

Pengenalan Jenis Rambu Lalu Lintas menggunakan Metode YOLO V5

Anak Agung Gde Bagus Janapriya^{a1}

^aProgram Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Bali, Indonesia
¹bgsjanapr@gmail.com

Abstrak

Meningkatnya jumlah kendaraan bermotor secara signifikan, tidak dapat dipungkiri turut mempengaruhi meningkatnya jumlah kecelakaan lalu lintas. Tingginya angka kecelakaan ini dapat diminimalisir dengan meningkatkan kesadaran masyarakat, dimana salah satunya dengan mengedukasi masyarakat mengenai aturan berlalu lintas. Bentuk edukasi masyarakat mengenai aturan lintas dapat dilakukan dengan mengembangkan sistem berbasis kecerdasan buatan yang dapat mengidentifikasi jenis rambu lalu lintas. Metode kecerdasan buatan yang sering digunakan adalah metode deep learning CNN. Penelitian ini mengembangkan metode untuk melakukan deteksi jenis rambu lalu lintas menggunakan metode YOLO V5 yang merupakan salah satu pengembangan dari metode CNN. Data yang digunakan terdiri dari 96 label jenis rambu lalu lintas dengan jumlah seluruh data sebanyak 1100 citra. Alokasi data train yang digunakan sebanyak 990 citra (90%), sedangkan alokasi data validasi sebanyak 110 citra (10%). Nilai identifikasi model yang diperoleh diantaranya precision sebesar 0,923, recall sebesar 0,826, mAP50 sebesar 0,965, dan mAP50-95 sebesar 0,924.

Keywords: Deteksi Objek, Rambu Lalu Lintas, YOLO V5

1. Pendahuluan

Di era globalisasi yang menuntut kemudahan akses akan transportasi mengakibatkan permintaan jumlah kendaraan bermotor di dunia meningkat secara tajam. Di Indonesia sendiri peningkatan jumlah kendaraan bermotor terbilang cukup signifikan. Menurut data Badan Pusat Statistik pada tahun 2020, jumlah kendaraan bermotor di Indonesia telah mencapai 136 juta kendaraan. Meningkatnya jumlah kendaraan bermotor secara signifikan, tidak dapat dipungkiri bahwa jumlah kecelakaan lalu lintas pun meningkat. Berdasarkan data yang dikeluarkan oleh WHO (*World Health Organization*), sebanyak 1,25 juta orang meninggal akibat kecelakaan lalu lintas yang melibatkan kendaraan bermotor. Direktorat Lalu Lintas (Ditlantas) Polda Metro Jaya menyatakan, rendahnya disiplin tertib dalam berkendara sebagai salah satu penyebab utama kecelakaan lalu lintas di jalan. Data BPS pada tahun 2019, angka kecelakaan menyentuh 116.411, korban meninggal sebanyak 25.671, korban luka berat sebanyak 12.475, dan korban luka ringan sebanyak 137.342. Tingginya angka kecelakaan ini dapat diminimalisir dengan meningkatkan kesadaran masyarakat, dimana salah satunya dengan mengedukasi masyarakat mengenai aturan berlalu lintas.

Beberapa metode *machine learning* konvensional sudah dikembangkan untuk pengenalan rambu lalu lintas. [1] menggunakan metode ekstraksi ciri *Wavelet Haar* dan metode klasifikasi yang berdasarkan jarak *Euclidean*. Penelitian ini hanya menggunakan data citra rambu lalu lintas larangan. Hasil pengujian menunjukkan tingkat akurasi sebesar 92%. Penelitian yang lain [2] mengkombinasikan metode ekstraksi fitur momen-warna dan metode klasifikasi *k-Nearest Neighbor*. Hasil pengujian menunjukkan tingkat akurasi 100% pada $k=3$, 86,6% pada $k=5$, dan 86,6% pada $k=7$. Metode *machine learning* konvensional lain yang dikembangkan pada pengenalan rambu lalu lintas adalah metode *Decision Tree* [3]. Penelitian ini menggunakan algoritma *Local Binary Pattern* sebagai metode ekstraksi ciri dan algoritma *Decision Tree J48* sebagai metode klasifikasi. Hasil pengujian menunjukkan tingkat akurasi 87.5%.

Untuk saat ini, ada kecenderungan pengembangan metode *deep learning Convolutional Neural Network* (CNN) untuk pengenalan rambu lalu lintas dengan menggunakan jumlah dataset yang cukup

banyak dan kelas yang bervariasi. [4] menggunakan metode *Circle Hough Transform* untuk mendeteksi rambu dan metode CNN untuk pengenalan rambu. Hasil pengujian pada penelitian ini menunjukkan tingkat akurasi sebesar 61,3% untuk deteksi rambu dan 75% untuk pengenalan rambu. Penelitian [5] lebih berfokus pada penentuan kombinasi terbaik antara *hyperparameter* beserta arsitektur dalam pengenalan rambu lalu lintas. Eksperimen pada penelitian menunjukkan bahwa arsitektur terbaik adalah *layer* CNN yang tersusun dari *convolutional layer*, *pooling layer* menggunakan operasi *maxpool* dan *fully connected layer* menggunakan algoritma *training Stochastic Gradient Descent (SGD)* pada kombinasi *hyperparameter learning rate* 0,005 dan jumlah filter sebanyak 48. Pada penelitian ini digunakan 10 jenis kelas rambu lalu lintas yang terdiri dari 1750 data citra latih dan 300 data citra uji. Hasil pengujian menunjukkan sistem mengenali rambu lalu lintas dengan galat 0,107 dan akurasi 97,33%. Penelitian CNN yang lain dilakukan pada [6], yang menggunakan sebanyak 10 kelas rambu. dengan jumlah data sebanyak 2050 citra. Arsitektur CNN yang digunakan diantaranya konvolusi dengan ukuran 3x3 sebanyak tiga *layer*, lapisan penggabungan (*Maxpool*) berukuran 2x2 sebanyak tiga *layer*, dan satu *fully-connected layer* dengan fungsi aktivasi *Softmax*. Tiap lapisan konvolusi menggunakan filter sejumlah 32. Jumlah *epoch* yang digunakan sebanyak 20 dan nilai *learning rate* 0,005. Hasil pengujian menunjukkan nilai akurasi mencapai 99,67%.

Pada penelitian ini, penulis menggunakan pengembangan dari arsitektur CNN yaitu model YOLO V5, dengan jumlah kelas untuk jenis rambu lalu lintas yang digunakan adalah 96 kelas, jumlah kelas yang besar yang membedakan penelitian ini dengan penelitian sebelumnya. Dataset yang digunakan sebanyak 1100 citra, dengan alokasi data *training* yang digunakan sebanyak 990 citra (90%), sedangkan alokasi data validasi/*testing* sebanyak 110 citra (10%).

2. Metode Penelitian

2.1 Spesifikasi Dataset

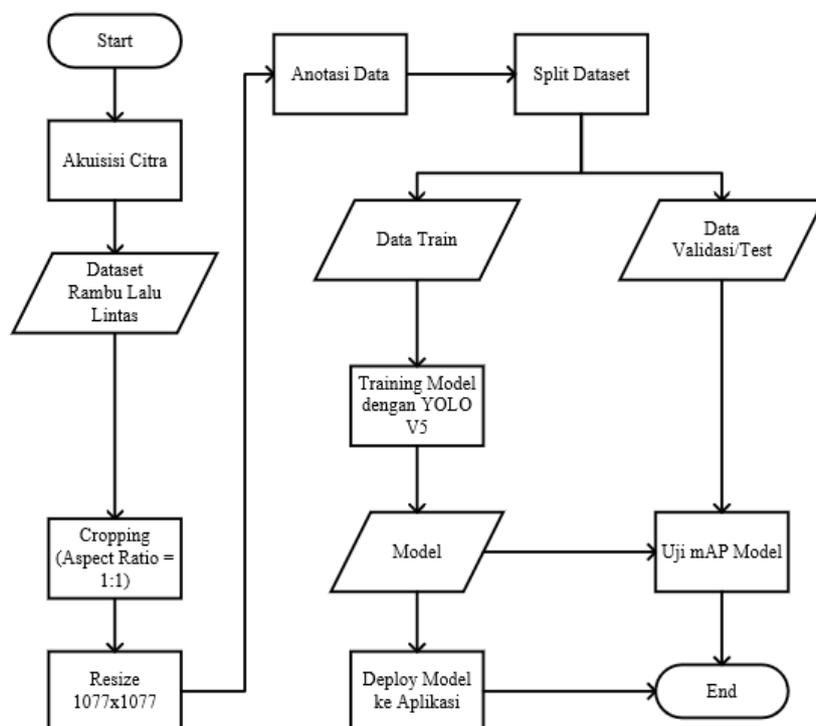
Data yang digunakan berupa citra rambu lalu lintas dengan format JPG. Resolusi citra diseragamkan menjadi ukuran 1077 x 1077 piksel. Dataset terklasifikasi menjadi 96 label jenis rambu lalu lintas, yang terdiri dari 46 label rambu peringatan, 11 label rambu petunjuk, 12 label rambu perintah, dan 27 label rambu larangan. Total data citra keseluruhan sebanyak 1100 citra, yang dibagi menjadi 990 citra (90%) data *train* dan 110 citra (10%) data validasi atau *test*. Contoh citra rambu lalu lintas bisa dilihat pada Tabel 1.

Tabel 1. Spesifikasi Dataset

Kelompok Rambu	Contoh Citra Rambu			Jumlah Label
Rambu Peringatan				46
Rambu Petunjuk				11
Rambu Perintah				12
Rambu Larangan				27
	Total			96

2.2 Tahapan Penelitian

Tahapan penelitian meliputi proses akuisisi citra hingga pengujian performa model pengenalan. Alur dari tiap tahapan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

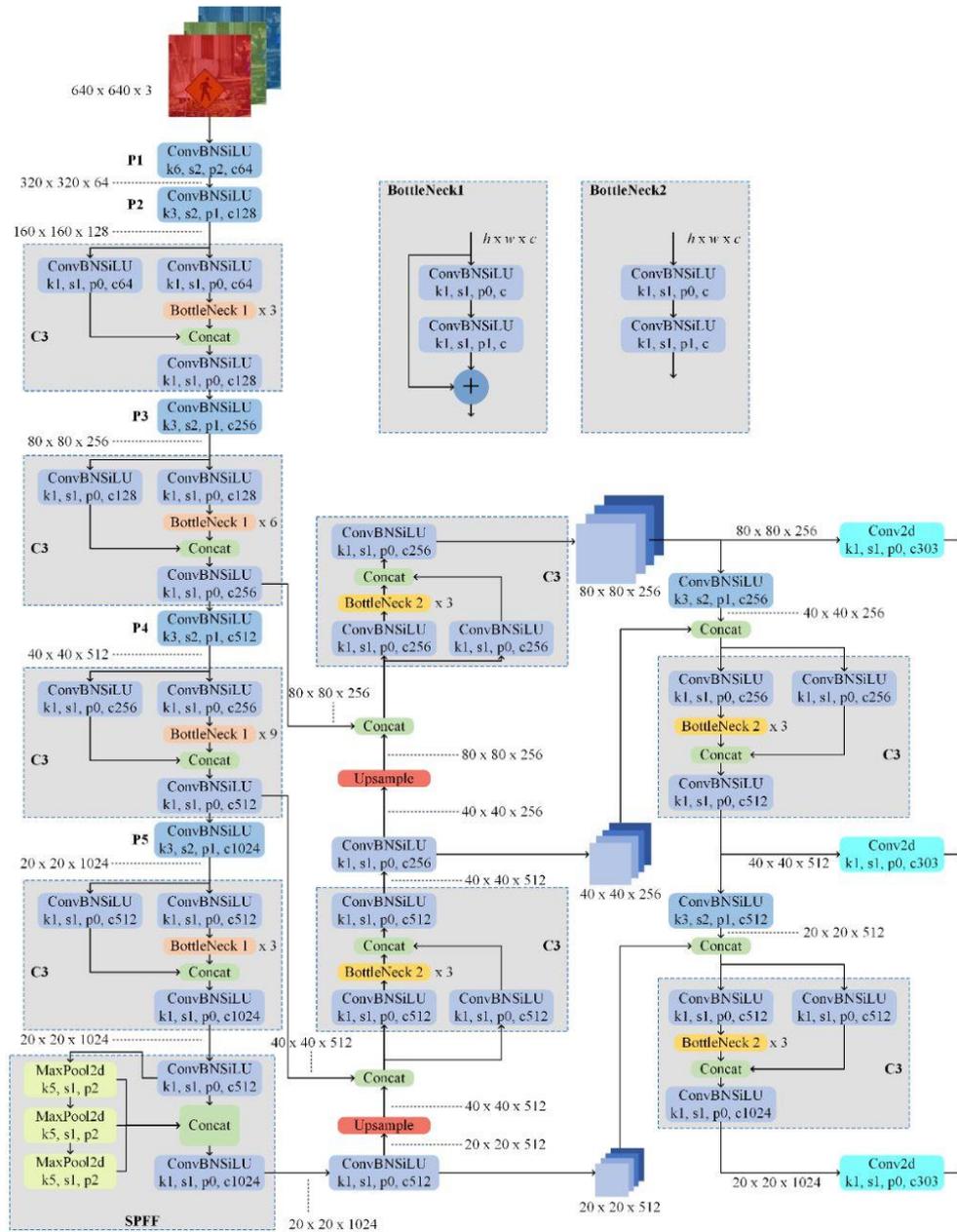
Proses akuisisi citra dilakukan dengan melakukan *screen capture* pada aplikasi *Google Street View*. Lokasi yang digunakan pada aplikasi berada di beberapa jalan yang ada di Bali. *Cropping* citra dilakukan secara manual dengan *tool* bawaan pada Windows 10. Proses *resizing* dilakukan untuk menyeragamkan ukuran citra. Dimensi citra hasil proses *resizing* berukuran 1077 x 1077. *Resizing* dilakukan untuk meningkatkan performa *training* model. Anotasi data adalah proses untuk menandai *region* piksel pada suatu citra dengan label. Proses anotasi data dilakukan dengan menggunakan *tool MakeSense* pada data citra *training* dan citra *testing*.

2.2 Training Model YOLO V5

YOLO singkatan dari "You Only Look Once" adalah metode pendeteksian objek yang membagi gambar menjadi sistem *grid*. Setiap sel dalam *grid* bertugas mendeteksi objek yang ada. YOLO V5 dirilis pada 18 Mei 2020 oleh Glenn Jocher menggunakan *PyTorch Framework* [7].

Proses *training* model dilakukan dengan menggunakan arsitektur YOLO V5. Arsitektur model yang digunakan dapat dilihat pada Gambar 2. Terdapat 4 bagian utama pada model YOLO V5 yaitu *Input*, *Backbone*, *Neck*, dan *Head/Prediction*.

Parameter *k* merupakan ukuran *kernel*, *s* merupakan *stride*, *p* merupakan jumlah *padding*, dan *c* merupakan besar *channel out*. *Resize* citra dilakukan di awal (bagian *Input*) untuk mengubah citra menjadi ukuran 640 x 640 (RGB). Arsitektur YOLO V5 memiliki 3 *output* yang dirancang untuk melakukan deteksi objek pada skala berbeda (objek ukuran besar, sedang, dan kecil).



Gambar 2. Arsitektur YOLO V5 Deteksi Jenis Rambu Lalu Lintas

Input merupakan lapisan (*layer*) pertama pada arsitektur YOLO V5. Lapisan ini menghubungkan citra dengan arsitektur YOLO V5. *Backbone* merupakan bagian arsitektur YOLO V5 setelah citra di-input. Backbone berfungsi untuk melakukan ekstraksi fitur citra. *Neck* merupakan bagian arsitektur YOLO V5 setelah proses ekstraksi fitur pada *Backbone*. *Neck* berfungsi untuk melakukan *feature aggregation*, yaitu menggabungkan fitur gambar untuk diteruskan untuk proses prediksi pada *Head*. *Head/Prediction* merupakan bagian akhir arsitektur YOLO V5. *Head* berfungsi untuk melakukan klasifikasi dan deteksi *bounding box* objek. YOLO V5 menggunakan *Generalized Intersection over Union* (GIoU) sebagai fungsi *loss* dari *bounding box*. YOLO V5 memiliki 3 *output tensor* yang selanjutnya dilakukan *flatten* dan *reshape* untuk dilakukan klasifikasi dan regresi.

Salah satu studi deteksi objek YOLO V5 sebelumnya [8] menggunakan nilai *hyper-parameter learning rate* 0,01, sedangkan untuk *epoch* 300 disarankan dalam dokumentasi YOLO V5 [7]. Hal ini menjadi acuan untuk kombinasi *hyper-parameter* yang digunakan pada penelitian. Nilai *hyperparameter epoch* yang digunakan sebanyak 300, *batch size* sebesar 16, dan *learning rate* sebesar 0,01. *Pre-trained weights* digunakan sebagai *initial weight* pada YOLO V5.

2.3 Metriks Evaluasi

Metriks evaluasi yang digunakan adalah *Precision*, *Recall*, *F1-Score*, *mAP50*, dan *mAP50-95*. *Mean Average Precision (mAP)* merupakan metriks evaluasi dalam deteksi objek. Metode YOLO V5 menggunakan *mAP50* dan *mAP50-95* secara default. *mAP* merupakan metriks evaluasi dalam deteksi objek. Pengukuran metriks *mAP* melibatkan penggunaan matriks *confusion* dalam melakukan penghitungan metriks *Precision* dan *Recall* dari tiap label pada citra [9]. Metriks *mAP50* menggunakan nilai ambang batas *IoU* sebesar 0,5. Metriks *mAP50-95* merupakan rata-rata *mAP* dengan 10 nilai ambang batas *IoU* yang berbeda yaitu 0,5, 0,55, 0,60, 0,65, 0,70, 0,75, 0,80, 0,85, 0,90, dan 0,95 [7].

Matriks *confusion* pada deteksi objek didasarkan pada nilai *IoU* untuk menghitung nilai *True Positive (TP)*, *False Positive (FP)*, dan *False Negative (FN)*. *True Negative (TN)* tidak digunakan karena tidak berpengaruh terhadap performa dari metode pendeteksian objek. Penjelasan dari setiap nilai metriks pada matriks *confusion* bisa dilihat pada Tabel 2.

Tabel 2. Matriks *Confusion* pada Deteksi Objek

Confusion Matrix	Penjelasan
TP (<i>True Positive</i>)	$IoU \geq \alpha$ (<i>threshold IoU</i>).
FP (<i>False Positive</i>)	$IoU < \alpha$ (<i>threshold IoU</i>).
TN (<i>True Negative</i>)	Seluruh bagian citra yang tidak terdapat objek dan tidak dideteksi sebagai objek.
FN (<i>False Negative</i>)	<i>Ground truth</i> yang sama sekali tidak terdeteksi.

3. Hasil dan Pembahasan

3.1. Anotasi Data

Anotasi data adalah proses untuk menandai *region* piksel pada suatu citra dengan label. *Tool MakeSense* digunakan dalam anotasi data pada laporan ini. Label anotasi berupa nama jenis rambu lalu lintas. *Export* dilakukan setelah proses anotasi pada seluruh citra selesai, sehingga diperoleh kumpulan file *.txt* dari tiap citra yang berisi representasi hasil anotasi citra seperti id label rambu lalu lintas dan area *bounding box* objek. Gambar 3 menampilkan contoh hasil anotasi data. Setiap data citra yang digunakan ditandai *bounding box* pada objek rambu lalu lintas beserta nama label rambu.

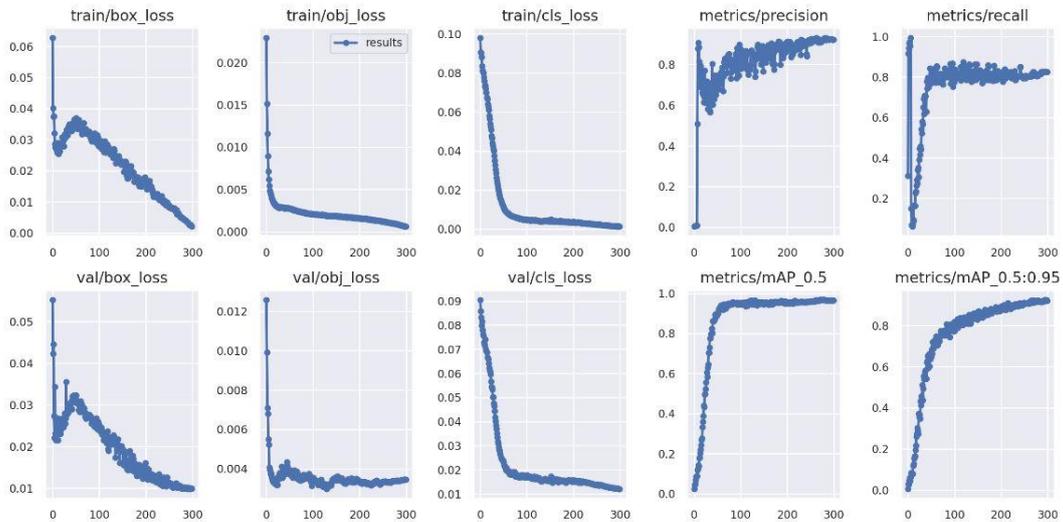
3.2. Training dan Validasi Model YOLO V5

Proses *training* model dilakukan pada data *training*. Metriks evaluasi seperti *Precision*, *Recall*, dan *mAP* diperoleh melalui proses validasi pada model yang dihasilkan.



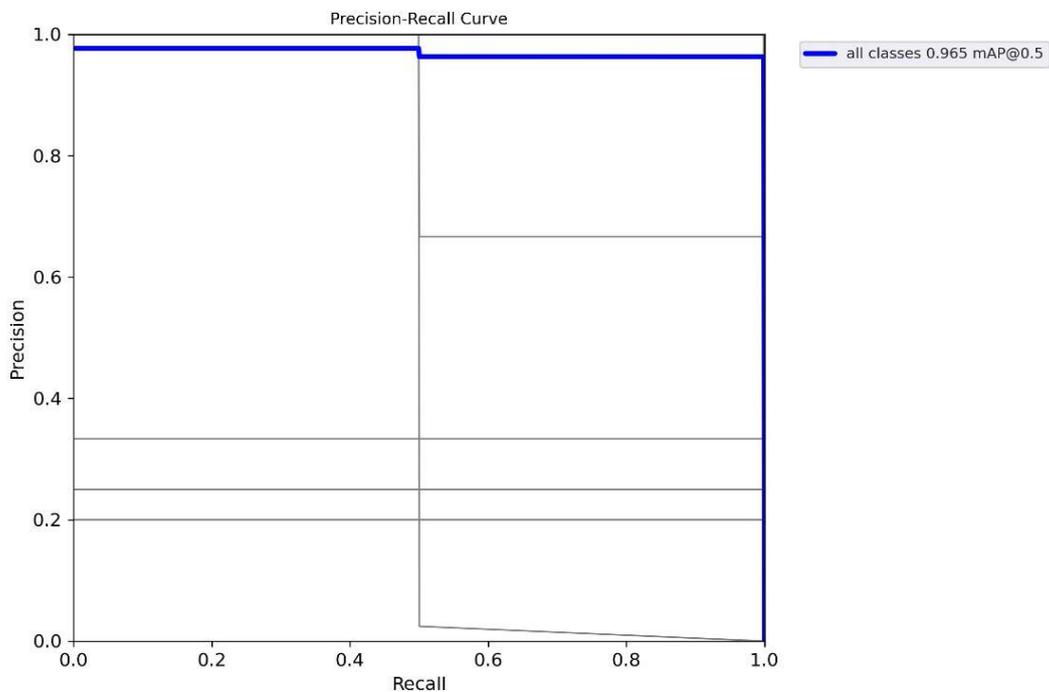
Gambar 3. Contoh Hasil Anotasi Data

Gambar 3 menyajikan sampel data citra yang dilakukan anotasi. Setiap data citra yang digunakan ditandai *bounding box* pada objek rambu lalu lintas beserta nama label rambu.



Gambar 4. Grafik Loss dan Metriks *Training* dan Validasi Model

Grafik *loss* (*box loss*, *object loss*, dan *class loss*) dan grafik metrik (*precision*, *recall*, *mAP@0.5* atau *mAP50*, dan *mAP@0.5:0.95* atau *mAP50-95*) selama proses *training* dan validasi ditunjukkan pada Gambar 4. Grafik menunjukkan penurunan yang stabil pada tingkat *loss* serta kenaikan nilai yang stabil pada grafik metrik selama 300 *epoch*. Nilai metrik keseluruhan (96 label rambu lalu lintas) diantaranya yaitu *precision* sebesar 0,923, *recall* sebesar 0,826, *mAP50* sebesar 0,965, dan *mAP50-95* sebesar 0,924.



Gambar 5. *Precision-Recall Curve* dari *Training Model*

Gambar 5 merupakan grafik *Precision-Recall Curve* secara keseluruhan (pada *mAP@0.5* atau *mAP50*). *Precision-Recall Curve* merupakan plot *precision* dengan *recall* yang menunjukkan *trade-off*

antara dua metrik tersebut untuk berbagai nilai *confidence* pada hasil prediksi model [10]. *mAP* dihitung berdasarkan luas AUC (*Area Under Curve*) yaitu area di bawah kurva *precision-recall*. Besar *mAP* berada pada kisaran 0 sampai 1. Grafik hasil plot label pada Gambar 5 sebagian besar saling berimpitan mendekati nilai 1. Besar *mAP50* yang diperoleh pada keseluruhan label sebesar 0.965.

3.3. Uji Deteksi Rambu Lalu Lintas

Uji deteksi dilakukan dengan melakukan pengujian *mAP* model. Pengujian *mAP* model pengenalan jenis rambu lalu lintas dilakukan terhadap seluruh citra test dari 96 label rambu lalu lintas (total sebanyak 110 citra). Hasil pengujian terhadap model YOLO V5 menghasilkan nilai metrik evaluasi diantaranya yaitu nilai *Precision* sebesar 0,923, *Recall* sebesar 0,826, *mAP50* sebesar 0,965, dan *mAP50-95* sebesar 0,924.

4. Kesimpulan

Pengenalan jenis rambu lalu lintas dengan 96 label berhasil dilakukan dengan menggunakan Model YOLO V5. Hasil pengujian menunjukkan nilai metrik evaluasi yang cukup baik yaitu nilai *Precision* sebesar 0,923, *Recall* sebesar 0,826, *mAP50* sebesar 0,965, dan *mAP50-95* sebesar 0,924. Hasil ini masih bisa ditingkatkan dengan menambahkan lebih banyak data *training* dan data *testing* untuk setiap label. Penambahan data bisa dilakukan dengan menambahkan proses *augmentasi* citra pada tahap *preprocessing*.

Daftar Pustaka

- [1] V. Abdi Gunawan, I. Imelda Fitriani, and L. Sandy Ade Putra, "Klasifikasi Rambu Lalu Lintas Menggunakan Ekstraksi Ciri Wavelet Dan Jarak Euclidean," *Jurnal ELTIKOM*, vol. 3, no. 1, pp. 26–35, 2019, doi: 10.31961/eltikom.v3i1.105.
- [2] R. E. F. Rizarta and D. Avianto, "Pengenalan Citra Rambu Lalu Lintas Menggunakan Ekstraksi Fitur Momenwarna Dan K-Nearest Neighbor," *Computatio : Journal of Computer Science and Information Systems*, vol. 3, no. 1, pp. 39–55, 2019, doi: 10.24912/computatio.v3i1.4272.
- [3] K. Tanuwidjaya and L. Hakim, "Klasifikasi Rambu Lalu Lintas Menggunakan Decision Tree J48 dan Local Binary Pattern," *JITTER-Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 1, pp. 1–7, 2022.
- [4] R. T. Nursetyawan and F. Utamingrum, "Pengembangan Sistem Rekognisi Rambu Kecepatan Menggunakan Circle Hough Transform dan Convolutional Neural Network Berbasis Raspberry Pi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 11, pp. 4012–4018, 2020.
- [5] M. Akbar, "Pengenalan rambu lalu lintas menggunakan convolutional neural networks," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 120–125, 2021, doi: 10.14710/jtsiskom.2021.13959.
- [6] M. Akbar, A. S. Purnomo, and S. Supatman, "Multi-Scale Convolutional Networks untuk Pengenalan Rambu Lalu Lintas di Indonesia," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 3, pp. 310–315, 2022, doi: 10.32736/sisfokom.v11i3.1452.
- [7] G. Jocher, "YOLO V5 by Ultralytics," *GitHub*, 2020. <https://github.com/ultralytics/yolov5> (accessed Feb. 24, 2023).
- [8] Z. Wang, Y. Wu, L. Yang, A. Thirunavukarasu, C. Evison, and Y. Zhao, "Fast Personal Protective Equipment Detection for Real Construction Sites using Deep Learning Approaches," *Sensors*, vol. 21, no. 10, pp. 1–22, 2021, doi: 10.3390/s21103478.
- [9] Adrian Rosebrock, *Deep Learning for Computer Vision with Python!*, 1st ed. PyImageSearch, 2017.
- [10] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The Area Under the Precision-Recall Curve as a Performance Metric for Rare Binary Events," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019, doi: 10.1111/2041-210X.13140.

This page is intentionally left blank.