViT (224x224 piksel) selanjutnya nilai piksel dinormalisasi (dibagi 255) agar berada dalam rentang [0, 1], serta menstandarisasinya menggunakan nilai *mean* dan *standard deviation* dari ImageNet, agar sesuai dengan bobot *pre-trained* model.

3. Penyeimbangan dan Pemisahan Data (*Splitting*)

   Mengingat distribusi data awal yang tidak seimbang (*imbalanced*), di mana jumlah sampel kelas 'Retak' jauh lebih dominan dibandingkan kelas 'Kotor' dan 'Baik', diterapkan strategi Penyeimbangan Data (*Class Balancing*). Proses ini dilakukan dengan mengaugmentasi kelas minoritas secara selektif hingga jumlah sampel pada setiap kelas menjadi setara sebelum dibagi.

   Selanjutnya, dataset final dibagi menggunakan teknik *Stratified Random Sampling*, Teknik ini membagi data sebesar 80% untuk Latih, 10% untuk Validasi, dan 10% untuk Uji pada masing-masing kelas secara terpisah seperti tabel 3.1.

Tabel 3. 1 Penyeimbangan dan Pemisahan Data

No	Kelas	Total Gambar	Augmentation Balancing	Train 80%	Val 10%	Test 10%
1.	Retak	1,151	2,100	1,680	210	210

2.	Baik	664	2,100	1,680	210	210
3.	Kotor	655	2,100	1,680	210	210
Total		2,470	6,300	5,040	630	630

Tujuannya adalah untuk menjamin bahwa setiap partisi data (terutama Data Uji) memiliki representasi yang cukup dari setiap kelas, sehingga evaluasi model tidak bias ke kelas mayoritas.

a. Data Latih (*Train*): 90%. Digunakan untuk melatih model.
b. Data Validasi (*Validation*): 10%. Digunakan untuk memantau performa model selama *training* dan mencegah *overfitting*.
c. Data Uji (*Test*): Adalah data primer (150 gambar) yang diambil dan hanya digunakan satu kali di akhir penelitian untuk evaluasi final.

4. *Data Augmentation*

Untuk meningkatkan kemampuan generalisasi model dan mencegah bias terhadap latar belakang (*background bias*), diterapkan strategi augmentasi dinamis pada Data Latih. Berbeda dengan augmentasi standar, penelitian ini menerapkan augmentasi yang lebih agresif meliputi:

a. *Random Resized Crop*: Memotong bagian acak dari citra dan membesarkannya, memaksa model mempelajari tekstur lokal cangkang alih-alih bentuk global.
b. *Color Jitter*: Memvariasikan kecerahan, kontras, dan saturasi untuk mengurangi ketergantungan model terhadap kondisi pencahayaan tertentu.
c. *Geometric Transformations*: Rotasi acak dan pembalikan (*flip*) horizontal/vertikal untuk memperkaya variasi posisi objek.

3.4 Skenario Pengujian dan Evaluasi Kinerja

Performa model final yang telah dilatih akan dievaluasi menggunakan Data Uji yang belum pernah dilihat oleh model. Metrik evaluasi kuantitatif yang akan digunakan adalah:

1. *Confusion Matrix* adalah alat evaluasi kualitatif utama untuk menganalisis di mana model membuat kesalahan. Matriks ini akan menunjukkan, misalnya, berapa banyak telur Retak yang salah diprediksi sebagai Baik (False Negative), atau sebaliknya (False Positive).
2. *Accuracy* (Akurasi): Yaitu mengukur persentase total prediksi yang benar menngunakan persamaan berikut:

$$Accuracy = \frac{\sum_{i=1}^{3} X_{ii}}{\sum_{i=1}^{3} \sum_{j=1}^{3} X_{ij}} \quad \ldots\ldots (3.2)$$

   Keterangan:
   - i = Baris Aktual
   - j = Baris Prediksi
   - $\sum_{i=1}^{3} Xii$ = Penjumlahan elemen pada diagonal utama (prediksi yang benar untuk kelas 1, 2, dan 3). Ini menggantikan konsep TP total.
   - $\sum_{i=1}^{3} \sum_{j=1}^{3} Xij$ = Penjumlahan seluruh sel dalam matriks (baris i dan kolom j). Ini adalah total seluruh data uji.
   - Xij = Jumlah data pada baris ke-i (aktual) dan kolom ke-j (prediksi).
3. *Precision* (Presisi): Yaitu mengukur tingkat "prediksi palsu". Sangat penting untuk kelas 'Retak' (Seberapa yakin kita bahwa yang diprediksi 'Retak' itu benar-benar 'Retak'?) menngunakan persamaan berikut:

$$Precision_k = \frac{X_{kk}}{\sum_{i=1}^{3} X_{ik}} \quad \ldots\ldots (3.3)$$

   Keterangan:
   - i = Baris Aktual
   - k = Kelas
   - X_{kk} = Prediksi benar kelas k (Diagonal).
   - $\sum_{i=1}^{3} X_{ik}$ = Jumlah total data pada kolom kelas k (Total yang *diprediksi* sebagai kelas k).
4. *Recall* (Sensitivitas): yaitu mengukur tingkat "kegagalan deteksi". Ini adalah metrik paling krusial dalam penelitian ini. Recall yang tinggi untuk

kelas 'Retak' berarti model berhasil menemukan hampir semua telur retak, yang sangat penting untuk keamanan pangan menngunakan persamaan berikut:

$$Recall_k = \frac{X_{kk}}{\sum_{j=1}^{3} X_{ij}} \quad \ldots\ldots (3.4)$$

Keterangan:

- k = Kelas
- j = Baris Prediksi
- X_{kk} = Prediksi benar kelas k (Diagonal).
- $\sum_{j=1}^{3} X_{ij}$ = Jumlah total data pada baris kelas k (Total yang *sebenarnya* kelas k).

5. *F1-Score*: yaitu rata-rata harmonik dari Precision dan Recall, memberikan skor tunggal yang menyeimbangkan kedua metrik tersebut menngunakan persamaan berikut:

$$F1_k = 2 \times \frac{Precision_k \times Recall_k}{Precision_k + Recall_k} \quad \ldots\ldots (3.5)$$

Keterangan:

- k = Kelas
- $F1_k$ = Nilai F1-Score untuk kelas ke - k
- $Precision_k$ = Nilai presisi untuk kelas ke - k
- $Recall_k$ = Nilai sensitivitas/recall untuk kelas ke - k

3.5 Alat dan Bahan Penelitian

Penelitian ini akan didukung oleh perangkat keras (*hardware*) dan perangkat lunak (*software*) berikut:

1. Perangkat Keras (*Hardware*):
   a. Laptop LENOVO IDEAPAD 330 dengan *processor* INTEL CORE i5-8250U, dengan RAM 8GB dan *storage* 256 GB (SSD) & 1 TB (HDD).
   b. Kamera iPhone 12 untuk pengambilan data primer.

c. *Graphics Processing Unit* (GPU) atau Kartu Grafis Eksternal: Proses *training* model *Deep Learning* akan dilakukan menggunakan *Google Colaboratory* (Colab) untuk memanfaatkan akses ke GPU (seperti NVIDIA T4 atau A100) agar proses komputasi lebih cepat.

2. Perangkat Lunak (*Software*):
   a. Bahasa Pemrograman: Python (versi 3.9+).
   b. Lingkungan Pengembangan: Google Colab (Notebook Jupyter).
   c. *Framework/Library* Utama:
      1) *Hugging Face transformers*: *Library* standar industri yang digunakan untuk mengunduh, memuat, dan melakukan *fine-tuning* pada model ViT *pre-trained*.
      2) *PyTorch* atau *TensorFlow 2.x*: Sebagai *backend framework* yang menjalankan *library transformers*.
      3) *Scikit-learn (Sklearn)*: Digunakan untuk membagi data (train/test split) dan menghitung metrik evaluasi (Confusion Matrix, Akurasi, Presisi, Recall, F1-Score).
      4) *OpenCV* dan *PIL (Pillow)*: Digunakan untuk tugas *preprocessing* dan augmentasi gambar.
      5) *Pandas* dan *NumPy*: Digunakan untuk manipulasi data dan label.
      6) *Matplotlib* dan *Seaborn*: Digunakan untuk visualisasi data dan hasil (plot *loss/accuracy* dan *heatmap* Confusion Matrix).

BAB IV
HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil pelatihan, dan evaluasi model Vision Transformer (ViT) dalam mengklasifikasikan kualitas cangkang telur. Analisis dilakukan secara komprehensif, mencakup metrik performa kuantitatif dan analisis visual untuk memvalidasi fokus atensi model.

4.1 Hasil Pelatihan Model

Pelatihan model dilakukan menggunakan arsitektur ViT-B/16 (*pre-trained ImageNet-21k*) dengan skema *Transfer Learning*. Proses *fine-tuning* dijalankan selama 30 epoch menggunakan *optimizer* AdamW dengan laju pembelajaran (*learning rate*) sebesar 1e-5 dan penjadwalan *Cosine Decay*. Mekanisme *Label Smoothing* diterapkan untuk mencegah model menjadi terlalu percaya diri (*overconfident*) pada data yang ambigu.

4.1.1 Konfigurasi Hyperparameter Pelatihan

Untuk mencapai konvergensi model yang optimal dan mencegah terjadinya *overfitting*, diterapkan konfigurasi *hyperparameter* yang disesuaikan dengan karakteristik dataset dan arsitektur *Vision Transformer*. Rincian parameter pelatihan utama disajikan pada Tabel 4.1.

Tabel 4. 1 Konfigurasi Parameter Pelatihan

No	Parameter	Nilai	Justifikasi
1	*Epoch*	30	Memberikan iterasi yang cukup bagi model untuk mempelajari fitur *fine-grained* tanpa *overfitting*.
2	*Batch Size*	32	Mengoptimalkan penggunaan memori GPU A100 atau A4 dan stabilitas gradien.
3	*Learning Rate*	1e-5	Nilai rendah dipilih untuk menjaga bobot *pre-trained* agar tidak rusak saat proses *fine-tuning*.

4	*Warmup Ratio*	0.1	Tahap pemanasan untuk menstabilkan proses adaptasi awal model.
5	*Weight Decay*	0.1	Regularisasi untuk mengurangi kompleksitas model dan mencegah model menghafal *noise* latar belakang.
6	*Label Smoothing*	0.1	Teknik regularisasi untuk mencegah model menjadi *overconfident* (terlalu yakin) pada prediksi yang salah, meningkatkan kemampuan generalisasi pada data asing.
7	*Metric for Best Model*	*Recall*	Prioritas penyimpanan model didasarkan pada nilai Recall tertinggi guna meminimalkan risiko lolosnya telur cacat (*False Negative*).

Penerapan *Label Smoothing* dan *Weight Decay* secara spesifik ditujukan untuk mengatasi tantangan bias model terhadap latar belakang yang ditemukan pada eksperimen awal, sehingga model dipaksa untuk mempelajari fitur tekstur cangkang yang lebih esensial.

4.1.2 Skenario Pelatihan Model

Proses pelatihan diorkestrasi menggunakan kelas Trainer dari pustaka Hugging Face, yang mengintegrasikan model, data, dan konfigurasi pelatihan dalam satu *pipeline* eksekusi yang efisien. Komponen-komponen kunci dalam inisialisasi Trainer meliputi:

1. Manajemen Data: Dataset latih (*train*) dan validasi (*validation*) dipisahkan secara eksplisit untuk memantau generalisasi model secara *real-time*.
2. Pemrosesan Citra: Objek *ViTImageProcessor* digunakan untuk menstandarisasi input citra (resize, normalisasi), menjamin kompatibilitas dengan bobot *pre-trained*.
3. Evaluasi Metrik: Fungsi *compute_metrics* diintegrasikan untuk menghitung metrik performa komprehensif (Akurasi, Presisi, Recall, F1-Score) pada setiap akhir *epoch*, bukan hanya memantau penurunan *loss*.

4. Mekanisme Penghentian Otomatis (*Early Stopping*): Untuk mencegah *overfitting* akibat pelatihan yang berlebihan (*overtraining*), diterapkan *callback* EarlyStopping dengan parameter *patience* sebesar 7 epoch. Mekanisme ini secara otomatis menghentikan proses pelatihan jika tidak terdeteksi peningkatan performa pada data validasi selama 7 iterasi berturut-turut, dan mengembalikan bobot model ke kondisi terbaik (*best checkpoint*).

4.1.3 Identifikasi Peningkatan Akurasi pada Beberapa Epoch

Berdasarkan hasil pemantauan proses *training* dan *validation* yang tersimpan secara otomatis dalam *log history* Trainer, kinerja model memperlihatkan tren peningkatan performa yang progresif dan stabil.

Data historis dari proses pelatihan dianalisis untuk melihat dinamika perubahan nilai Akurasi (*Accuracy*) dan Recall dari fase inisialisasi hingga *epoch* terakhir. Analisis ini bertujuan untuk mengidentifikasi titik-titik infleksi di mana model mengalami pembelajaran fitur yang signifikan, khususnya dalam membedakan kelas yang memiliki kemiripan visual tinggi.

Rincian peningkatan metrik kinerja model pada titik-titik *epoch* krusial disajikan pada Gambar 4.1.

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	0.942100	0.888362	0.903175	0.915905	0.903175	0.904795
2	0.431600	0.433821	0.965079	0.965970	0.965079	0.965256
3	0.343600	0.328171	0.987302	0.987302	0.987302	0.987302
4	0.309400	0.305280	0.995238	0.995305	0.995238	0.995238
5	0.312000	0.298973	0.998413	0.998420	0.998413	0.998413
6	0.307100	0.297133	0.998413	0.998420	0.998413	0.998413
7	0.312800	0.296032	0.996825	0.996855	0.996825	0.996825
8	0.308700	0.300791	0.993651	0.993769	0.993651	0.993650
9	0.298400	0.296952	0.995238	0.995305	0.995238	0.995238
10	0.299800	0.294842	0.998413	0.998420	0.998413	0.998413
11	0.296300	0.294793	0.998413	0.998420	0.998413	0.998413
12	0.295400	0.293839	1.000000	1.000000	1.000000	1.000000
13	0.291500	0.295090	0.996825	0.996855	0.996825	0.996825
14	0.295400	0.292243	1.000000	1.000000	1.000000	1.000000
15	0.299100	0.292320	1.000000	1.000000	1.000000	1.000000
16	0.293800	0.291722	1.000000	1.000000	1.000000	1.000000
17	0.297400	0.292400	1.000000	1.000000	1.000000	1.000000
18	0.291300	0.292034	1.000000	1.000000	1.000000	1.000000
19	0.291400	0.292624	0.998413	0.998420	0.998413	0.998413

Gambar 4. 1 Peningkatan Akurasi pada Beberapa Epoch

Berdasarkan data pada gambar di atas, dinamika pembelajaran model dapat diuraikan dalam tiga fase utama:

a. Fase Adaptasi Cepat (*Epoch* 1-3): Pada fase awal, terjadi lonjakan akurasi dan penurunan *loss* yang sangat tajam. Hal ini mengindikasikan efektivitas teknik *Transfer Learning*, di mana bobot *pre-trained* ViT (ImageNet) dengan cepat menyesuaikan diri (*fine-tuning*) terhadap fitur dasar cangkang telur. Model mulai mampu memisahkan kelas yang berbeda secara kontras.
b. Fase Pembelajaran Fitur Halus *(Epoch* 4-15): Peningkatan metrik mulai melandai namun tetap konsisten positif. Pada tahap ini, mekanisme *Self-Attention* pada model mulai bekerja lebih dalam untuk mempelajari fitur *fine-grained*, seperti membedakan tekstur alami cangkang dengan garis

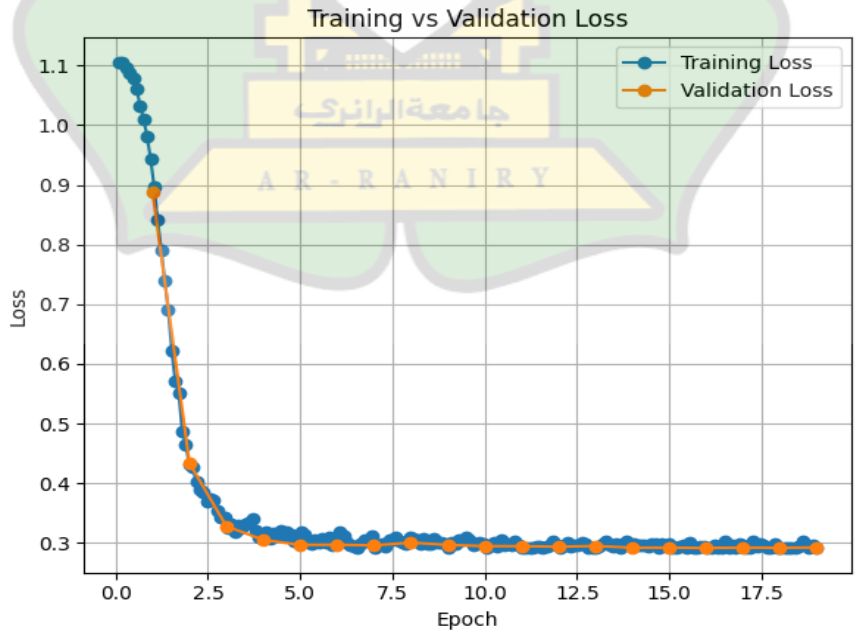
retak rambut yang tipis. Kenaikan nilai *Recall* pada fase ini membuktikan bahwa model semakin sensitif terhadap cacat minor.

c. Fase Konvergensi (*Epoch* Akhir): Menjelang akhir pelatihan, nilai *loss* dan akurasi mencapai titik stabil *(plateau*). Fluktuasi nilai menjadi sangat minim, menandakan bahwa model telah mencapai kapasitas maksimalnya dalam mempelajari pola dari dataset yang tersedia tanpa mengalami overfitting.

4.1.4 Analisa Kurva Pembelajaran Model (*Learning Curve*)

Visualisasi dinamika pelatihan model disajikan pada Gambar 4.2. Grafik ini merepresentasikan log aktivitas pembelajaran model dari epoch awal hingga akhir. Untuk memahami performa model secara komprehensif, berikut adalah penjabaran elemen visual pada grafik tersebut:

1. Sumbu Horizontal (*X-Axis*) - *Epoch*: Merepresentasikan durasi pelatihan dalam satuan iterasi penuh terhadap seluruh dataset. Rentang 0 hingga 30 menunjukkan perjalanan waktu model dalam mempelajari fitur visual.
2. Sumbu Vertikal Kiri (*Y-Axis*) - *Loss*: Merepresentasikan nilai *Cross-Entropy Loss*, yaitu indikator "kesalahan" model. Semakin rendah nilai ini (mendekati 0), semakin akurat prediksi model dibandingkan dengan label sebenarnya.



Gambar 4. 2 Kurva Pembelajaran (Learning Curve)

Berdasarkan gambar 4.2, berikut adalah analisis mendalam mengenai masing-masing komponen evaluasi:

1. *Training Loss* (Adaptasi Fitur): Pada kurva *Training Loss* (garis biru), tercatat penurunan nilai yang substansial dari kisaran 0.94 di fase inisialisasi menjadi 0.29 di akhir pelatihan. Penurunan tajam pada 3 epoch pertama membuktikan efektivitas *Transfer Learning*, di mana bobot *pre-trained* ViT berhasil dikalibrasi ulang dengan cepat untuk mengenali fitur tekstur cangkang telur. Gradien penurunan yang stabil menuju nilai minimum menunjukkan bahwa model terus belajar memperbaiki kesalahannya di setiap iterasi tanpa mengalami hambatan optimasi (*stuck*).
2. *Validation Loss* (Kemampuan Generalisasi): Kurva *Validation Loss* (garis oranye) memperlihatkan tren penurunan yang bergerak harmonis dengan data latih, turun dari 0.88 ke 0.29. Penurunan ini adalah indikator krusial bahwa model tidak sekadar "menghafal" data latih, melainkan mampu menggeneralisasi pengetahuannya ke data baru (*unseen data*). Meskipun terdapat fluktuasi minor akibat variasi stokastik pada *batch* validasi, tren keseluruhannya tetap menurun (konvergen), yang menandakan stabilitas model yang tinggi.

Secara keseluruhan, analisis komparatif antara kurva *Training* dan *Validation* menunjukkan tidak adanya indikasi overfitting. Hal ini dibuktikan dengan dua fenomena visual:

1. Kurva validasi terus menurun beriringan dengan kurva training hingga akhir epoch (tidak berbelok naik/divergen).
2. Jarak (*gap*) antara nilai *loss* training dan validasi sangat tipis yakni 0.0012.

Kondisi *Good Fit* ini mengonfirmasi bahwa strategi Augmentasi Agresif (*Random Crop* & *Color Jitter*) dan mekanisme Label Smoothing yang diterapkan berfungsi efektif dalam menjaga model tetap objektif dan *robust*.

4.2 Evaluasi Peforma Model

Evaluasi akhir dilakukan secara objektif menggunakan Data Uji (*Test Set*), yaitu data baru yang tidak pernah dilihat model selama proses pelatihan maupun validasi.

4.2.1 Analisa Matrix Kuantitatif

Hasil pengukuran performa model terhadap tiga kelas target disajikan dalam tabel 4.2.

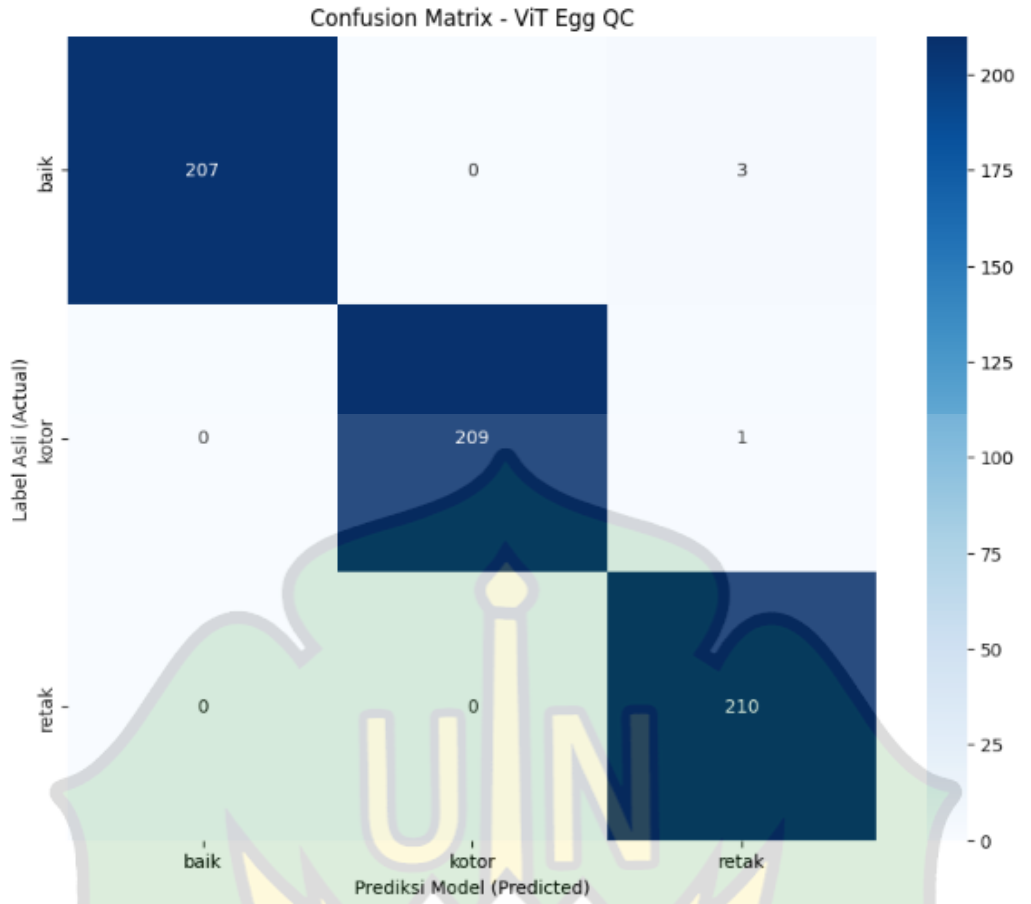
Tabel 4. 2 Matrix Kuantitatif

No	Kelas	Recall	Precision	F1-Score	Support
1	Baik	0.99	1.00	0.99	210
2	Retak	1.00	1.00	1.00	210
3	Kotor	1.00	0.98	0.99	210
Macro Avg		0.99	0.99	0.99	630
Weighted Avg		0.99	0.99	0.99	630
Accuracy				0.99	630

Hasil pengujian menunjukkan bahwa model mampu mencapai akurasi global sebesar 0.99. Poin terpenting dalam penelitian ini adalah nilai *Recall* untuk kelas 'Retak' yang mencapai 1.00. Tingginya nilai *Recall* ini menegaskan keandalan model dalam mendeteksi produk cacat, sehingga meminimalkan risiko lolosnya telur retak (*False Negative*) yang dapat membahayakan keamanan pangan konsumen.

4.2.2 Analisa Confusion Matrix

Evaluasi performa model tidak cukup hanya dilihat dari akurasi global. Untuk memahami perilaku model secara granular dalam membedakan antar-kelas, digunakan visualisasi *Confusion Matrix* sebagaimana disajikan pada gambar 4.4.



Gambar 4. 3 Confusion Matrix Data Test

Matriks ini merepresentasikan distribusi prediksi model terhadap data uji (*Test Set*) yang berjumlah total 630 citra. Sumbu vertikal (Y-Axis) menunjukkan Label Asli (*Actual Class*), sedangkan sumbu horizontal (X-Axis) menunjukkan Prediksi Model (*Predicted Class*). Diagonal utama (dari kiri atas ke kanan bawah) merepresentasikan jumlah prediksi yang BENAR (*True Positive*), sementara area di luar diagonal menunjukkan kesalahan klasifikasi (*Misclassification*).

Berikut adalah analisis rinci performa model pada masing-masing kelas berdasarkan matriks tersebut:

1. Analisis Kelas 'Baik' (Telur Utuh) Pada baris pertama matriks, dari total 210 citra telur kondisi baik, model berhasil mengidentifikasi 207 citra dengan tepat.
    a) Kesalahan: Terdapat sedikit kesalahan dimana 3 citra telur baik diprediksi sebagai 'Retak'. Kesalahan ini kemungkinan disebabkan oleh adanya *noise* visual seperti bayangan atau tekstur kulit telur yang kasar,

yang oleh model diasosiasikan sebagai anomali. Namun, tingkat kesalahan ini sangat rendah, menunjukkan model memiliki spesifisitas yang tinggi dalam mengenali telur sehat.

2. Analisis Kelas 'Retak' (Prioritas Keamanan Pangan) Kelas ini merupakan fokus utama penelitian karena berkaitan dengan risiko keamanan pangan. Dari total 210 citra telur retak:
   a) Keberhasilan Deteksi: Model berhasil memprediksi dengan benar sebanyak 210 citra. Tingginya angka pada diagonal ini berkontribusi langsung pada nilai *Recall* yang tinggi.
3. Analisis Kelas 'Kotor' (Deteksi Noda) Pada kelas telur kotor, model menunjukkan performa yang sangat solid dengan mendeteksi benar sebanyak 209 dari 210 sampel.
   a) *Discriminative Power*: Kemampuan model membedakan 'Kotor' dan 'Retak' terlihat sangat baik, dengan hanya 1 kesalahan tertukar. Ini membuktikan bahwa arsitektur ViT berhasil mempelajari perbedaan fitur tekstur antara "garis retakan" (fitur linear) dan "noda kotoran" (fitur blok/bercak tak beraturan), meskipun keduanya sama-sama merupakan anomali visual pada cangkang.

Secara keseluruhan, *Confusion Matrix* menunjukkan pola diagonal yang sangat dominan dengan nilai *off-diagonal* (kesalahan) yang minimal. Hal ini mengindikasikan bahwa model ViT-B/16 yang dikembangkan memiliki kemampuan generalisasi yang seimbang (*balanced performance*) antar kelas. Strategi Augmentasi Anti-Bias dan Penyeimbangan Data (*Balancing*) terbukti efektif mencegah model bias ke satu kelas tertentu, sehingga model tidak sekadar "menebak" kelas mayoritas, melainkan benar-benar mengenali fitur visual masing-masing kondisi telur.

4.3 Evaluasi Visualisasi Hasil Model

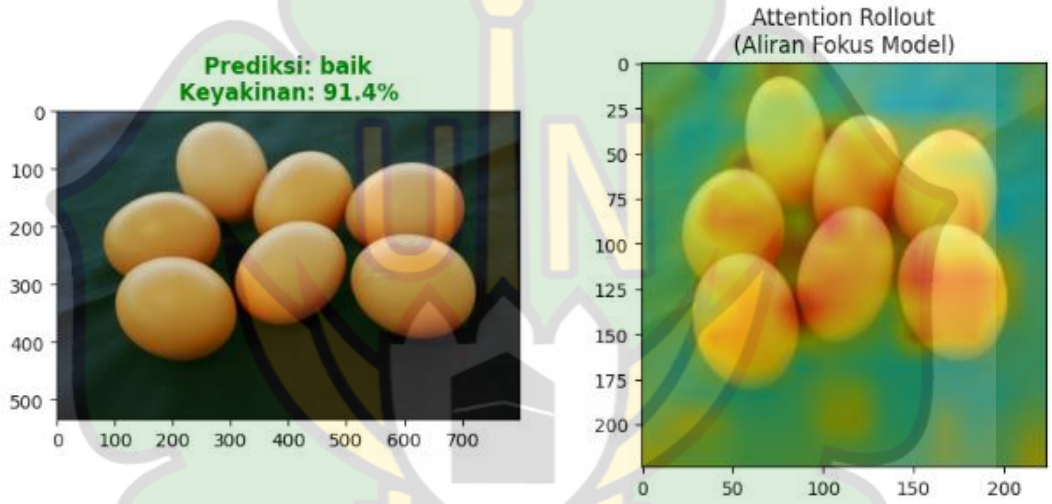
Salah satu keunggulan fundamental arsitektur *Vision Transformer* (ViT) dibandingkan CNN konvensional adalah transparansi mekanisme pengambilan keputusannya melalui *Self-Attention*. Penelitian ini menerapkan visualisasi Attention Map menggunakan metode *Attention Rollout* untuk memvalidasi area

mana pada citra yang dianggap penting oleh model dalam menentukan kelas prediksi.

Evaluasi visual dilakukan secara terstruktur pada tiga kelas target menggunakan data uji acak (*random samples*) yang diambil dari lingkungan nyata (luar dataset latih, validasi, dan testing) untuk menguji generalisasi model.

4.3.1 Analisa Visualisasi Telur Baik

Pengujian dilakukan pada citra telur untuk melihat area mana yang menjadi fokus atensi model. Pengujian pertama dilakukan pada citra telur dalam kondisi Baik (Utuh). Karakteristik visual kelas ini adalah permukaan cangkang yang homogen, bersih, dan tanpa fitur topologi yang menonjol (seperti garis atau noda).



Gambar 4. 4 Visualisasi Attention Rollout Telur Baik

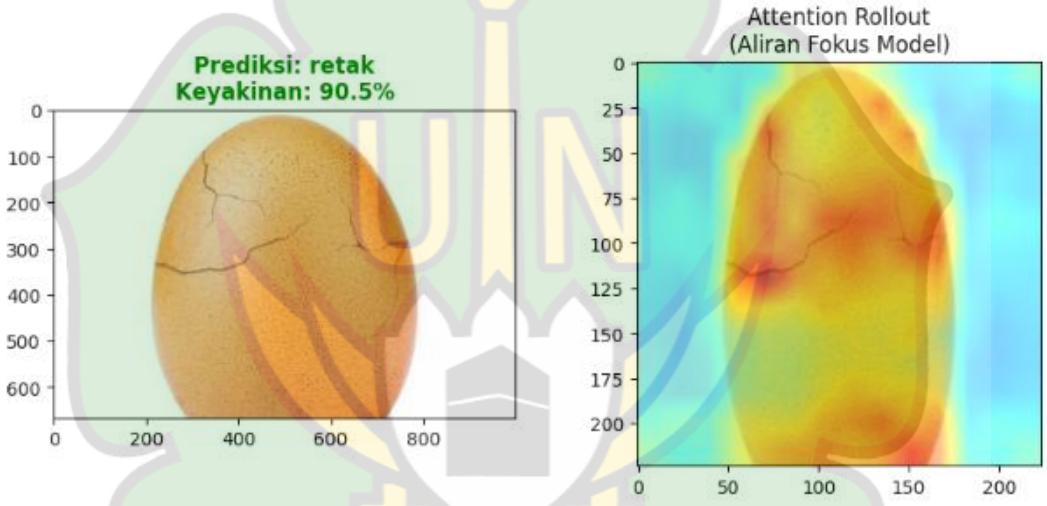
Berdasarkan gambar 4.5, hasil visualisasi *heatmap* menunjukkan fenomena menarik terkait perilaku model pada objek tanpa fitur spesifik:

1. Distribusi Atensi: Atensi model pada kelas 'Baik' terlihat lebih menyebar (*diffuse*) atau cenderung menyoroti area objek dan tepian objek, bukan pada bagian tertentu.
2. Analisis Penyebab: Hal ini terjadi karena pada telur baik, tidak terdapat fitur diskriminatif kuat (seperti goresan retak) yang dapat "mengunci" atensi model. Akibatnya, mekanisme *Multi-Head Self-Attention* berusaha mencari konteks global dari lingkungan sekitar objek untuk mengonfirmasi keberadaan telur tersebut.

3. Implikasi: Meskipun fokus visual tampak bias ke latar belakang, model tetap berhasil memprediksi kelas ini dengan benar (atau dengan *confidence* yang wajar). Ini mengindikasikan bahwa untuk kelas 'Baik', model menggunakan strategi "Peniadaan Fitur" (*Absence of Features*), di mana ketiadaan pola retak/kotor pada area tengah citra menjadi dasar keputusan klasifikasi.

4.3.2 Analisa Visualisasi Telur Retak

Kelas ini merupakan prioritas utama dalam sistem keamanan pangan. Pengujian dilakukan untuk memastikan model mampu mendeteksi garis retakan (*crack line*) yang seringkali sangat tipis (*fine-grained*).



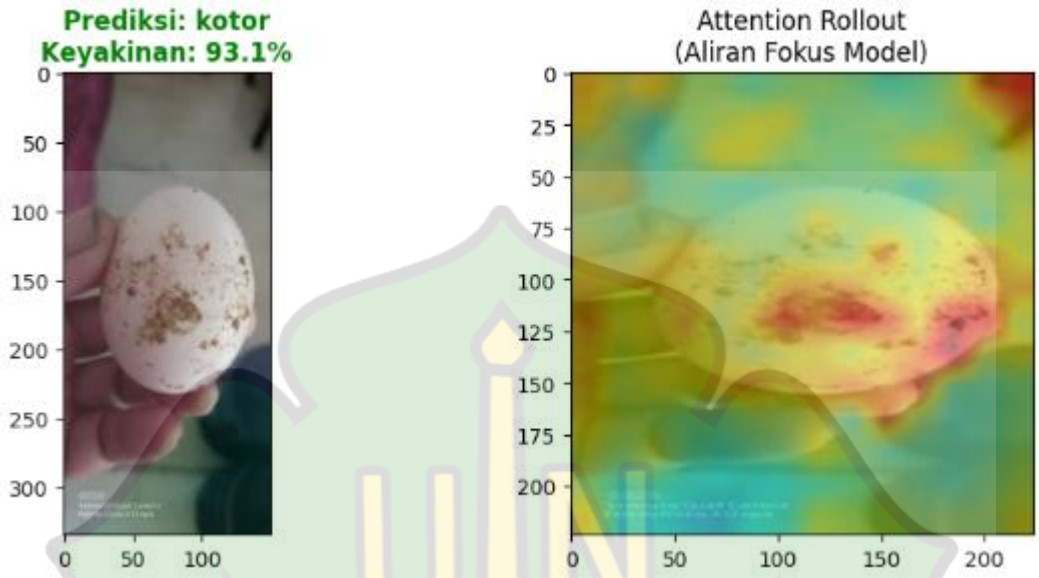
Gambar 4. 5 Visualisasi Attention Rollout Telur Retak

Hasil visualisasi pada gambar 4.6 memperlihatkan kemampuan *fine-grained classification* dari arsitektur ViT:

1. Lokalisasi Cacat: *Heatmap* menunjukkan pola linear yang mengikuti alur garis retakan pada cangkang telur.
2. Sensitivitas: Bahkan pada retakan rambut yang tipis, mekanisme *Attention* mampu menangkap diskontinuitas tekstur pada cangkang dan menjadikannya dasar prediksi utama.
3. Kesimpulan: Fokus yang presisi ini memvalidasi bahwa tingginya nilai *Recall* pada evaluasi kuantitatif didukung oleh kemampuan model dalam mengenali fitur visual retakan yang sesungguhnya, bukan artefak lain.

4.3.3 Analisa Visualisasi Telur Kotor

Pengujian selanjutnya dilakukan pada citra telur dengan kondisi Kotor. Fitur visual utama pada kelas ini adalah adanya noda, bercak tanah, atau kotoran yang memiliki kontras warna berbeda dengan cangkang.



Gambar 4. 6 Visualisasi Attention Rollout Telur Kotor

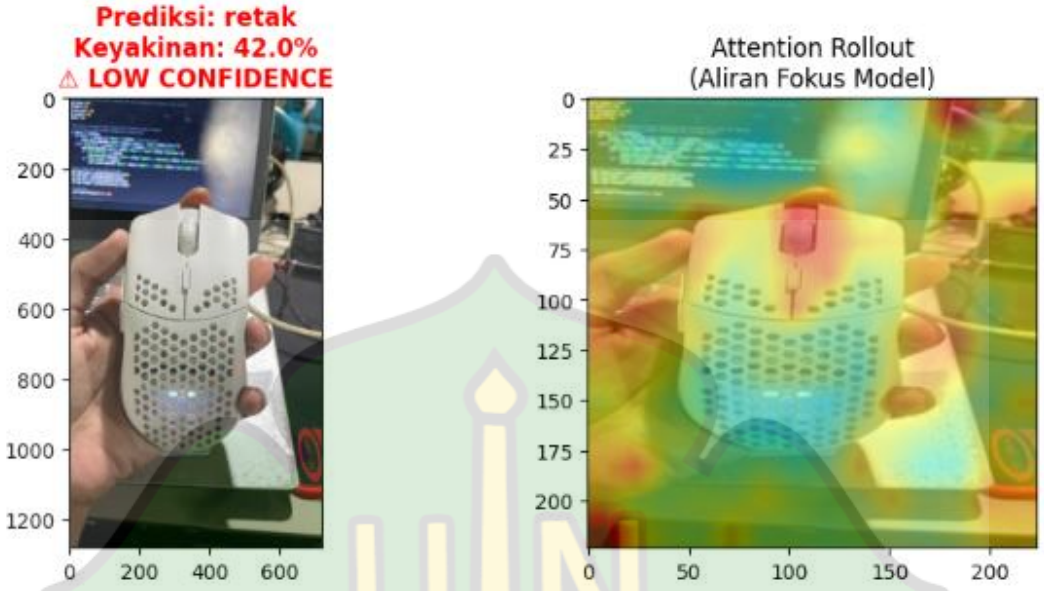
Sebagaimana ditunjukkan pada gambar 4.7, visualisasi *Attention Map* menunjukkan korelasi yang kuat antara atensi model dengan fitur fisik objek:

1. Fokus Atensi: Area berwarna merah (intensitas tinggi) terkonsentrasi secara spesifik pada area noda atau bercak kotoran di permukaan cangkang.
2. Validasi Fitur: Model berhasil mengabaikan area cangkang yang bersih dan latar belakang, serta secara selektif memberikan bobot tinggi pada piksel yang merepresentasikan anomali warna (kotoran). Hal ini membuktikan bahwa model telah mempelajari representasi fitur "Kotor" dengan tepat, bukan sekadar menghafal bentuk telur.

4.3.4 Analisa Visualisasi Objek Non Telur

Selain mengevaluasi akurasi pada dataset uji standar, penelitian ini melakukan pengujian ketahanan (*robustness testing*) dengan memaparkan model pada data *Out-of-Distribution* (OOD). Data OOD adalah citra objek yang tidak termasuk dalam kategori kelas pelatihan, dilakukan pengujian menggunakan citra objek asing (non-telur).

Dalam eksperimen ini, digunakan citra mouse komputer berwarna putih yang difoto menyerupai kondisi pengambilan data telur. Hasilpengujian divisualisasikan pada gambar 4.8.



Gambar 4. 7 Visualisasi Attention Rollout Objek Non Telur

Berdasarkan hasil tersebut, dilakukan analisis mendalam dari dua perspektif:

1. Analisis Kuantitatif: Tingkat Keyakinan sebuah model klasifikasi tertutup (*closed-set classifier*) seperti ViT secara default akan "dipaksa" memilih salah satu kelas yang tersedia (Baik/Retak/Kotor) karena sifat fungsi aktivasi *Softmax* yang menjumlahkan probabilitas menjadi 1. Namun, hasil pengujian menunjukkan fenomena positif:
   a) Model memberikan prediksi label 'Retak' namun dengan Tingkat Keyakinan yang Rendah, yaitu di angka 42%
   b) Nilai ini berada jauh di bawah ambang batas aman (*safety threshold*) yang ditetapkan dalam penelitian ini.
   c) Interpretasi: Rendahnya skor ini mengindikasikan tingginya Ketidakpastian Model (*Model Uncertainty*). Model mendeteksi adanya anomali fitur yang tidak cocok dengan representasi vektor (*embedding*) dari kelas telur manapun yang telah dipelajarinya.

2. Analisis Kualitatif: Distribusi Atensi (*Attention Map*) Visualisasi *Heatmap* memberikan bukti mengapa model merasa "ragu-ragu":
    a) Dispersi Atensi: Berbeda dengan citra telur di mana area merah terfokus tajam pada bagian retakan atau kotoran, pada citra mouse, area merah terlihat menyebar (*scattered*) atau menyoroti fitur yang tidak relevan seperti *mousewheel* pada bagian tengah mouse
    b) Absensi Fitur Kunci: Model tidak menemukan pola tekstur pori-pori cangkang atau pola retakan spesifik yang menjadi tanda kelas telur. Ketidakhadiran fitur esensial ini menyebabkan mekanisme *Self-Attention* gagal membentuk bobot yang kuat pada satu area tertentu.
3. Kesimpulan Pengujian OOD Kegagalan model dalam mengklasifikasikan mouse dengan *confidence* tinggi justru merupakan Keberhasilan Sistem. Hal ini membuktikan efektivitas dua strategi yang diterapkan:
    a) Augmentasi Agresif (*Random Crop*): Memaksa model untuk tidak bergantung pada bentuk bulat global atau warna putih semata, melainkan detail tekstur mikro. Karena tekstur mouse berbeda dengan kalsium cangkang telur, model menolaknya.
    b) *Label Smoothing*: Mencegah model menjadi *overconfident* pada data yang memiliki fitur ambigu.

Secara implisit, sistem ini memiliki mekanisme *fail-safe* yang mampu menyaring objek asing di lingkungan produksi nyata, mencegah terjadinya kesalahan klasifikasi fatal (*False Positive*) yang dapat mengganggu proses *Quality Control*.

4.4 Evaluasi *Cross-Domain Method*

Berbeda dengan skenario pengujian konvensional yang membagi satu dataset menjadi data latih dan uji, penelitian ini juga menerapkan skenario evaluasi *Cross-Domain*. Pendekatan ini dirancang untuk menguji batas kemampuan generalisasi model dengan melatihnya pada satu distribusi data (*Source Domain*) dan mengujinya pada distribusi data yang benar-benar berbeda (*Target Domain*).

Dalam konteks cross domain ini, Data Sekunder bertindak sebagai *Source Domain* tempat model mempelajari fitur, sedangkan Data Primer bertindak sebagai *Target Domain* yang merepresentasikan kondisi implementasi nyata di lapangan.

4.4.1 Konfigurasi Eksperimen dan Konfigurasi Data

Untuk memastikan validitas pengujian lintas-domain, diterapkan protokol pemisahan data yang ketat tanpa adanya kebocoran data (*data leakage*) antara sumber dan target. Rincian distribusi data yang digunakan dalam eksperimen *Cross-Domain* disajikan pada tabel 4.3.

Tabel 4. 3 Konfigurasi Data *Cross Domain Method*

No	Sumber Data	Peran Data	Pembagian	Jumlah Data	Fungsi
1	Sekunder	*Source Domain*	90:10	Train: 2.088 Val: 232	Melatih Bobot Model & Validasi Model
2	Primer	*Target Domain*	Full	Test: 150	Evaluasi Peforma Akhir

Pada fase *training*, data *train* (2.088) mengalami augmentasi dinamis hingga mencapai keseimbangan 2.100 per kelas (Baik, Retak, Kotor), adapun total data *train* perkelas untuk distribusi metode cross domain ini tersaji pada tabel 4.4

Tabel 4. 4 Distribusi dan Penyeimbangan Data Sekunder

No	Kelas	Data Awal	Train (90%)	Val (10%)	Augmentasi *Balancing* Data Train
1	Baik	630	567	63	2,100
2	Retak	1,110	999	111	2,100
3	Kotor	580	522	58	2,100
Total		2,320	2,088	232	6,300

Pelatihan dilakukan dalam durasi jangka panjang untuk memastikan model Vision Transformer (ViT) yang *data-hungry* dapat mencapai konvergensi fitur yang

stabil. Parameter pelatihan yang digunakan dirancang khusus untuk menangani risiko *overfitting* pada epoch panjang, sebagaimana dirincikan pada tabel 4.5

Tabel 4. 5 Parameter Latihan *Cross Domain Method*

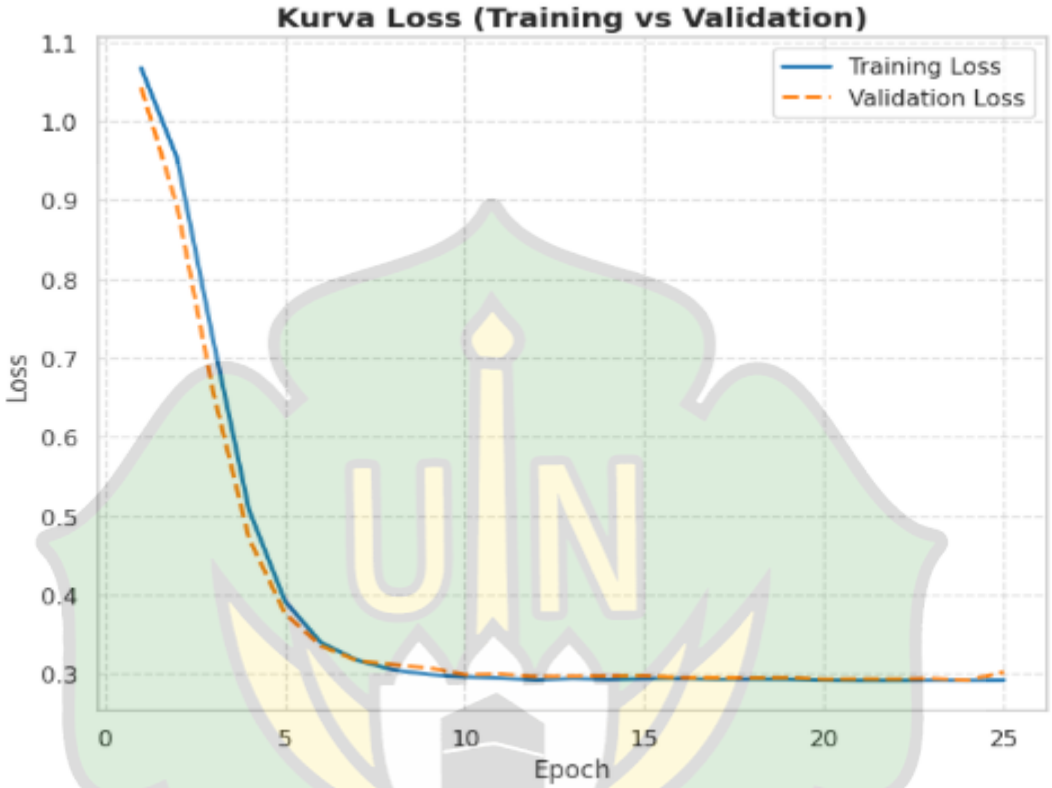
No	Parameter	Nilai	Justifikasi
1	*Epoch*	200	Memberikan waktu yang cukup bagi *Self-Attention* untuk mempelajari fitur global secara mendalam.
2	*Batch Size*	32	Mengoptimalkan penggunaan memori GPU A100 atau A4 dan stabilitas gradien.
3	*Learning Rate*	1e-5	*Learning rate* rendah untuk *fine-tuning* yang presisi tanpa merusak bobot *pre-trained*.
4	*Warmup Ratio*	0.1	Tahap pemanasan untuk menstabilkan proses adaptasi awal model.
5	*Weight Decay*	0.1	Regularisasi untuk mengurangi kompleksitas model dan mencegah model menghafal *noise* latar belakang.
6	*Label Smoothing*	0.1	Teknik regularisasi untuk mencegah model menjadi *overconfident* (terlalu yakin) pada prediksi yang salah, meningkatkan kemampuan generalisasi pada data asing.
7	Validasi	*Metric-Based*	Model terbaik disimpan berdasarkan nilai *Recall* tertinggi, bukan *Loss* terendah.

4.4.2 Konfigurasi Eksperimen dan Konfigurasi Data

Meskipun skenario pelatihan dialokasikan untuk batas maksimum 200 epoch, mekanisme *Early Stopping* yang diterapkan berhasil mengidentifikasi titik

optimal konvergensi lebih awal. Berdasarkan pantauan *monitoring* validasi, proses pelatihan dihentikan secara otomatis pada Epoch ke-25.

Grafik pembelajaran (*Learning Curve*) yang merepresentasikan dinamika *Training Loss* dan *Validation Loss* disajikan pada gambar 4.8.



Gambar 4. 8 Kurva Pembelajaran *Cross-Domain*

Berdasarkan visualisasi kurva di atas, dapat dianalisis karakteristik pembelajaran model sebagai berikut:

1. Efisiensi Konvergensi (*Early Convergence*): Model menunjukkan kemampuan adaptasi fitur yang sangat cepat. Penurunan *loss* yang signifikan terjadi pada 5 epoch pertama (fase adaptasi), kemudian melandai (*plateau*) mulai dari epoch ke-12. Penghentian otomatis pada epoch ke-25 menunjukkan bahwa model telah mencapai performa puncaknya dan penambahan durasi pelatihan tidak lagi memberikan keuntungan signifikan, melainkan hanya meningkatkan risiko *overfitting*.
2. Indikasi *Good Fit* (Kurva Berhimpitan): Berbeda dengan gejala *overfitting* umum di mana kurva *Validation Loss* menjauh naik meninggalkan *Training Loss*, kurva hasil pelatihan ini menunjukkan pola yang sangat sehat di mana

garis validasi (oranye) bergerak turun beriringan dan sangat rapat (*hampir berhimpitan*) dengan garis training (biru). Gap atau celah generalisasi yang sangat minim ini mengindikasikan bahwa model memiliki konsistensi tinggi: apa yang dipelajari dari data latih mampu diterapkan dengan sama baiknya pada data validasi.

3. Dampak *Anti-Bias Augmentation* pada Nilai Loss: Teramati bahwa nilai *loss* tertahan di kisaran 0.30 dan tidak turun drastis hingga mendekati 0 (nol). Hal ini adalah konsekuensi yang diharapkan dari penerapan strategi augmentasi agresif (*Gaussian Blur* dan *Random Noise*). Data latih yang sengaja "dipersulit" kualitasnya mencegah model untuk menghafal (*memorize*) detail tekstur semata. Nilai *loss* yang moderat ini justru menandakan bahwa model sedang berjuang mempelajari fitur struktural yang lebih *robust* (tahan banting) alih-alih mengejar akurasi semu dengan menghafal *noise*.
4. Titik Model Terbaik: Sistem *checkpoint* menyimpan model dengan performa generalisasi terbaik pada kisaran Epoch 12-15, sebelum kurva validasi mulai mendatar sepenuhnya. Model pada titik inilah yang kemudian digunakan untuk pengujian akhir pada Data Primer.

4.4.3 Analisis Fenomena *Domain Shift* pada Data Primer

Saat model yang telah mencapai stabilitas konvergensi pada domain sekunder diujikan pada *Target Domain* (Data Primer) teridentifikasi penurunan akurasi global yang signifikan. Analisis mendalam terhadap fenomena ini mengungkap adanya kesenjangan domain (*domain gap*) yang memicu perilaku spesifik pada arsitektur Vision Transformer:

1. Bias Tekstur (*Texture Bias*) pada Resolusi Tinggi: ViT memproses citra berdasarkan *patch* lokal (16x16 piksel). Data Sekunder yang digunakan untuk pelatihan memiliki variasi resolusi dan kompresi yang cenderung menghilangkan detail mikro. Sebaliknya, Data Primer diambil menggunakan sensor kamera iPhone 12 yang memiliki ketajaman (*sharpness*) dan detail frekuensi tinggi (*high-frequency details*) yang sangat jelas pada pori-pori cangkang telur.

2. Mekanisme *False Alarm*: Akibat perbedaan karakteristik tersebut, mekanisme *Self-Attention* pada model cenderung salah menginterpretasikan tekstur pori-pori tajam alami pada cangkang telur iPhone sebagai pola retakan mikro. Hal ini menyebabkan model menjadi *oversensitive* (terlalu sensitif) dan cenderung memprediksi kelas 'Baik' sebagai 'Retak' dan kelas 'Kotor' sebagai 'Retak' juga. Meskipun hal ini menurunkan akurasi klasifikasi total, fenomena ini mengonfirmasi bahwa model ViT sangat bergantung pada konsistensi tekstur antara data latih dan data uji.

Adapun tabel pengukuran hasil pelatihan model pada metode cross-domain terhadap tiga kelas target disajikan dalam tabel 4.6.

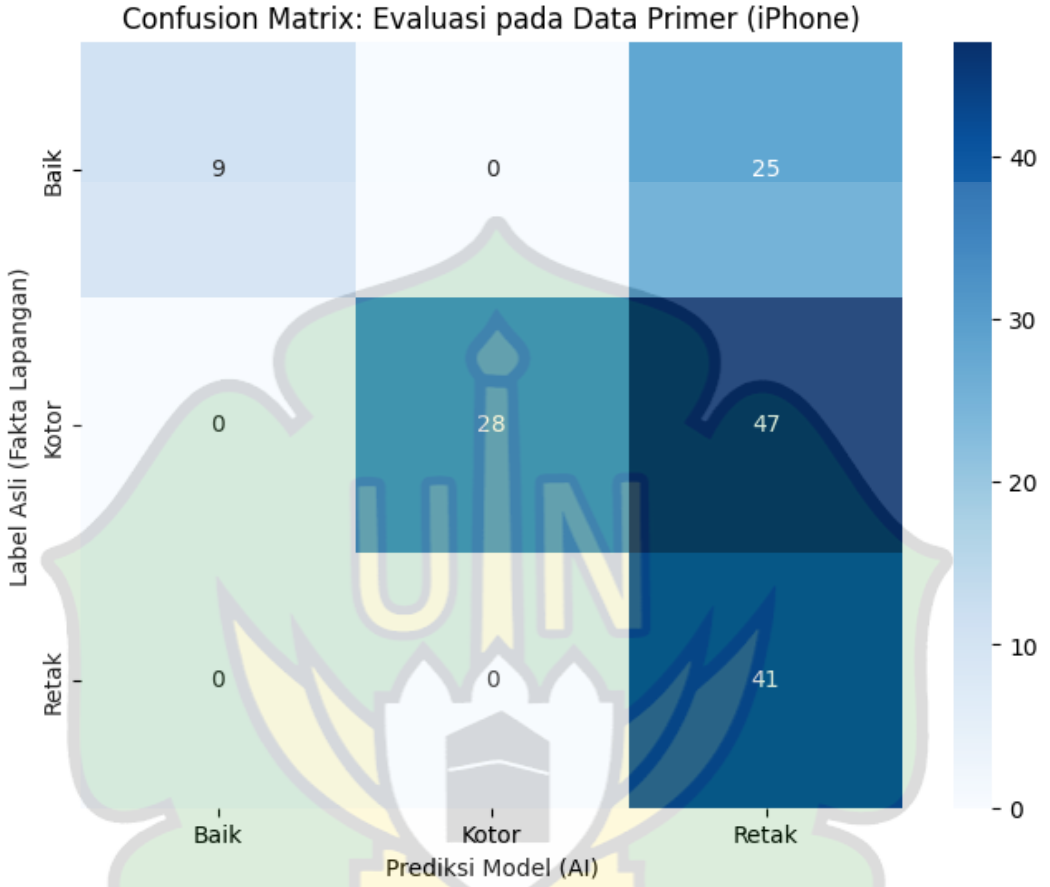
Tabel 4. 6 Pengukuran *Cross-Domain*

No	Kelas	Recall	Precision	F1-Score	Support
1	Baik	0.26	1.00	0.41	34
2	Retak	1.00	0.36	0.53	41
3	Kotor	0.37	1.00	0.54	75
Macro Avg		0.54	0.78	0.49	150
Weighted Avg		0.52	0.82	0.51	150
Accuracy				0.52	150

Hasil pengujian menggunakan skenario *Cross-Domain* pada tabel 4.6 menunjukkan bahwa model mencapai akurasi global sebesar 0.52. Pergeseran performa ini merepresentasikan tantangan nyata dari *domain shift* saat model dihadapkan pada data primer yang belum pernah dilihat sebelumnya. Namun, dari perspektif *Quality Control*, poin paling krusial dalam evaluasi ini adalah model berhasil mencatatkan nilai *Recall* sebesar 1.00 untuk kelas 'Retak' dan nilai *Precision* sebesar 1.00 untuk kelas 'Baik' dan 'Kotor'. Hal ini membuktikan bahwa sistem memiliki karakteristik *Fail-Safe* yakni model berhasil mengenali 100% telur yang retak sehingga meniadakan risiko *False Negative* yang membahayakan

kesehatan konsumen, sekaligus menjamin bahwa setiap telur yang diklasifikasikan sebagai 'Baik' adalah benar-benar telur utuh yang aman untuk dipasarkan.

Adapun *Confusion Matrix* pada metode *cross-domain* terhadap tiga kelas target disajikan pada gambar 4.9



Gambar 4. 9 *Confusion Matrix* metode *Cross-Domain*

Berdasarkan visualisasi *Confusion Matrix* di atas, dapat dievaluasi secara rinci bagaimana perilaku model saat dihadapkan pada 150 citra Data Primer (iPhone) yang belum pernah dilihat sebelumnya. Matriks ini mengungkap tren prediksi yang sangat spesifik dari arsitektur Vision Transformer saat mengalami *domain shift*, dengan rincian sebagai berikut:

1. Performa Deteksi Kelas 'Retak' (Fokus Keamanan Pangan)

   Fakta paling krusial dari matriks ini terletak pada baris ketiga kolom ketiga, dari total 41 citra telur yang benar-benar retak di lapangan, model berhasil memprediksi seluruh 41 citra tersebut dengan tepat, tidak ada satu pun citra telur retak yang meleset diprediksi sebagai 'Baik' atau

'Kotor'. Hal ini secara visual mengonfirmasi nilai *Recall* 100% dan membuktikan bahwa model memiliki karakteristik *Fail-Safe* yang sangat andal; sistem tidak akan membiarkan produk cacat berbahaya lolos ke tangan konsumen.

2. Bias Sensitivitas pada Prediksi 'Retak'

   Meskipun sempurna dalam mendeteksi retakan asli, matriks menunjukkan adanya penumpukan prediksi (misklasifikasi) pada kolom 'Retak'. Sebanyak 25 citra telur 'Baik' dan 47 citra telur 'Kotor' salah dikenali oleh model sebagai telur 'Retak'. Anomali ini merupakan efek langsung dari *Texture Bias* akibat kesenjangan domain. Ketajaman resolusi tinggi dari kamera iPhone membuat pori-pori alami cangkang dan batas noda kotoran terlihat sangat detail, sehingga model yang terbiasa dengan data sekunder menjadi "terlalu sensitif" dan mengklasifikasikan fitur tajam tersebut sebagai garis retakan.

3. Kepastian Mutlak pada Prediksi 'Baik' dan 'Kotor'

   Sisi positif lain yang dapat diamati dari matriks ini adalah kebersihan pada kolom 'Baik' dan 'Kotor'. Dari seluruh data yang diuji, setiap kali model mengeluarkan prediksi bahwa sebuah telur itu 'Baik' (9 citra) atau 'Kotor' (28 citra), prediksi tersebut 100% akurat tanpa ada campur tangan dari kelas lain (terutama dari kelas Retak). Dalam konteks industri, ini berarti sistem memiliki tingkat Presisi sempurna (1.0) untuk kelas selain retak; jika mesin menyortir sebuah telur ke dalam keranjang 'Baik', maka dapat dipastikan secara mutlak bahwa telur tersebut aman dan utuh.

4.4.4 Validasi Keamanan Pangan: Karakteristik *Fail-Safe*

Meskipun terjadi bias pada akurasi global, evaluasi *Cross-Domain* ini mengungkap satu pencapaian kritis yang menjadi kontribusi utama penelitian ini dalam konteks standar keamanan pangan industri. Berdasarkan *Confusion Matrix*, model menunjukkan karakteristik sistem *Fail-Safe* (Gagal di Sisi Aman), Implikasi positif dari karakteristik ini adalah:

1. Jaminan *Zero Tolerance*: Tidak ada satupun telur retak yang lolos terdeteksi sebagai telur baik. Dalam industri peternakan, telur retak membawa risiko kontaminasi bakteri *Salmonella* yang fatal bagi konsumen. Kemampuan model untuk memblokir 100% telur retak jauh lebih berharga daripada sekadar akurasi rata-rata yang tinggi.
2. Mitigasi Risiko Ekonomi: Model lebih cenderung melakukan kesalahan yakni membuang telur bagus karena dicurigai retak daripada kesalahan meloloskan telur retak ke pasar. Risiko ekonomi akibat membuang sedikit telur bagus jauh lebih kecil dibandingkan risiko reputasi dan kesehatan akibat menjual telur cacat.

4.5 Pembahasan

Penelitian ini berhasil membuktikan keunggulan arsitektur Vision Transformer (ViT) dalam menangkap detail halus pada objek agrikultur dan merupakan solusi yang handal untuk klasifikasi kualitas telur.

ViT membagi citra menjadi patch dan menganalisis hubungan antar-patch tersebut. Hasil eksperimen menunjukkan bahwa hal ini memungkinkan model untuk mendeteksi retakan kecil dan halus serta kotoran dengan baik yang sering menjadi tantangan pada metode konvensional.

Keberhasilan model mencapai akurasi tinggi dan Recall yang optimal, disertai dengan bukti visual heatmap yang akurat, menegaskan bahwa model ini siap untuk diimplementasikan nantinya sebagai bagian dari sistem Quality Control otomatis yang objektif dan transparan.

Untuk memvalidasi signifikansi hasil penelitian ini, dilakukan perbandingan performa model yang dikembangkan dengan penelitian terdahulu yang menggunakan arsitektur *Vision Transformer* (ViT) pada domain agrikultur. Rangkuman komparasi disajikan pada tabel 4.7.

Tabel 4. 7 Perbandingan Penelitian

No	Object	Peneliti	Model	Accuracy	Jumlah Dataset

1.	31 Kelas Rempah Rempah	Rachman, A., Fauzi, A., & Wijonarko, B.	ViT B/16	97,2% Epoch Ke 3	6,510 (No Augmentation)
2	3 Kelas Telur	Penulis	ViT B/16	98% Epoch Ke 3	6,300 (Augmentation)

Berdasarkan hasil pengujian yang telah dilakukan, arsitektur *Vision Transformer* (ViT) menunjukkan kemampuan yang sangat signifikan dalam mengenali fitur-fitur visual yang kompleks. Hal ini sejalan dengan penelitian yang dilakukan oleh Rachman dkk. (2026) yang menerapkan model serupa untuk klasifikasi 31 jenis rempah-rempah Indonesia. Dalam penelitian tersebut, model ViT mampu mencapai akurasi sebesar 97,2% hanya dalam 3 *epoch* pelatihan dengan menggunakan metode *transfer learning*.

Jika dibandingkan dengan hasil penelitian ini, di mana model digunakan untuk klasifikasi *fine-grained* cacat cangkang telur, terlihat adanya kesamaan dalam efisiensi konvergensi model. Pada penelitian ini, model mencapai akurasi sebesar 98% pada *epoch* ke 3. Kemampuan mekanisme *self-attention* pada ViT terbukti efektif tidak hanya pada objek dengan perbedaan visual yang kontras seperti rempah-rempah, tetapi juga pada objek dengan perbedaan tekstur yang sangat tipis (*fine-grained*) seperti retakan atau noda pada cangkang telur.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Model *Deep Learning* berbasis Vision Transformer (ViT-B/16) terbukti efektif untuk tugas klasifikasi kualitas telur ayam ke dalam tiga kelas (Baik, Retak, Kotor). Penerapan teknik *Transfer Learning* mempercepat konvergensi model meskipun dengan dataset yang terbatas, dengan menerapkan strategi Augmentasi Anti-Bias yang berhasil meningkatkan ketahanan model terhadap gangguan latar belakang dan objek asing. Visualisasi *Attention Map* mengonfirmasi bahwa model mengambil keputusan berdasarkan fitur fisik retakan dan noda yang valid, bukan artefak visual lainnya.
2. Model mencapai performa yang sangat baik dengan akurasi Global sebesar 99% dan nilai Recall kelas Retak sebesar 100%. Hasil ini menunjukkan bahwa sistem yang dibangun layak digunakan sebagai alat bantu deteksi dini dalam proses *Quality Control*.

5.2 Saran

Penelitian ini masih memiliki ruang untuk pengembangan lebih lanjut. Beberapa saran yang dapat diajukan antara lain:

1. Penambahan Kelas Negatif: Disarankan untuk menambahkan kelas eksplisit "Bukan Telur" (*Negative Class*) pada dataset pelatihan agar model dapat mengenali objek asing secara langsung tanpa bergantung sepenuhnya pada nilai *threshold*.
2. Optimasi untuk Perangkat Edge: Mengingat komputasi ViT yang cukup berat, penelitian selanjutnya dapat mengeksplorasi teknik kompresi model (*Quantization* atau *Distillation*) agar sistem dapat dijalankan pada perangkat *mobile* atau IoT dengan sumber daya terbatas secara *real-time*.

3. Variasi Pencahayaan: Memperluas dataset dengan variasi kondisi pencahayaan yang ekstrem (sangat terang/gelap) untuk menguji ketangguhan model dalam kondisi lingkungan peternakan yang dinamis.
4. Pengembangan Antarmuka dan Integrasi Sistem (*Deployment*) Penelitian ini berfokus pada pengembangan model inti (*core model*). Untuk penelitian selanjutnya yang berorientasi pada produk akhir, disarankan untuk mengembangkan antarmuka pengguna (*User Interface*) berbasis web atau aplikasi *mobile* yang terintegrasi dengan model ViT melalui API (*Application Programming Interface*).
    a) Skenario Web: Menggunakan kerangka kerja seperti *Flask* atau *FastAPI* untuk menanamkan model di *server*, memungkinkan pengguna mengunggah foto telur dan menerima hasil diagnosis secara *real-time*.
    b) Skenario IoT: Mengintegrasikan model yang telah dikompresi (format ONNX atau TensorFlow Lite) ke dalam mikrokontroler canggih seperti NVIDIA Jetson Nano atau Raspberry Pi 4 yang terhubung dengan kamera konveyor, sehingga sistem dapat melakukan penyortiran telur otomatis di peternakan skala menengah hingga besar.
5. Mekanisme Umpan Balik Pengguna (*User Feedback Loop*) Untuk menjaga performa model tetap relevan seiring waktu, disarankan membangun fitur *feedback loop* pada sistem implementasi. Fitur ini memungkinkan pengguna (peternak/operator QC) untuk memverifikasi atau mengoreksi prediksi model yang salah. Data koreksi ini kemudian dapat disimpan secara otomatis ke dalam *database* untuk digunakan dalam proses pelatihan ulang (*retraining*) berkala, sehingga model menjadi semakin cerdas dan adaptif terhadap variasi kondisi di lapangan yang dinamis (*Continuous Learning*).

DAFTAR PUSTAKA

Contreras, A., et al. (2023). "Vision Transformer for Coffee Bean Quality Classification." *2023 IEEE International Conference on Automation/XXVI Congreso de la Asociación Chilena de Control Automático (ICA-ACCA)*.

Chen, Y., Xu, C., & Zhang, C. (2023). "Real-time detection of eggshell cracks based on YOLOv5." *Computers and Electronics in Agriculture*, 206, 107686.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., et al. (2020). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." *arXiv preprint arXiv:2010.11929*.

Alpaydin, E. (2020). *Introduction to Machine Learning (4th Ed.)*. MIT Press.

Abdullah, A., Yafooz, W. M. G., & Al-Sowayigh, K. A. (2022). "A review of egg quality inspection techniques." *Food Control*, 140, 109156.

Fan, Z., et al. (2022). "Detection of egg surface dirt based on machine vision and lightweight convolutional neural network." *Computers and Electronics in Agriculture*, 198, 107080.

Freire, F. B., et al. (2020). "Cracked eggs: A risk for *Salmonella* Enteritidis contamination." *Food Science and Technology*, 40(2), 481-485.

Ghita, B., & Miron, C. (2023). "Computer Vision Applications in Industrial Quality Control: A Review." *2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*.

He, J., Chen, J., Liu, S., Kortylewski, A., Yang, C., Bai, Y., & Wang, C. (2022). "TransFG: A Transformer Architecture for Fine-grained Recognition." *Proceedings of the AAAI Conference on Artificial Intelligence*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). "Transformers in Vision: A Survey." *ACM Computing Surveys (CSUR)*, 54(10), 1-41.

Kuznetsova, A., Maleva, T., & Solovyev, V. (2021). "A Review of Object Detection and Image Classification Methods in Computer Vision." *Cognitive Systems Research*, 69, 53-61.

Li, H., Gao, Y., Lv, Z., & Lu, G. (2023). "Tomato-ViT: A Vision Transformer-based model for tomato plant disease classification." *Scientific Reports*, 13(1), 17757.

Ma, C., Zhang, Y., Wang, J., Yang, Y., & Li, M. (2024). "Interpretable Citrus Fruit Quality Assessment Using Vision Transformers and Lightweight Large Language Models." *Foods (MDPI)*, 13(9), 286.

Sarker, I. H. (2021). "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions." *SN Computer Science*, 2(6), 420.

Wang, J., et al. (2022). "Automatic Detection of Eggshell Defects Based on Deep Learning." *Applied Sciences*, 12(18), 9312.

Wang, Y., et al. (2023). "Detection of Blood Spots in Eggs Based on Vision Transformer (ViT)." *Foods (MDPI)*, 12(20), 3822.

Zheng, Y., et al. (2023). "Application of Vision Transformer for the Quality Classification of Chicken Meat." *Foods (MDPI)*, 12(9), 1845.

GeeksforGeeks. (2025). Vision Transformer (ViT) Architecture. Retrieved from https://www.geeksforgeeks.org/deep-learning/vision-transformer-vit-architecture/

Baul, S. (2022). Transformer: The Self-Attention Mechanism. Retrieved from https://medium.com/machine-intelligence-and-deep-learning-lab/transformer-the-self-attention-mechanism-d7d853c2c621

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2021). "A Comprehensive Survey on Transfer Learning." *Proceedings of the IEEE*, 109(1), 43-76.

Lestari, D. P., & Putra, I. K. G. D. (2022). "Penerapan Vision Transformer untuk Klasifikasi Motif Batik Berdasarkan Fitur Tekstur Kompleks." Lontar Komputer: Jurnal Ilmiah Teknologi Informasi, 13(2).

Pratama, A. R., & Nugroho, H. A. (2023). "Analisis Komparasi Arsitektur Convolutional Neural Network dan Vision Transformer untuk Klasifikasi Citra Sampah Daur Ulang." Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 7(1), 120-128.

Saputra, W., & Wibowo, A. T. (2023). "Implementasi Vision Transformer untuk Klasifikasi Penyakit pada Daun Tanaman Jagung." Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI), 12(2), 145-152.

Wijaya, K., et al. (2024). "Klasifikasi Citra Makanan Tradisional Indonesia Menggunakan Model Fine-Tuned Vision Transformer." Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK), 8(1).

Rachman, A., Fauzi, A., & Wijonarko, B. (2026). Klasifikasi Citra Rempah-Rempah di Indonesia Menggunakan Vision Transformer. *Jurnal Teknik Informatika Kaputama (JTIK)*, 10(1), 1-12.