

Implementasi Deep Learning Berbasis YOLOv5 Untuk Identifikasi Kucing Individu pada Pet Feeder

1st Muhammad Rayyan Aqilah Manna
 Fakultas Teknik Elektro
 Universitas Telkom
 Bandung, Indonesia
 rayyanaqilah@student.telkomuniversity.ac.id

2nd Meta Kallista
 Fakultas Teknik Elektro
 Universitas Telkom
 Bandung, Indonesia
 metakallista@telkomuniversity.ac.id

3rd Rifqi Muhammad Fikri
 Fakultas Teknik Elektro
 Universitas Telkom
 Bandung, Indonesia
 rifqmff@telkomuniversity.ac.id

Abstrak — Penelitian ini mengembangkan dan mengimplementasikan model deep learning pada sistem Smart Pet Feeder untuk mengenali kucing peliharaan individu berdasarkan foto yang diunggah pengguna melalui aplikasi. Teknologi You Only Look Once (YOLO) dipilih karena memiliki kecepatan deteksi tinggi dan akurasi yang memadai untuk pengenalan secara real-time. Dataset dibuat secara khusus dari foto-foto kucing milik pengguna dengan berbagai sudut dan kondisi pencahayaan guna meningkatkan kemampuan generalisasi model. Proses pelatihan dilakukan dengan pengaturan hyperparameter yang divariasikan, kemudian dievaluasi menggunakan metrik mean Average Precision (mAP) dan confusion matrix untuk mengukur performa prediksi. Hasil pengujian menunjukkan model mampu mengenali kucing dengan akurasi tinggi pada kondisi pencahayaan terang, namun mengalami penurunan performa pada kondisi minim cahaya. Tingkat keakurasaian model pada alat sangat bergantung pada kualitas, sudut, dan pencahayaan foto yang diunggah oleh pengguna. Integrasi model ke dalam Smart Pet Feeder memungkinkan sistem mengatur akses makan hanya untuk kucing yang terdaftar, sehingga meningkatkan keamanan, mengurangi risiko makanan diakses kucing lain, dan membantu pemilik memantau aktivitas makan hewan peliharaan secara efektif.

Kata kunci— computer vision, deep learning, deteksi real-time, identifikasi kucing, smart pet feeder, YOLO

I. PENDAHULUAN

Kucing merupakan salah satu hewan peliharaan yang populer di Indonesia karena sifatnya yang mandiri, bersahabat, dan mampu memberikan dukungan emosional kepada pemiliknya. Meskipun demikian, pemeliharaan kucing memerlukan perhatian khusus, terutama dalam hal pemberian makan yang teratur dan sesuai takaran. Pemilik yang memiliki kesibukan tinggi sering kali menghadapi kesulitan dalam memastikan kucingnya mendapatkan makanan pada waktu yang tepat. Ketidakteraturan dalam pemberian pakan tidak hanya berdampak pada kesehatan fisik kucing, tetapi juga dapat memicu perubahan perilaku yang berpotensi menurunkan kesejahteraan hewan.

Perkembangan teknologi *Internet of Things* (IoT) dan *computer vision* telah membuka peluang untuk menciptakan solusi otomatis dalam pengelolaan pakan hewan. Berbagai produk *smart pet feeder* telah tersedia di pasaran, sebagian dilengkapi dengan kamera dan sistem penjadwalan, namun masih jarang yang menerapkan teknologi pengenalan kucing secara individual. Penelitian-penelitian terdahulu umumnya berfokus pada deteksi ras atau pengenalan wajah hewan menggunakan dataset umum, sedangkan pendekatan untuk identifikasi kucing spesifik milik pengguna dengan basis foto pribadi masih terbatas.

Permasalahan utama yang ingin diatasi adalah bagaimana memastikan hanya kucing terdaftar yang dapat mengakses pakan, sehingga mengurangi risiko makanan dikonsumsi kucing lain, sekaligus memberikan keamanan dan kontrol lebih bagi pemilik. Untuk itu, penelitian ini mengusulkan penerapan model *deep learning* berbasis algoritma *You Only Look Once* (YOLO) yang dilatih menggunakan foto kucing peliharaan yang diunggah oleh pemilik melalui aplikasi. Model ini diintegrasikan ke dalam sistem Smart Pet Feeder, sehingga mampu melakukan deteksi secara *real-time* dan mengatur pemberian pakan secara otomatis. Tujuan dari penelitian ini adalah menghasilkan sistem yang andal dalam mengidentifikasi kucing individu, meningkatkan efisiensi pemberian pakan, serta memberikan pengalaman pengelolaan hewan peliharaan yang lebih praktis dan aman bagi pemilik.

II. KAJIAN TEORI

A. Deep Learning

Deep learning adalah cabang dari machine learning yang menggunakan arsitektur jaringan saraf tiruan (artificial neural networks) dengan banyak lapisan tersembunyi untuk mengekstraksi fitur dari data secara otomatis. Teknik ini sangat efektif untuk pengolahan data visual karena mampu mempelajari representasi hierarkis dari citra, mulai dari fitur sederhana seperti tepi dan bentuk, hingga pola kompleks yang spesifik terhadap objek tertentu. Dalam konteks penelitian ini, deep learning digunakan untuk melatih model yang dapat mengidentifikasi kucing peliharaan individu berdasarkan foto yang diunggah oleh pengguna.

B. YOLO (You Only Look Once)

YOLO merupakan algoritma deteksi objek yang memproses gambar secara end-to-end dalam satu tahap, sehingga memiliki kecepatan tinggi tanpa mengorbankan akurasi secara signifikan. Berbeda dengan metode deteksi dua tahap seperti R-CNN yang memisahkan proses ekstraksi wilayah kandidat dan klasifikasi, YOLO langsung membagi gambar menjadi grid dan memprediksi bounding box serta kelas objek secara bersamaan. Keunggulan ini menjadikannya cocok untuk aplikasi real-time seperti Smart Pet Feeder.

C. Computer Vision

Computer vision adalah bidang ilmu yang berfokus pada pengembangan metode untuk memungkinkan komputer memahami dan menginterpretasikan informasi visual dari dunia nyata. Dengan bantuan deep learning, sistem dapat mengenali, melacak, dan mengklasifikasikan objek dalam gambar atau video. Dalam penelitian ini, computer vision berperan dalam memproses citra yang diambil oleh kamera Smart Pet Feeder untuk mengidentifikasi kucing yang terdaftar.

D. Internet of Things (IoT) untuk Smart Pet Feeder

IoT adalah konsep di mana perangkat fisik terhubung ke internet untuk saling bertukar data dan dapat dikendalikan dari jarak jauh. Smart Pet Feeder yang dikembangkan memanfaatkan IoT untuk mengintegrasikan kamera, sensor, dan modul pengendali dengan aplikasi mobile. Hal ini memungkinkan pengguna memantau kucing, mengatur jadwal pemberian pakan, dan memverifikasi identitas kucing sebelum makanan dikeluarkan. Integrasi ini memastikan bahwa sistem dapat dioperasikan secara efisien dan memberikan pengalaman pemeliharaan hewan yang lebih aman dan praktis.

III. METODE

A. Singkatan dan Akronim

Berikut adalah singkatan dan akronim yang digunakan dalam penulisan jurnal ini :

1. **AI (Artificial Intelligence)** – Kecerdasan buatan, bidang ilmu komputer yang berfokus pada pembuatan sistem yang dapat meniru kecerdasan manusia.
2. **IoT (Internet of Things)** – Jaringan perangkat fisik yang saling terhubung dan dapat bertukar data melalui internet.
3. **mAP (mean Average Precision)** – Metrik evaluasi yang mengukur rata-rata presisi deteksi objek pada berbagai tingkat *recall*.
4. **TP (True Positive)** – Deteksi benar ketika objek yang dimaksud teridentifikasi dengan tepat.
5. **TN (True Negative)** – Prediksi benar ketika objek yang tidak dimaksud berhasil diabaikan oleh sistem.
6. **FP (False Positive)** – Deteksi salah ketika sistem mengidentifikasi objek yang seharusnya tidak ada.
7. **FN (False Negative)** – Kegagalan deteksi ketika objek yang dimaksud tidak teridentifikasi.
8. **YOLO (You Only Look Once)** – Algoritma deteksi objek *real-time* yang melakukan prediksi *bounding box* dan klasifikasi dalam satu tahap.

B. Persamaan

Pengukuran performa model deteksi kucing individu dilakukan menggunakan beberapa metrik evaluasi, yaitu Precision, Recall, Accuracy, dan F1 Score.

Precision mengukur proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan model, sebagaimana ditunjukkan pada persamaan (1).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall mengukur proporsi prediksi positif yang benar terhadap seluruh data positif aktual dalam dataset, sebagaimana ditunjukkan pada persamaan (2).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

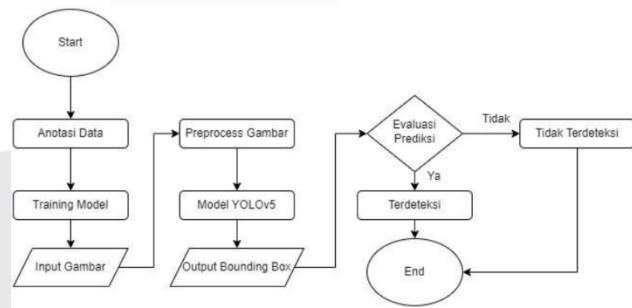
Accuracy merepresentasikan proporsi prediksi yang benar (baik positif maupun negatif) terhadap seluruh data uji, sebagaimana ditunjukkan pada persamaan (3).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

F1 Score merupakan harmonisasi antara nilai Precision dan Recall, yang memberikan gambaran keseimbangan antara keduanya, sebagaimana ditunjukkan pada persamaan (4).

$$F1 Score = 2 * \frac{Precision * Recall}{(Precision + Recall)} \quad (4)$$

C. DIAGRAM SISTEM YOLO



GAMBAR 1
Diagram Sistem YOLO

Diagram pada Gambar 1 menggambarkan alur proses sistem deteksi kucing individu yang digunakan pada Smart Pet Feeder berbasis YOLOv5. Proses dimulai dengan tahap anotasi data, di mana setiap citra kucing diberi penandaan (bounding box) sesuai letak objek yang akan dikenali. Data hasil anotasi kemudian digunakan dalam pelatihan model untuk menghasilkan model deteksi yang mampu mengenali kucing sesuai data pelatihan. Setelah model terlatih, sistem menerima input gambar dari kamera, yang kemudian melalui tahap pra-pemrosesan untuk penyesuaian ukuran, normalisasi, dan peningkatan kualitas citra. Gambar yang telah diproses dimasukkan ke dalam model YOLOv5 untuk menghasilkan output bounding box beserta prediksi identitas

kucing. Selanjutnya, hasil prediksi masuk ke tahap evaluasi untuk memeriksa apakah objek yang terdeteksi sesuai dengan kucing yang telah terdaftar. Jika hasil evaluasi sesuai, status terdeteksi akan dicatat dan sistem melanjutkan ke proses pemberian pakan. Namun, jika tidak sesuai, sistem menandai objek sebagai tidak terdeteksi dan pakan tidak dikeluarkan. Proses ini berulang setiap kali ada citra baru yang masuk dari kamera.

IV. HASIL DAN PEMBAHASAN

Pada saat melakukan percobaan membuat model deep learning terbaik, dibutuhkan percobaan untuk mencari hyperparameter terbaik dengan mengubah hyperparameter setiap percobaan dan mencari hasil yang paling bagus, untuk melakukan ini, dilakukan percobaan mengganti hyperparameter pada dataset yang mengidentifikasi kucing berdasarkan rasnya. Kucing yang diidentifikasi dari model ini adalah Abyssinian, American Shorthair, Bengal, Birman, Bombay, British Shorthair, Egyptian Mau, Maine Coon, Persian, Ragdoll, Russian Blue, Scottish Fold, Siamese, dan Sphynx. Menurut sebuah penelitian yang membandingkan 22 macam ras kucing, ditemukan bahwa kucing-kucing ini memiliki perbedaan genetika sehingga morfologinya pun tidak jauh beda dari yang lain sehingga sering terjadi kesulitan dalam membedakan ras sebuah kucing, sehingga model sedikit kesulitan dalam melakukan identifikasi kepada kucing yang memiliki fitur yang persis. Pada percobaan ini, hyperparameter yang diganti adalah split data, epoch, dan juga learning rate. Berikut hasil percobaannya :

TABEL 1

| NO | SPLIT DATA (TRAIN/VALID/TEST) | EPOCH | LEARNING RATE | MAP50 |
|----|----------------------------------|-------|------------------|-------|
| 1 | 60/20/20 | 50 | 0.1 | 0.68 |
| 2 | | | 0.01 | 0.915 |
| 3 | | | 0.001 | 0.454 |
| 4 | | 100 | 0.1 | 0.846 |
| 5 | | | 0.01 | 0.923 |
| 6 | | | 0.001 | 0.757 |
| 7 | | 200 | 0.1 | 0.897 |
| 8 | | | 0.01 | 0.932 |
| 9 | | | 0.001 | 0.887 |
| 10 | 70/20/10 | 50 | 0.1 | 0.729 |
| 11 | | | 0.01 | 0.928 |
| 12 | | | 0.001 | 0.485 |
| 13 | | 100 | 0.1 | 0.882 |
| 14 | | | 0.01 | 0.933 |
| 15 | | | 0.001 | 0.801 |
| 16 | | 200 | 0.1 | 0.914 |
| 17 | | | 0.01 | 0.923 |
| 18 | | | 0.001 | 0.892 |
| 19 | 80/10/10 | 50 | 0.1 | 0.808 |
| 20 | | | 0.01 | 0.911 |
| 21 | | | 0.001 | 0.564 |

| | | | | | |
|----|--------|--|-----|-------|-------|
| 22 | | | 100 | 0.1 | 0.874 |
| 23 | | | 100 | 0.01 | 0.924 |
| 24 | | | 100 | 0.001 | 0.832 |
| 25 | | | 200 | 0.1 | 0.903 |
| 26 | | | 200 | 0.01 | 0.92 |
| 27 | | | 200 | 0.001 | 0.901 |
| 28 | | | 50 | 0.1 | 0.809 |
| 29 | | | 50 | 0.01 | 0.922 |
| 30 | | | 50 | 0.001 | 0.626 |
| 31 | 90/5/5 | | 100 | 0.1 | 0.911 |
| 32 | 90/5/5 | | 100 | 0.01 | 0.96 |
| 33 | 90/5/5 | | 100 | 0.001 | 0.865 |
| 34 | 90/5/5 | | 200 | 0.1 | 0.922 |
| 35 | 90/5/5 | | 200 | 0.01 | 0.946 |
| 36 | 90/5/5 | | 200 | 0.001 | 0.916 |

Dari hasil percobaan hyperparameter diatas, ditemukan bahwa hasil terbaik selalu ada pada epoch 100 dengan learning rate sebesar 0.01, oleh karena itu dalam pembuatan model deteksi kucing spesifik, setiap training model terbaru akan menggunakan epoch 100 dengan learning rate 0.01.

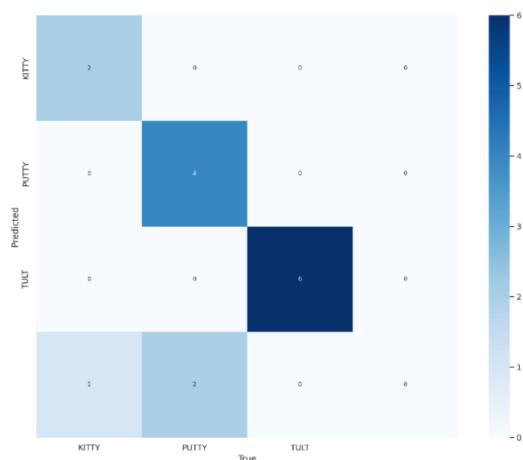
Untuk membuat pengalaman dalam menggunakan aplikasi yang dibuat tidak merepotkan pengguna, maka data foto yang akan digunakan akan meminta 20 foto dari satu kucing milik pengguna dari beberapa sudut dan pencahayaan yang berbeda. Dengan menggunakan hyperparameter yang sudah ditentukan sebelumnya, dilakukan percobaan pembuatan model untuk membedakan 3 kucing berbeda Bernama kitty, putty, dan tult dengan foto terbatas sebanyak 20 foto per kucing. Berikut hasil percobaan pembuatan modelnya :

```
100 epochs completed in 0:02:11 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 3.9MB
Optimizer stripped from runs/train/exp/weights/best.pt, 3.9MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 2763224 parameters, 0 gradients, 4.1 GFLOPs
Class Images Instances P R mAP50 mAP50-95: 100% 1/1 [00:00:00:00, 8.93/s]
all 15 15 0.93 0.768 0.855 0.489
KITTY 15 3 0.79 0.667 0.896 0.134
PUTTY 15 6 1 0.647 0.764 0.465
TULT 15 6 1 0.99 0.995 0.629
Results saved to runs/train/exp
```

GAMBAR 2
Hasil Train Model Kucing Spesifik

Dari hasil training diatas, ditemukan bahwa mAP pada kucing TULT jauh lebih tinggi dibandingkan dari kitty dan putty. Hal ini disebabkan karena kitty dan putty memiliki fitur wajah dan warna yang serupa sehingga model sulit membedakan keduanya, dan karena kucing tult memiliki warna bulu dan fitur wajah yang berbeda, model tidak memiliki kesulitan dalam melakukan prediksi.



GAMBAR 3
Confusion Matrix Model

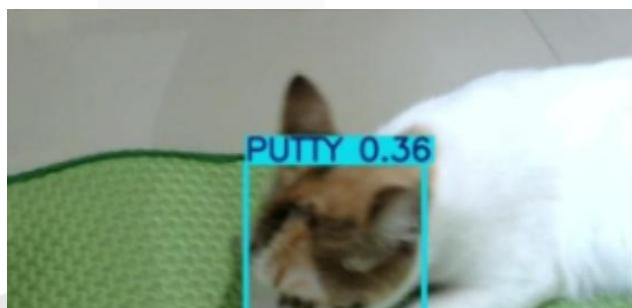
Hasil running model dan confusion matrix diatas adalah hasil testing menggunakan 3 ekor kucing, dan yang didapatkan adalah model cukup baik dalam membedakan ke 3 kucing tersebut, meskipun pada saat memprediksi kitty dan putty, model masih melakukan kesalahan dalam memprediksi kitty dan putty. Confusion matrix menunjukkan performa model dalam mengklasifikasikan 3 jenis kucing dan background.



GAMBAR 4
Kucing TULT



GAMBAR 5
Kucing Kitty



GAMBAR 6
Kucing Putty

Dapat dilihat dari hasil prediksi diatas, model memiliki nilai confidence yang cukup rendah, hal ini dikarenakan jumlah dataset yang kurang mencukupi agar model dapat mendapatkan nilai confidence yang tinggi, dataset harus memiliki banyak foto dari berbagai pose kucing dan sudut pengambilan foto juga sangat berpengaruh. Jadi semakin banyak ragam foto yang digunakan akan membuat model semakin akurat, tetapi dengan mengorbankan kenyamanan penggunaan aplikasi mobile karena user harus melakukan upload foto yang jauh lebih banyak. Apabila memiliki unit komputasi yang bagus, maka saran yang dapat diberikan adalah untuk menambahkan dataset banyak foto kucing sebagai dataset negative agar pada saat model memprediksi kucing yang bukan milik pengguna, maka model akan mendeteksi kucing tersebut sebagai kucing negative dan tidak memaksakannya ke dalam class kucing milik pengguna.

V. KESIMPULAN

Pembuatan fitur untuk melakukan prediksi kucing spesifik menggunakan YOLO, terutama YOLOv5 memungkinkan untuk dilakukan meskipun tidak terlalu akurat karena model akan lebih mudah dalam membedakan warna kucing dan pola bulu kucing. Model hasil training dari YOLO akan kesulitan dalam membedakan fitur wajah seekor kucing karena perbedaan fitur wajah dari kucing satu dan yang lain tidak terlalu berbeda. Oleh karena itu, YOLO akan lebih bagus dalam membaca warna dan pola bulu kucing. Untuk menghindari hasil prediksi *False Positive*, pada saat pembuatan dataset, dapat disiapkan dataset tetap sebelum ditambahkan dataset yang akan diupload oleh pengguna, dataset tetap ini akan berfungsi sebagai dataset negatif agar model tidak memaksakan hasil prediksinya sebagai salah satu kucing aktif pengguna.

REFERENSI

- [1] M. J. Lipinski, L. A. Lyons, S. K. Durward-Akhurst, J. C. Bachman, and N. C. Pedersen, "The ascent of cat breeds:

Genetic evaluations of cats using pedigree and DNA data," Journal of Heredity, vol. 98, no. 6, pp. 537–544, Nov. 2007.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 25, 2012, pp. 1097–1105.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.