

# Prototipe Sistem Pengenalan Pelat Kendaraan Otomatis Berbasis YOLOv5 pada Mekanisme Pintu Masuk Departemen Elektro UNHAS

M. Nur Bramasta  
Indonesia-Australia Social Cognitive Robotics  
Departemen Teknik Elektro  
Universitas Hasanuddin  
Gowa, 92171, Sulawesi Selatan, Indonesia  
e-mail: bramasta.mn@gmail.com

## Abstrak

Mekanisme gate di tempat parkir Departemen Teknik Elektro Universitas Hasanuddin, Indonesia mengandalkan sistem RFID untuk memverifikasi akses ke tempat parkir, yang mengharuskan pengguna untuk membawa kartu RFID setiap kali akan menggunakan tempat parkir Departemen. Hal ini mengurangi kenyamanan pengguna setiap kali mereka akan masuk dan keluar dari tempat parkir. Salah satu opsi untuk meningkatkan kemudahan akses adalah dengan mengganti sistem RFID dengan sistem Automatic License Plate Recognition (ALPR), di mana hanya pengguna yang plat nomornya ada di database sistem diterima ke tempat parkir, yang dapat meningkatkan kenyamanan pengguna dan menghilangkan kebutuhan pengguna tempat parkir untuk harus membawa kartu RFID. Dalam paper ini, ALPR diimplementasikan menggunakan algoritme deteksi objek YOLOv5, di mana karakter plat nomor yang dikenali diperiksa terhadap plat nomor yang berada di database sistem. Hasilnya kemudian digunakan untuk menentukan apakah gerbang tetap tertutup atau tidak.

**Kata-kunci:** *Computer vision; YOLOv5; Internet of Things; parking lot; boom barrier; parking gate; ALPR; synthetic training data; Indonesian license plate.*

## 1. Pendahuluan

Lahan kosong yang dikhususkan untuk parkir kendaraan biasa disebut sebagai tempat parkir atau garasi parkir, dan biasanya digunakan oleh sektor publik atau swasta [1]. Salah satu dari beberapa bagian tempat parkir adalah sistem pembatas yang biasa disebut sebagai gerbang parkir, gerbang atau penghalang, yang berfungsi sebagai pengatur pergerakan dan keamanan kendaraan yang masuk dan keluar tempat parkir. [2]. Salah satu cara penghalang

melakukan kendali seperti demikian adalah dengan mengangkat batang atau tiang penghalang ke posisi vertikal untuk memungkinkan keluar-masuk kendaraan, dan diturunkan ke posisi horizontal untuk menghentikannya.

Penghalang dapat dikendalikan secara manual, (dengan tenaga manusia), atau secara otomatis, misalnya dengan mikrokontroler atau komputer. Di tempat parkir Departemen Teknik Elektro Universitas Hasanuddin, sistem penghalangnya dikendalikan secara otomatis; dua *RFID readers* – satu untuk masuk, dan yang lainnya untuk keluar – terhubung melalui komputer Raspberry Pi yang mengontrol mekanisme penghalang [3].

Dalam sistem kendali penghalang saat ini, penghalang berada dalam posisi horizontal – sehingga menutup pintu masuk tempat parkir – secara *default*. Pembaca RFID secara aktif menunggu untuk membaca kartu RFID. Ketika kartu dibaca, identitas (ID) diperiksa terhadap *database* yang disimpan di penyimpanan komputer. Jika ID cocok dengan entri dalam *database*, penghalang dinaikkan, memungkinkan kendaraan untuk melewatinya. Jika tidak, penghalang akan tetap dalam posisi horizontal.

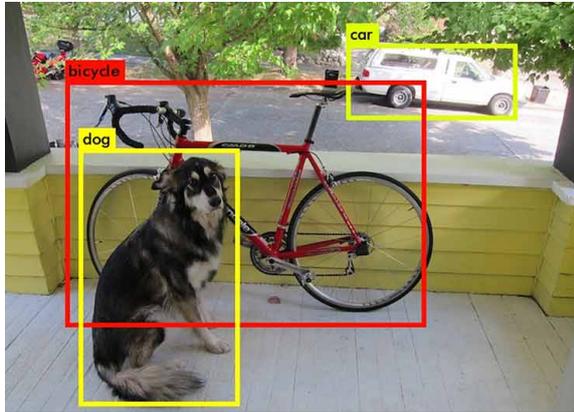
Sistem RFID yang digunakan saat ini memiliki beberapa kelemahan. Pengguna tempat parkir diharuskan mengeluarkan dan secara manual menempelkan kartu RFID mereka ke pembaca, yang dapat mengurangi kenyamanan pengguna. Misalnya, pengemudi mobil diharuskan membuka jendela saat hujan agar dapat mengetuk kartunya untuk mengakses tempat parkir. Pemeliharaan mungkin juga perlu sering dilakukan, karena lingkungan sekitar dapat mempengaruhi kualitas pembaca kartu; penggantian komponen yang lengkap diperlukan untuk salah satu pembaca, karena *pin header*-nya telah berkarat.

Pada paper ini, diusulkan rancang bangun sistem baru yang berupaya menghilangkan kelemahan tersebut dengan mengganti sistem RFID dengan yang didasarkan pada algoritme *Automatic License Plate Recognition* (ALPR). Dengan itu, pengguna tidak perlu lagi secara manual mengidentifikasi diri selain

dengan menghentikan kendaraan mereka di depan gate. Membawa kartu untuk keperluan identifikasi kendaraan juga tidak diperlukan lagi.

## 2. Penelitian Terkait

### 2.1. Deteksi Objek



**Gambar 1. Contoh deteksi objek [4]. Posisi dan skala objek yang terdeteksi ditunjukkan dengan penggunaan kotak pembatas.**

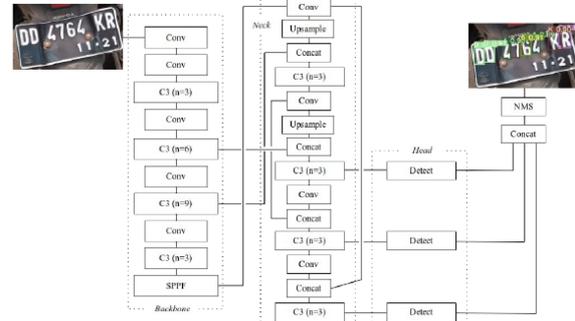
Deteksi objek adalah penerapan bidang pembelajaran mesin untuk tujuan penglihatan komputer, di mana deteksi objek yang memiliki kelas-kelas (*classes*) tertentu (misalnya manusia, bangunan, kendaraan) dilakukan. Dalam deteksi objek, suatu algoritme bertugas untuk menghasilkan daftar objek yang terdeteksi pada suatu citra. Posisi dan skala objek yang terdeteksi kemudian ditunjukkan dengan menggambar kotak pembatas ke masing-masing kelas yang terdeteksi [5].

Arsitektur deteksi objek umumnya terdiri dari tiga bagian: *backbone*, *neck*, dan *head*. *Backbone* detektor objek berfungsi sebagai bagian yang mengekstrak fitur-fitur yang dimiliki oleh citra [6]; *neck* menggabungkan fitur yang diekstraksi di lapisan sebelumnya [7]; dan *head* melakukan prediksi kelas objek yang terdeteksi dan kotak pembatas [7].

### 2.2. YOLOv5

YOLOv5 (*You Only Look Once, version 5*) adalah arsitektur jaringan saraf *convolutional* yang diimplementasikan untuk tujuan deteksi objek. Arsitekturnya dikembangkan oleh Ultralytics, dan merupakan pengembangan dari YOLOv3 versi PyTorch yang juga dikembangkan oleh perusahaan yang sama. Awalnya, arsitekturnya akan diberi nama YOLOv4, namun, sudah ada arsitektur pendeteksi objek lain dengan nama yang sama, yang dikembangkan dari versi YOLOv3 dengan *framework* Darknet. Untuk menghindari tabrakan versi, nama YOLOv5 kemudian digunakan [8].

Arsitektur YOLOv5 hadir dengan beberapa versi: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l dan YOLOv5x. Perbedaan utama antara versi ini terletak pada jumlah modul ekstraksi fitur dan jumlah kernel di lapisan konvolusi [6]. Pada Gbr. 2 arsitektur YOLOv5 dapat dilihat.



**Gambar 2. Arsitektur YOLOv5**

Variabel  $n$  pada Gbr. 2, beserta variabel yang dinamakan *depth multiple* atau *depth gain* ( $g_d$ ) digunakan untuk menentukan jumlah modul C3 yang digunakan arsitektur. Hubungan antara  $n$ ,  $g_d$ , dan banyaknya modul C3 yang digunakan ( $n_{new}$ ) adalah:

$$n_{new} = \begin{cases} 1 & \text{if } n \leq 1 \\ \max(\lfloor n \times g_d \rfloor, 1) & \end{cases} \quad (1)$$

Variabel yang disebut *width multiple* atau *width gain* ( $g_w$ ) digunakan untuk menentukan jumlah kernel yang digunakan arsitektur saat melakukan konvolusi. Jumlah kernel ini secara tidak langsung ditentukan dengan mengubah jumlah *channels* yang ingin di-*output* lapisan Conv ( $D_{out}$ ) :

$$D_{out} \leftarrow \lceil \frac{D_{out} \times g_w}{8} \rceil \times 8 \quad (2)$$

### 2.3. Sensor Ultrasonik

Sensor ultrasonik bekerja dengan cara memancarkan gelombang dengan frekuensi ultrasonik, yaitu jenis frekuensi yang tidak dapat didengar oleh manusia ( $f > 20\text{kHz}$ ) [9]. Gelombang ultrasonik ini kemudian akan dipantulkan kembali ke sensor oleh objek yang ditemui gelombang tersebut. Sensor ultrasonik dapat digunakan untuk mengukur jarak suatu titik ke objek lain, [10] dan dapat dipilih karena biaya yang dibutuhkan oleh sensor lain untuk tujuan yang sama, seperti sensor laser, lebih mahal [11].

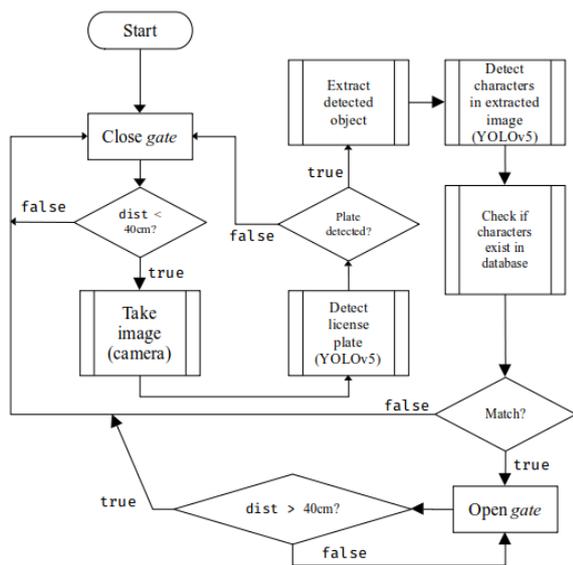
Pada sensor ultrasonik JSN-SR04T, modul *transmitter* memancarkan gelombang ultrasonik dengan frekuensi 40kHz setelah sensor menerima sinyal pemicu (misalnya dari mikrokontroler seperti Arduino, atau dari komputer seperti Raspberry Pi). Setelah gelombang dipantulkan oleh suatu penghalang,

modul *receiver* sensor akan menerima gelombang pantul (disebut juga *echo*) yang frekuensinya sama dengan gelombang yang dipancarkan setelah sensor terpicu, yaitu 40kHz. Waktu tempuh gelombang dari pemancar ke penerima adalah dua kali jarak antara sensor dan penghalang yang memantulkan gelombang ultrasonik yang dipancarkan.

Ketika sensor memancarkan gelombang ultrasonik, juga menghasilkan sinyal TTL yang tinggi, yang menunjukkan bahwa sensor telah mulai menghitung waktu tempuh gelombang. Setelah penerima menerima gelombang yang dipantulkan, sinyal TTL dikembalikan ke rendah, menghentikan penghitungan waktu perjalanan gelombang [12]. Jika waktu tempuh gelombang adalah  $t$ , dan gelombang suara merambat dengan kecepatan  $c$ , maka jarak  $d$  antara sensor dan rintangan di depannya adalah [11]:

$$d = \frac{t \times c}{2} \quad (3)$$

### 3. Usulan Sistem



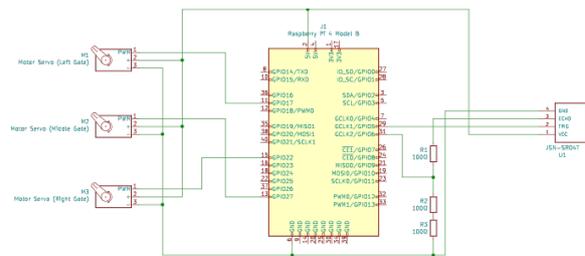
Gambar 3. Diagram alir sistem yang diusulkan.

Pada paper ini dirancang sistem pendeteksi pelat kendaraan untuk simulasi mekanisme buka tutup pintu parkir. Sistem ini diimplementasikan menggunakan teknologi deteksi objek. Alur kerja sistem ini dimulai dengan gerbang tertutup. Jika jarak yang terbaca oleh sensor ultrasonik lebih kecil dari threshold tertentu, kamera akan mengambil citra objek yang menyebabkan ambang batas tersebut tercapai. Menggunakan bantuan YOLOv5, lokasi pelat nomor pada citra yang telah diambil oleh kamera kemudian dideteksi, dan citra tersebut dipotong sehingga hanya bagian pelat nomor yang terdeteksi yang tersisa. Setelah proses ekstraksi ini selesai, YOLOv5 kembali digunakan untuk membaca pelat

nomor yang diekstraksi. Jika urutan karakter yang dibaca ada di *database*, gerbang akan dibuka. Jika tidak, gerbang tidak akan dibuka. Pintu gerbang kemudian ditutup kembali setelah kendaraan meninggalkan lokasi yang ditunjukkan dengan pembacaan sensor ultrasonik yang nilainya lebih besar dari ambang batas yang telah ditentukan. Diagram alir sistem dapat dilihat pada Gbr. 3.

### 3.1. Rancangan Perangkat Keras

Sebuah miniatur lahan parkir dibangun, yang terdiri dari tiga tempat parkir kendaraan, di mana setiap tempat parkir ditutup oleh sebuah gerbang yang digerakkan oleh motor servo. Sensor ultrasonik JSN-SR04T digunakan untuk mensimulasikan keberadaan kendaraan yang berhenti di depan tempat parkir. Satu sensor citra OV5647 digunakan untuk mengambil citra. Seluruh komponen elektronik yang digunakan terhubung dengan komputer Raspberry Pi 4 Model B yang berfungsi sebagai pengontrol komponen elektronik pada prototipe.



Gambar 4. Skematik prototipe.

Tiga pin PWM motor servo yang masing-masing mewakili *gate* kiri, tengah, dan kanan dihubungkan ke pin GPIO17, GPIO27 dan GPIO22 pada komputer Raspberry Pi. Pin TRIG pada sensor ultrasonik JSN-SR04T dihubungkan dengan pin GPIO5 pada Raspberry Pi, dan digunakan pembagi tegangan untuk menghubungkan pin ECHO ke pin GPIO6. Pembagi tegangan diperlukan untuk menurunkan 5V antara pin ECHO dan GND menjadi 3,3V antara GPIO6 dan GND. Tiga resistor 100Ω digunakan untuk pembagi tegangan ini.

Selain motor servo dan sensor ultrasonik yang terhubung ke pin GPIO Raspberry Pi 4, sensor citra OV5647 juga terhubung ke slot kabel CSI-2 pada komputer Raspberry Pi 4. Sensor ultrasonik dan kamera ditempatkan berdampingan di miniatur tempat parkir.

### 3.2. Rancangan Perangkat Lunak

**3.2.1. Pembangunan dataset** Pengumpulan *training data* yang diperlukan dilakukan dengan mengambil dua jenis foto: yang pertama untuk mendeteksi lokasi pelat nomor kendaraan, dan yang kedua untuk mendeteksi urutan karakter milik pelat no-

mor. Tipe pertama terdiri dari citra kendaraan (mobil dan motor) yang dengan jelas memiliki pelat nomor Indonesia yang dengan latar belakang hitam. Jenis kedua terdiri dari citra pelat nomor itu sendiri. Foto-foto yang diperoleh kemudian diberi *annotation*. Proses pelatihan kemudian dilakukan dengan menggunakan dua dataset tersebut, sehingga menghasilkan dua model yang masing-masing berfungsi sebagai pendeteksi lokasi pelat nomor, dan pembaca pelat nomor. Pelatihan digunakan dengan bantuan Kaggle, yang menyediakan modul GPU Tesla P100-PCIE-16GB gratis untuk melatih metode AI.

Anotasi setiap citra dalam kumpulan data disimpan sebagai file teks, di mana setiap citra dalam kumpulan data memiliki file teksnya sendiri, yang menyimpan label objek dalam citra tersebut. Anotasi data di YOLOv5 berbentuk:

$$class, x, y, w, h \quad (4)$$

di mana *class* adalah kelas objek yang dideteksi; *x* dan *y* titik-titik tengah *bounding box*; dan *w* dan *h* panjang dan lebar *bounding box*. Variabel *x*, *y*, *w* dan *h* memiliki tipe data *floating points* dengan *range* 0 ke 1, *class* bertipe data *integer*.

**Tabel 1. Jenis augmentasi yang dilakukan untuk membuat dataset untuk lokalisasi pelat nomor.**

No.	Augmentasi yang dilakukan
1	Ubah perspective citra secara acak. Ubah ukuran citra ke $512 \times 512$ piksel.
2.	Ubah brightness citra secara acak. Ubah perspective citra secara acak. Ubah ukuran citra ke $512 \times 512$ piksel.
3.	Blur citra. Ubah perspective citra secara acak. Ubah ukuran citra ke $512 \times 512$ piksel.
4.	Ubah ukuran citra ke $512 \times 512$ piksel. Ubah perspective citra secara acak. Tambahkan cutout pada bagian-bagian acak pada citra.
5.	Ubah perspective citra secara acak. Flip citra secara horizontal. Ubah ukuran citra ke $512 \times 512$ piksel.
6.	Ubah ukuran citra ke $512 \times 512$ piksel.

Dataset yang digunakan untuk membangun model pendeteksi lokasi pelat nomor ini dibuat dari kumpulan gambar sepeda motor dan mobil yang memiliki pelat nomor Indonesia dengan latar belakang hitam. Koleksinya berisi 311 gambar beranotasi, 248 di antaranya dipilih secara acak untuk pelatihan, dan 63 untuk pengujian. Pada 248 gambar pelatihan tersebut, enam teknik augmentasi yang berbeda diterapkan. Untuk setiap gambar dalam set data pelatihan, enam gambar baru dihasilkan yang merupakan augmentasi dari satu gambar tersebut.

Jenis augmentasi yang dilakukan untuk membangun dataset ini dapat dilihat pada Tabel 1. Setelah augmentasi masing-masing citra selesai, diperoleh 1488 citra pelat nomor, dengan ukuran  $512 \times 512$  piksel dan variasi sudut pandang dan kecerahan.

Dataset yang digunakan untuk membangun model pengenalan karakter pelat nomor ini dibuat dari kumpulan pelat nomor Indonesia yang berlatar belakang hitam. 1025 gambar beranotasi pelat nomor dikumpulkan, 799 di antaranya dipilih secara acak untuk pelatihan, dan 266 untuk pengujian. Augmentasi kemudian dilakukan pada 799 citra. Citra-citra tersebut diputar pertama kali sebesar  $30^\circ$  dan  $-30^\circ$ , menghasilkan 2.397 gambar pelatihan. Kemudian dilakukan enam jenis augmentasi dari 2.397 citra tersebut, yang jenisnya dapat dilihat pada Tabel 2.

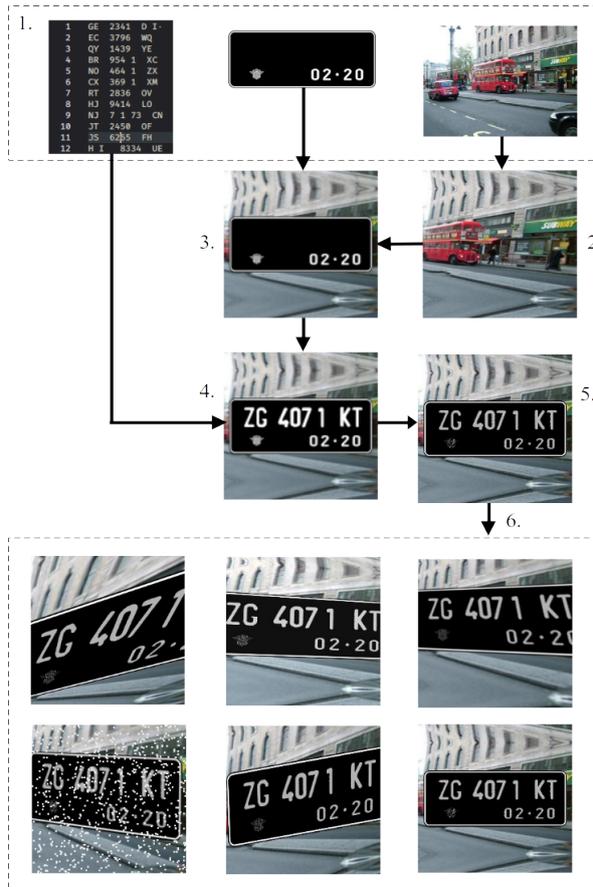
**Tabel 2. Jenis augmentasi yang dilakukan untuk membuat dataset pengenalan karakter.**

No.	Augmentasi yang dilakukan
1	Ubah perspective citra secara acak. Ubah ukuran citra ke $256 \times 256$ piksel.
2.	Ubah brightness citra secara acak. Ubah perspective citra secara acak. Ubah ukuran citra ke $256 \times 256$ piksel.
3.	Blur citra. Ubah perspective citra secara acak. Ubah ukuran citra ke $256 \times 256$ piksel.
4.	Ubah ukuran citra ke $256 \times 256$ piksel. Ubah perspective citra secara acak. Tambahkan cutout pada bagian-bagian acak pada citra.
5.	Ubah perspective citra secara acak. Ubah ukuran citra ke $256 \times 256$ piksel.
6.	Ubah ukuran citra ke $256 \times 256$ piksel.

Setelah augmentasi masing-masing citra selesai, diperoleh 14.382 citra pelat nomor dengan ukuran  $256 \times 256$  piksel, dengan variasi sudut pandang, kecerahan, dan kemiringan.

Selain kumpulan data yang dibuat dari foto pelat nomor kehidupan nyata, kumpulan data sintetik juga dibuat untuk membantu proses pembelajaran. Ini diperlukan karena label untuk beberapa karakter dalam kumpulan data yang dibuat untuk pengenalan karakter lebih sedikit daripada label untuk beberapa karakter lain.

Pembuatan dataset sintetik ini dimulai dengan *generate* 10.000 *strings* dengan pola *letter-letter-number-number-letter-letter*, yang sesuai dengan pola pelat nomor Indonesia. 10.000 string ini kemudian ditempelkan ke gambar pelat nomor yang dibuat komputer, menggunakan *font face* yang diperoleh dengan *tracing* karakter dari citra pelat nomor Indonesia di dunia nyata. Pelat nomor *computer-generated* ini kemudian ditempelkan ke berbagai citra latar belakang.



**Gambar 5. Pembuatan data sintetik (1) pembuatan *strings*, citra pelat nomor kosong dan latar belakang, (2) latar belakang dibuat memiliki lebar dan tinggi yang sama, (3) menempelkan citra pelat nomor kosong ke latar belakang, (4) menempelkan *string* ke dalam citra pelat nomor, (5) menerapkan filter *emboss* dan *spread* ke citra, (6) augmentasi sesuai dengan Tabel 2.**

Kumpulan latar belakang yang digunakan untuk pembuatan kumpulan data diambil dari *Stanford Background Dataset Collection* [13] yang berisi 715 citra dengan ukuran sekitar  $320 \times 240$  piksel. Dataset ini dirancang untuk tujuan segmentasi semantik, sehingga selain kumpulan citra, terdapat juga anotasi dari berbagai objek yang terdapat di setiap gambar. Dalam *paper* ini, hanya citra yang diperlukan dalam pembuatan dataset ini, anotasinya tidak digunakan. Setiap *citra* kemudian diubah ukurannya agar memiliki panjang dan lebar yang sama.

Looping melalui 10.000 *strings* kemudian dilakukan. Di setiap loop, satu dari 715 citra latar diambil secara acak, dan *string* di setiap loop kemudian ditempelkan ke latar belakang yang dipilih. Anotasi setiap karakter pada citra juga dilakukan pada tahap ini. Setelah loop selesai, 10.000 citra pelat nomor sintesis diperoleh dengan anotasinya masing-

masing. Filter *emboss* dan *spread* dalam perangkat lunak GIMP kemudian diterapkan ke masing-masing data sintesis ini. *Emboss* digunakan untuk memberikan efek tiga dimensi pada gambar, dan *spread* untuk menambahkan *noise* pada gambar. Setelah filter diterapkan, dilakukan augmentasi sesuai Tabel 2. Hasil akhir dari pembangkitan dataset ini adalah dataset yang berisi 60.000 citra pelat nomor sintetik, dengan ukuran  $256 \times 256$  piksel dan variasi sudut pandang dan kecerahan. Seluruh citra pada dataset ini ditambahkan pada 14.382 citra *training* yang telah dimiliki, sehingga dimiliki 74.382 citra untuk pelatihan pendeteksi karakter pelat nomor.

Diagram alir proses pembuatan data sintetik ini dapat dilihat pada Gbr. 5.

**3.2.2. Training YOLOv5.** Setelah kumpulan data yang diperlukan selesai dibangun, pelatihan dilakukan dengan bantuan Kaggle, yang menyediakan modul GPU Tesla P100-PCIE-16GB untuk tujuan pelatihan AI. Ukuran *batch* yang digunakan adalah 16, dan jumlah *epoch* yang harus dilalui adalah 100 untuk pendeteksi pelat nomor, dan 10 untuk pendeteksi karakter. Arsitektur YOLOv5s digunakan untuk pendeteksi lokasi pelat nomor, dan YOLOv5m digunakan untuk pendeteksi karakter. Saat melatih detektor karakter pelat nomor, hiperparameter *fliplr* dan *flipud*, yang menentukan probabilitas YOLOv5 untuk membalik gambar yang digunakan untuk pelatihan secara horizontal dan vertikal, disetel ke nol. Nilai *fliplr* yang digunakan selama pelatihan detektor pelat nomor adalah 0,5, yang merupakan nilai default dari hiperparameter. Untuk hiperparameter lain, nilai *default* digunakan.

## 4. Hasil

Prototipe yang dibangun diuji dengan meletakkan foto cetakan pelat nomor kendaraan di depan sensor ultrasonik prototipe, sehingga memicu kamera di sampingnya untuk mengambil gambar.

## 5. Referensi

- [1] H. Börklü and S. Kalyon, "Design Study of an Innovative Barrier System for Personal Parking Lots", *Gazi University Journal of Science Part A: Engineering and Innovation*, pp. 113-123, 2017.
- [2] M. Chaudhuri, "Boom barrier to jaywalking", *The Telegraph India*, 2018. [Online]. Available: <http://www.telegraphindia.com/west-bengal/boom-barrier-to-jaywalking/cid/1410167>.
- [3] D. Djameluddin, "Sosialisasi Penggunaan Parkir Cerdas Pada Departemen Teknik Elektro Universitas Hasanuddin", *Jurnal Tepat*, vol. 4, no. 1,

pp. 94-100, 2021. Available: 10.25042/jurnal\_tepat.v4i1.164.

[4] J. Redmon, "YOLO: Real-Time Object Detection". [Online]. Available: <https://pjreddie.com/darknet/yolo/>. [Accessed: 01- Oct- 2020].

[5] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015. Available: 10.1007/s11263-015-0816-y.

[6] B. Yan et al. "A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5", *Remote Sensing*, vol. 13, no. 9, p. 1619, 2021. Available: 10.3390/rs13091619.

[7] A. Bochkovskiy, C. Wang, and H.M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020. [Online]. Available: arXiv:2004.10934.

[8] J. Solawetz, "YOLOv5 New Version - Improvements And Evaluation", *Roboflow Blog*, 2020. [Online]. Available: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>. [Accessed: 01-Oct-2020].

[9] M. Anshar, R. Sadjad, Dewiani, M. Hanan, R. Prayudha and M. Abry, "Design and Implementation Monitoring and Booking Systems for Smart Parking at Engineering Faculty Campus", *IOP Conference Series: Materials Science and Engineering*, vol. 875, no. 1, p. 012036, 2020. Available: 10.1088/1757-899x/875/1/012036.

[10] M. H. Abiyyi, "Rancang Bangun Sistem Monitoring dan Reservasi Parkir Berbasis Online Lahan Parkir Mobil Fakultas Teknik Universitas Hasanuddin", Undergraduate, Universitas Hasanuddin, 2020.

[11] A. Al-Mahturi and R. Rahim, "Ultrasonic Sensor for Distance Measurement", *Progress in Process Tomography & Instrumentation System*, vol. 24, pp. 9-14, 2016.

[12] S. Dswilan, Harmadi and Marzuki, "Flood monitoring system using ultrasonic sensor SN-SR04T and SIM 900A", *Journal of Physics: Conference Series*, vol. 1876, no. 1, 2021. Available: 10.1088/1742-6596/1876/1/012003.

[13] S. Gould, R. Fulton and D. Koller, "Decomposing a scene into geometric and semantically consistent regions", *2009 IEEE 12th International Conference on Computer Vision*, pp. 1-8, 2009. Available: 10.1109/ICCV.2009.5459211.