PENGEMBANGAN APLIKASI BERBASIS *WEB* UNTUK DETEKSI KUALITAS TELUR AYAM MENGGUNAKAN ALGORITMA *YOLOV5*

1st M.Irgy Syah.Daffa
Direktorat Kampus Surabaya
Universitas Telkom Surabaya
Surabaya
irgydaffa@
student.telkomuniversity.ac.id

2nd Dewi Rahmawati
Direktorat Kampus Surabaya
Universitas Telkom Surabaya
Surabaya
dewirahmawati@telkomuniversity.ac.id

3rd Achmad Muzakki
Direktorat Kampus Surabaya
Universitas Telkom Surabaya
Surabaya
achmadmuzakki@
telkomuniversity.ac.id

***Abstrak* — *Rooftop Farming Center* (*RFC*) Universitas Telkom Surabaya merupakan inovasi agrikultur urban yang menghadapi tantangan dalam standardisasi kualitas telur, sebuah faktor krusial untuk pendapatan dan kepuasan pelanggan. Penelitian ini bertujuan mengembangkan aplikasi berbasis *web* untuk deteksi kualitas telur ayam secara otomatis menggunakan algoritma *YOLOv5*. Metode pengembangan sistem yang digunakan adalah *Waterfall*, dengan arsitektur yang memisahkan antarmuka *web* (*Laravel*) dan *API* pemrosesan model (*Flask*). *Dataset* citra telur yang dikumpulkan akan melalui proses *preprocessing* dan augmentasi data digunakan untuk melatih tiga varian model: *YOLOv5s*, *YOLOv5m*, dan *YOLOv5l*. Hasil evaluasi komparatif menunjukkan *YOLOv5m* sebagai model optimal dengan *mean Average Precision* (mAP) pada *Intersection over Union* (*IoU)* 0.5 sebesar 0.95, *recall* 0.94, dan *precision* 0.91. Pengujian fungsionalitas sistem melalui *black-box testing* menunjukkan bahwa seluruh fitur aplikasi berjalan sesuai spesifikasi. Penelitian ini membuktikan bahwa *YOLOv5* dapat diimplementasikan secara fungsional dalam sistem *web* untuk meningkatkan objektivitas sortir kualitas telur.**

Kata kunci*— deteksi kualitas telur, *yolov5*, *deep learning*, aplikasi *web*, *waterfall

I. PENDAHULUAN

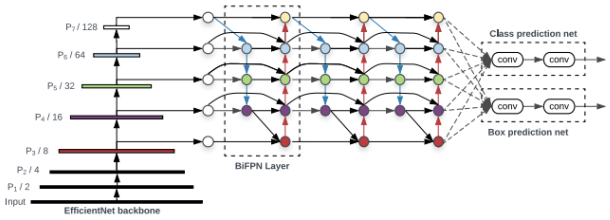
Telur merupakan kebutuhan pangan hewani yang banyak dikonsumsi masyarakat karena mudah diperoleh dan bergizi tinggi. Kandungan seperti *protein*, *vitamin D*, dan *B12* menjadikan telur sebagai sumber nutrisi penting. Telur juga fleksibel dalam pengolahan menjadi berbagai makanan seperti kue dan masakan [1]. Menurut Pusat Data dan Sistem Informasi Pertanian (PUSDATIN), produksi telur di Indonesia tahun 2024 mencapai 5,70 juta ton. Produksi ini meningkat seiring populasi ayam petelur dan pengelolaan yang lebih baik. Jawa Timur menjadi penyumbang terbesar produksi nasional dengan kontribusi 32% [2].

Transformasi digital mendorong adopsi kecerdasan buatan dalam sektor peternakan. Salah satu proses penting adalah penilaian mutu telur, yang secara manual sering tidak konsisten, subjektif, dan lambat.Untuk mengatasi hal tersebut, teknologi deteksi objek berbasis deep learning menjadi solusi. Algoritma *You Only Look Once (YOLO)*, khususnya versi *YOLOv5*, banyak digunakan untuk deteksi objek secara *real-time*, cocok untuk aplikasi berbasis *web*.

Penelitian ini dilakukan di *Rooftop Farming Center* Universitas Telkom Surabaya. Unit ini membutuhkan sistem klasifikasi mutu telur untuk sortasi dan pelaporan produksi. Aplikasi *web* dikembangkan untuk mengintegrasikan model deteksi mutu telur dengan algoritma *YOLOv5*. Pengembangan aplikasi menggunakan metode *Waterfall* karena kebutuhan sistem sudah jelas. Arsitektur sistem terdiri dari *backend Flask* untuk pemrosesan model, dan frontend Laravel untuk antarmuka pengguna. Sistem juga mencakup *preprocessing* gambar, unggah gambar, serta visualisasi hasil deteksi.

II. KAJIAN TEORI

A. Algoritma *YOLO (You Only Look Once)*



Gambar 1 Arsitektur *YOLOv5*

YOLO adalah algoritma deteksi objek berbasis *deep learning* yang memproses gambar secara keseluruhan dalam satu kali inferensi, memungkinkannya mencapai kecepatan *real-time* . *YOLOv5*, salah satu varian yang paling stabil dan efisien, memiliki arsitektur yang terdiri dari tiga komponen utama: *Backbone* (CSPDarknet53) untuk ekstraksi fitur, *Neck* (PANet) untuk agregasi fitur dari berbagai skala, dan *Head* untuk melakukan prediksi akhir berupa *bounding box*, skor keyakinan, dan klasifikasi objek [3].

B. *SDLC Waterfall*

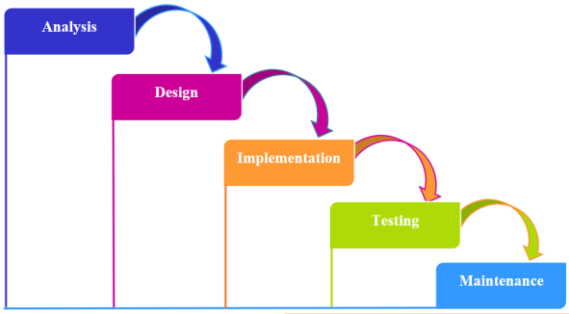
Metodologi *Waterfall* adalah model pengembangan perangkat lunak yang bersifat sekuensial dan terstruktur. Prosesnya mengalir melalui fase-fase yang berbeda secara berurutan: *Requirement Analysis*, *Design*, *Implementation*, *Testing*, dan *Maintenance*. Setiap fase harus diselesaikan sepenuhnya sebelum melanjutkan ke fase berikutnya, menjadikannya pendekatan yang cocok untuk proyek dengan ruang lingkup yang jelas dan stabil [4].

C. Penelitian Terdahulu

Penelitian terdahulu telah banyak mengeksplorasi penggunaan metode *deep learning* untuk deteksi kualitas produk agrikultur dan penerapan metodologi pengembangan sistem yang relevan. Sebuah penelitian yang mendeteksi citra buah-buahan menggunakan algoritma *YOLO* berhasil mencapai nilai *mAP* sebesar 91% [5], menunjukkan efektivitas *YOLO* untuk objek agrikultur . Dalam konteks

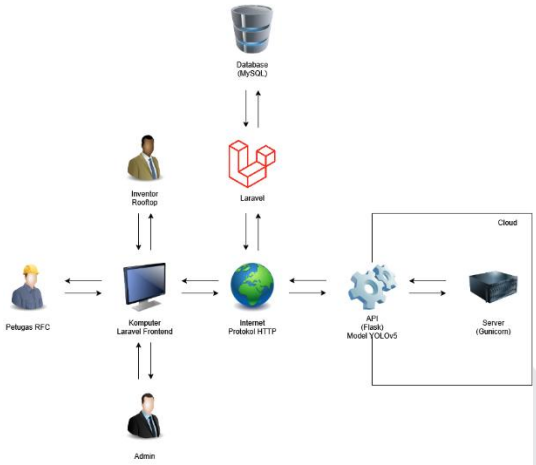
deteksi kualitas telur, metode klasik seperti *GLCM* dan *KNN* mampu mencapai akurasi 82,35% [6], sementara pendekatan yang lebih modern menggunakan CNN untuk deteksi kesegaran telur berdasarkan citra cangkang mencapai akurasi 90%[7] . Dari sisi pengembangan sistem, metodologi .

III. METODE



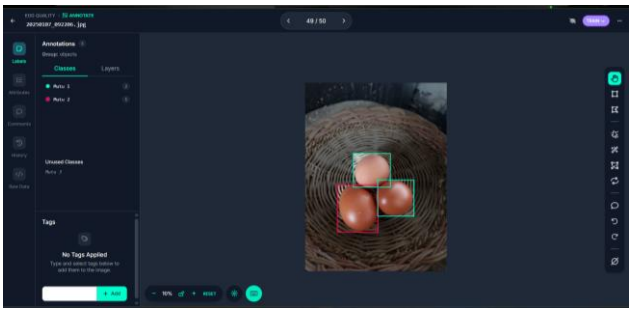
Gambar 2 *SDLC Waterfall*

Penelitian ini menerapkan metodologi pengembangan sistem *Waterfall* yang terdiri dari lima tahapan sekuensial. Proses dimulai dengan analisis kebutuhan, di mana kebutuhan fungsional dan non-fungsional sistem didefinisikan, termasuk spesifikasi dataset. Tahap desain mencakup perancangan arsitektur sistem dan basis data (*MySQL*).



Gambar 3 Arsitektur Sistem

Arsitektur sistem yang dirancang menggambarkan alur kerja yang terstruktur, melibatkan interaksi antara pengguna, aplikasi *web*, *API*, dan basis data, seperti yang diilustrasikan pada Gambar 1. Pengguna (Petugas RFC, Inventor Rooftop, dan Admin) berinteraksi dengan antarmuka *web* yang dibangun menggunakan *Laravel*. Ketika seorang petugas mengunggah gambar telur, permintaan dikirim melalui protokol *HTTP* ke *API* yang dikembangkan dengan *Flask*. *API* ini berfungsi sebagai jembatan yang memuat model *YOLOv5* untuk melakukan inferensi. Hasil deteksi kemudian disimpan atau diambil dari basis data *MySQL* melalui *server Gunicorn*, yang mengelola komunikasi antara *API* dan basis data.

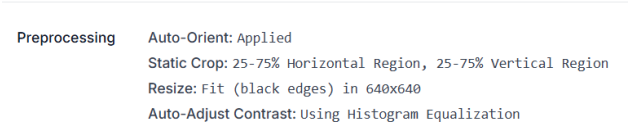


Gambar 4 Pelabelan Dataset

Proses akuisisi dan preparasi dataset menjadi inti dari tahap implementasi. *Dataset* awal terdiri dari 432 gambar telur ayam yang dikumpulkan dari RFC Telkom University Surabaya dan sumber daring, yang kemudian dianotasi secara manual ke dalam tiga kelas: Mutu 1 (bersih, tanpa cacat), Mutu 2 (sedikit noda), dan Mutu 3 (noda berat atau pecah). total objek telur yang berhasil diberi label terdistribusi seperti pada Tabel 1.

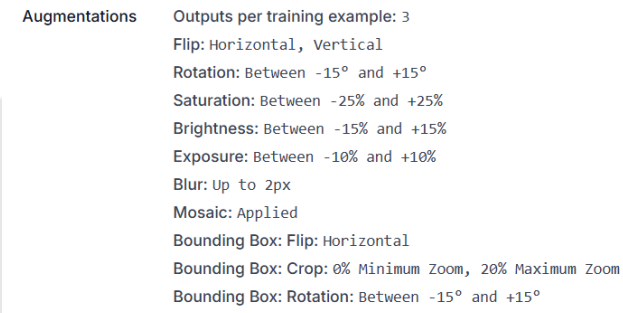
Tabel 1 Distribusi Objek Telur Berdasarkan Kelas

Kelas	Jumlah Objek
Mutu 1	408
Mutu 2	189
Mutu 3	307



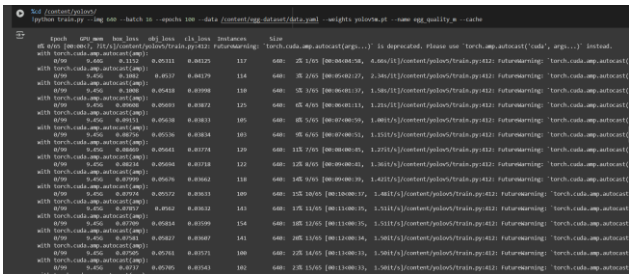
Gambar 5 Teknik *Preprocessing*

Untuk standardisasi dan peningkatan kualitas, *dataset* melalui serangkaian langkah *preprocessing* yang mencakup *Auto-Orient, Static Crop, Resize* ke resolusi 640×640 *piksel*, dan penyesuaian kontras.



Gambar 6 Teknik *Augmentation*

Selanjutnya, teknik augmentasi data seperti *Flip, Rotation (±15∘),* penyesuaian *Saturation dan Brightness,* serta *Mosaic Augmentation* diterapkan untuk memperluas *dataset* menjadi 1.124 gambar. *Dataset* ini kemudian dibagi menjadi 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian.



Gambar 7 Lingkungan *Training*

Tiga varian model—*YOLOv5s, YOLOv5m*, dan *YOLOv5l*—dilatih di lingkungan *Google Colaboratory* selama 100 *epoch* dengan ukuran *batch* 16 dan *optimizer SGD* untuk perbandingan kinerja.

Tahap pengujian melibatkan dua aspek. Pertama, evaluasi kinerja kuantitatif model menggunakan metrik *Precision, Recall,* dan *mean Average Precision (mAP)* pada ambang batas *Intersection over Union (IoU)* 0.5. Kedua, validasi fungsionalitas aplikasi *web* secara keseluruhan menggunakan metode *black-box testing* yang mencakup 20 skenario pengujian untuk memastikan semua fitur berjalan sesuai spesifikasi.

IV. HASIL DAN PEMBAHASAN

Evaluasi komparatif terhadap tiga varian model *YOLOv5* menunjukkan perbedaan kinerja yang signifikan, seperti yang dirangkum pada Tabel 2.

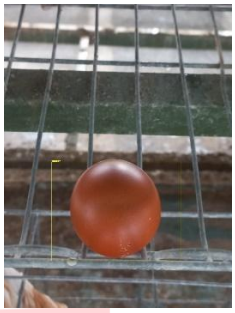
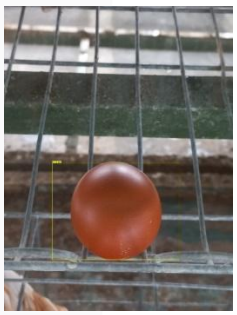
Tabel 2 Hasil Evaluasi *Training YOLOv5*

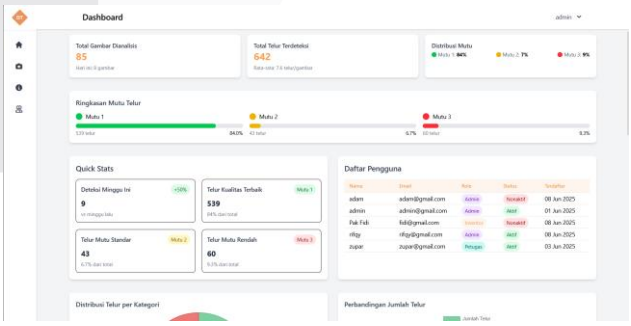
Metrik	Yolov5s	Yolov5m	Yolov5l
mAP@0.5	0.947	0.952	0.951
Precision	0.944	0.910	0.914
Recall	0.893	0.943	0.903

Berdasarkan data di atas, *YOLOv5m* dipilih sebagai model optimal untuk diimplementasikan dalam sistem. Meskipun *YOLOv5l* menunjukkan *mAP* pada rentang *IoU* yang lebih tinggi (*mAP*@0.5:0.95 sebesar 0.751), model ini menunjukkan penurunan *recall* yang tajam pada tingkat keyakinan tinggi, menandakan model menjadi "rapuh" dan cenderung melewatkan deteksi valid. Di sisi lain, *YOLOv5s*, meskipun memiliki *precision* tertinggi, mencatatkan *recall* terendah dan menunjukkan fluktuasi prediksi yang signifikan, terutama untuk kelas Mutu 1. *YOLOv5m* menawarkan keseimbangan terbaik, dengan *mAP*@0.5 tertinggi dan *recall* yang jauh lebih unggul (0.943), yang krusial untuk meminimalkan kesalahan *false negative* dalam aplikasi praktis. Stabilitas prediksi dan efisiensi komputasi yang lebih baik dibandingkan *YOLOv5l* menjadikan *YOLOv5m* pilihan yang paling pragmatis dan kuat.

Untuk memberikan gambaran visual mengenai kemampuan dan perbedaan deteksi dari setiap model, disajikan perbandingan hasil deteksi yang dilakukan oleh setiap varian pada citra uji yang sama pada Tabel 3.

Tabel 3 Hasil Deteksi dari varian Model *YOLOv5*

Model	Hasil Deteksi	Keterangan
YOLOv5s		Telur terdeteksi sebagai Mutu 2 dengan *confidence* 84%
YOLOv5m		Telur terdeteksi sebagai Mutu 2 dengan *confidence* 94%. Terdapat deteksi salah pada latar belakang sebagai Mutu 1 (*confidence* 68%).
YOLOv5l		Telur terdeteksi sebagai Mutu 2 dengan *confidence* 95%.



Gambar 8 Tampilan *Dashboard*

Aplikasi *web* yang dikembangkan menyediakan antarmuka yang intuitif bagi pengguna untuk berinteraksi dengan sistem deteksi. Implementasi UI mencakup beberapa halaman utama yang dirancang untuk memenuhi kebutuhan fungsional setiap peran pengguna:

- **Halaman *Login*:** Sebagai gerbang masuk, halaman ini memvalidasi kredensial pengguna (email dan password) dan mengarahkan mereka ke dasbor sesuai dengan perannya (Admin, Petugas, atau Inventor).

- ***Dashboard*:** Menampilkan ringkasan visual data, termasuk grafik jumlah telur berdasarkan mutu dan tabel riwayat deteksi terbaru, memungkinkan pemantauan tren kualitas secara cepat.

- **Halaman Deteksi:** Fitur inti di mana pengguna dapat mengunggah gambar telur. Gambar tersebut kemudian diproses oleh *API Flask*, dan hasilnya ditampilkan kembali kepada pengguna, lengkap dengan *bounding box* dan label mutu yang diprediksi, seperti yang ditunjukkan pada Gambar 2.

- **Halaman Laporan:** Memungkinkan Admin dan Inventor untuk menghasilkan laporan hasil panen yang dapat difilter berdasarkan tanggal, mutu, dan pengguna, serta dapat dicetak dalam format PDF.

- **Manajemen Pengguna:** Halaman khusus Admin untuk mengelola akun pengguna, termasuk menambah, mengubah, dan menonaktifkan akun.

Validasi fungsional sistem dilakukan melalui *black-box testing* terhadap 20 skenario yang mencakup semua alur kerja utama, mulai dari otentikasi pengguna, unggah gambar, proses deteksi, hingga pembuatan laporan. Hasil pengujian menunjukkan bahwa semua 20 *test case* berhasil dieksekusi dengan status "Sukses". Hal ini mengonfirmasi bahwa integrasi antara antarmuka *web Laravel, API Flask*, dan model *YOLOv5m* berjalan dengan baik dan semua fungsionalitas sistem telah memenuhi spesifikasi yang ditentukan.

V. KESIMPULAN

Penelitian ini telah berhasil merancang, mengembangkan, dan memvalidasi sebuah sistem aplikasi berbasis *web* untuk deteksi kualitas telur ayam secara otomatis. Algoritma *YOLOv5* terbukti baik untuk tugas ini, di mana model *YOLOv5m* terpilih sebagai arsitektur optimal yang memberikan keseimbangan terbaik antara akurasi (*mAP@*0.5 sebesar 0.95), *recall* (0.94), dan efisiensi komputasi. Sistem yang dibangun menggunakan metodologi *Waterfall* dengan arsitektur terpisah *(Laravel dan Flask)* telah berhasil divalidasi melalui pengujian *black-box testing*, yang menunjukkan bahwa seluruh fungsionalitas berjalan sesuai dengan spesifikasi. Hasil penelitian ini tidak hanya menghasilkan sebuah model *machine learning* yang dapat mendeteksi kualitas telur, tetapi juga sebuah solusi *end-to-end* yang fungsional dan siap pakai, memberikan kontribusi nyata untuk meningkatkan objektivitas dalam kontrol kualitas di lingkungan pertanian urban seperti *Rooftop Farming Center*.

REFERENSI

[1] K. R. gharizah Alhuur, A. Pratama, and E. Yuniarti, "Kualitas dan Cara Penyimpanan Telur Yang Baik dalam Upaya Menjaga Asupan Gizi Optimal di Masa Pandemi COVID-19," *Farmers: Journal of Community Services*, vol. 1, no. 1, p. 24, Aug. 2020, doi: 10.24198/fjcs.v1i1.28647.

[2] S. Jenderal -Kementerian, P. Sekretariat, and J.-K. Pertanian, "Pusat Data dan Sistem Informasi Pertanian."

[3] R. Hussein Hasan, R. Majid Hassoo, and I. Salman Aboud, "Yolo Versions Architecture: Review," *International Journal of Advances in Scientific Research and Engineering*, vol. 09, no. 11, pp. 73–92, 2023, doi: 10.31695/ijasre.2023.9.11.7.

[4] U. S. Senarath, "Waterfall Methodology, Prototyping and Agile Development," 2021, doi: 10.13140/RG.2.2.17918.72001.

[5] B. I. J. Dede Haris Saputra, "OBJECT DETECTION UNTUK MENDETEKSI CITRA BUAH BUAHAN MENGGUNAKAN METODE YOLO," 2023.

[6] R. Kaamtsaalil Salsabiilaa, B. Hidayat, and S. Darana, "Deteksi Kualitas dan Kesegaran Telur Ayam Ras Berdasarkan Deteksi Objek Transparan dengan Metode Grey Level Co-Occurance Matrix (GLCM) dan Klasifikasi K-NEAREST Neighbor (KNN) [Rizka Kaamtsaalil Salsabiilaa] DETEKSI KUALITAS DAN KESEGARAN TELUR AYAM RAS BERDASARKAN DETEKSI OBJEK TRANSPARAN DENGAN METODE GREY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN KLASIFIKASI K-NEAREST NEIGHBOR (KNN)," 2016.

[7] S. Maulana, "Deteksi Kesegaran Telur Ayam pada Citra Cangkang Menggunakan Metode Convolutional Neural Network (CNN)," 2024.