

# Sistem Pengenalan Plat Nomor Kendaraan Menggunakan Mask RCNN dan CNN

Anthony Mesakh <sup>1)</sup>

<sup>1)</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,  
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia  
email : [anthonymel211@gmail.com](mailto:anthonymel211@gmail.com)

## ABSTRACT

Plat nomor kendaraan adalah sebuah objek yang berfungsi sebagai tanda pengenal dari sebuah kendaraan. Plat nomor kendaraan dapat digunakan untuk mengidentifikasi sebuah kendaraan secara unik. Sistem yang dapat mengenali plat nomor kendaraan secara otomatis dapat digunakan dalam berbagai macam skenario. Misalnya untuk sistem parkir kendaraan dimana lama kendaraan berada di dalam lapangan parkir dapat ditentukan secara otomatis melalui waktu masuk dan keluar kendaraan. Ataupun pada sistem keamanan dimana plat nomor kendaraan seorang kriminal dapat dideteksi secara otomatis melalui sistem yang dapat memberitahu pihak berwenang secara otomatis. Untuk dapat mengembangkan sistem yang bisa mengenali plat nomor kendaraan secara otomatis, digunakan metode Mask R-CNN dan juga CNN.

Dari 2 model program yang diuji, yaitu model cepat dan akurat, sistem mendapatkan tingkat akurasi 73,8% dan 74,2% untuk keseluruhan karakter dalam plat nomor. Hal ini juga merupakan peningkatan dari hasil penelitian sebelumnya dimana segmentasi karakter membutuhkan keadaan yang sangat spesifik agar bisa mendapatkan hasil yang baik[1].

## Key words

Plat Nomor, Mask R-CNN, CNN

## 1. Pendahuluan

Plat nomor kendaraan adalah salah satu karakteristik dari kendaraan. Plat nomor kendaraan berisi nomor seri unik yang dikhususkan untuk sebuah kendaraan, sehingga plat nomor kendaraan dapat digunakan untuk mengidentifikasi sebuah kendaraan dari kendaraan lainnya. Karena sifat ini, sistem yang dapat mengenali plat nomor kendaraan secara otomatis dapat dimanfaatkan dalam berbagai bidang seperti tilang elektronik, parkir otomatis, pencarian kendaraan curian, dan lainnya.

Sebagai contoh, salah satu implementasi dari pengenalan plat nomor kendaraan adalah sistem parkir. Pada umumnya, sistem parkir membutuhkan interaksi secara manual antara manusia dengan sistem digital. Pengendara yang memasuki area parkir diharuskan untuk

menekan tombol untuk mendapatkan karcis. Saat kendaraan keluar, pengendara memberikan karcis kepada seorang pegawai yang menerima kertas karcis dan menginputnya kepada sistem secara manual, dilanjutkan dengan pembayaran dengan uang tunai atau cashless. Dengan menggunakan sistem pengenalan plat nomor kendaraan otomatis, dapat diimplementasi sistem parkir yang mengidentifikasi plat nomor kendaraan pada waktu masuk dan keluar dari area parkir. Informasi tersebut dapat digunakan untuk menghitung jumlah uang yang harus dibayarkan oleh pengendara dan pengendara dapat membayarkan uang tersebut melalui saluran pembayaran yang tersedia.

Perkembangan teknologi khususnya pada bidang AI (Artificial Intelligence) telah membantu manusia memecahkan masalah-masalah yang sulit dipecahkan dengan program konvensional. Salah satu perkembangan tersebut adalah dikembangkannya pembelajaran mesin (machine learning). Penggunaan machine learning untuk pengenalan objek (object detection) meningkatkan performa dan juga akurasi dari metode sebelumnya (8,64 detik per citra menggunakan metode PCA-SIFT [2]) dengan cukup signifikan (0,1 detik per citra menggunakan metode Faster R-CNN ).

Salah satu metode pengenalan objek berbasis pembelajaran mesin adalah metode Mask R-CNN yang menghasilkan hasil berupa bentuk dari objek yang dideteksi. Model dapat digunakan untuk mengekstraksi instansi dari objek dalam citra, termasuk plat nomor kendaraan. Karakter dalam citra plat nomor juga dapat dikenali dengan basis pembelajaran mesin. Dengan penggabungan metode-metode ini, dapat dikembangkan sistem yang dapat mengenali nomor sebuah kendaraan secara otomatis.

## 2. Modul Pengenalan

Modul pengenalan plat nomor kendaraan dibagi menjadi 3 buah submodul. Yaitu sebagai berikut:

### 2.1 Mask R-CNN

Guna dari modul *Mask R-CNN* [3] dalam program adalah untuk mendapatkan posisi dan bentuk dari plat

nomor dalam citra. Modul *Mask R-CNN* mengeluarkan citra *mask* dalam pewarnaan *grayscale*. Contoh hasil dari modul *Mask R-CNN* dapat dilihat pada gambar berikut ini:



Gambar 1 Citra Mask

## 2.2 Modul Segmentasi Karakter

Untuk melakukan segmentasi karakter, pertama-tama modul mencari keseluruhan titik terluar dalam citra *mask*. Dimana setiap *pixel* yang memiliki paling sedikit 1 *pixel* tetangga berwarna gelap masuk kedalam daftar titik *points*.

Untuk mendapatkan quadrilateral atau 4 titik yang merupakan bentuk dari plat nomor, digunakan persamaan berikut ini:

1. Untuk mencari titik paling kanan, dicari titik dengan komponen x maksimal. Titik ini disebut dengan *p1*.
2. Untuk mencari titik paling kanan-bawah, dicari titik dengan komponen x + y maksimal. Titik ini disebut dengan *p2*.
3. Untuk mencari titik paling bawah, dicari titik dengan komponen y maksimal. Titik ini disebut dengan *p3*.
4. Untuk mencari titik paling kiri-bawah, dicari titik dengan komponen -x + y maksimal. Titik ini disebut dengan *p4*.
5. Untuk mencari titik paling kiri, dicari titik dengan komponen -x maksimal. Titik ini disebut dengan *p5*.
6. Untuk mencari titik paling kiri-atas, dicari titik dengan komponen -x -y maksimal. Titik ini disebut dengan *p6*.
7. Untuk mencari titik paling atas, dicari titik dengan komponen -y maksimal. Titik ini disebut dengan *p7*.
8. Untuk mencari titik paling kanan-atas, dicari titik dengan komponen x - y maksimal. Titik ini disebut dengan *p8*.

Dari 8 titik tersebut, diambil 4 sisi dengan panjang maksimal, dimana sisi-sisi yang dievaluasi adalah:

1. Sisi 1 : (p1, p2)
2. Sisi 2 : (p2, p3)
3. Sisi 3 : (p3, p4)
4. Sisi 4 : (p4, p5)
5. Sisi 5 : (p5, p6)
6. Sisi 6 : (p6, p7)
7. Sisi 7 : (p7, p8)
8. Sisi 8 : (p8, p1)

Setelah didapatkan 4 sisi tersebut, dicari perpotongan diantara 4 sisi tersebut dengan sistem persamaan linear (1) untuk mendapatkan koefisien a dan b untuk sisi tersebut, lalu untuk mendapatkan titik perpotongan, digunakan rumus (2)

$$a = \frac{(y1-y2)}{(x1-x2)}, b = y1 - a * x1 \dots (1)$$

$$x = \frac{(b2-b1)}{(a1-a2)}, y = a1 * x + b1 \dots (2)$$

Keempat titik yang didapatkan menentukan quadrilateral dari citra *mask*. Dimana dengan memetakan keempat titik tersebut kepada sebuah persegi empat, didapatkan citra tanpa distorsi seperti pada gambar 2.



Gambar 2 Citra Setelah Perspective Correction

Untuk memetakan ulang sebuah pixel, digunakan rumus interpolasi linear sebagai berikut:

$$f_x(x_{min}, x_{max}, t) = x_{min} + (x_{max} - x_{min}) * t \dots (3)$$

$$f_y(y_{min}, y_{max}, t) = y_{min} + (y_{max} - y_{min}) * t \dots (4)$$

Variabel  $x_{min}, x_{max}, y_{min}, y_{max}$  didapatkan dari quadrilateral untuk mendapatkan titik x dan y pada citra baru.

Setelah didapatkan rumus interpolasi linear untuk komponen x dan y, diciptakan citra baru berukuran 600\*150 *pixel*, dimana untuk setiap *pixel* pada citra tersebut, diberikan value yang diambil dari citra lama dengan posisi yang didapatkan dari rumus interpolasi linear tersebut. Sebagai contoh, untuk titik (200,30) diambil pixel dari citra asli dengan posisi  $x = f_x(a, b, 0.33)$  dimana t adalah  $0.33 = \frac{200}{600}$ . Dan posisi  $y = f_y(a, b, 0.2)$  dimana t adalah  $0.2 = \frac{30}{150}$ .

Setelah didapatkan citra dengan perbaikan perspektif, pada citra dilakukan binary thresholding [4] untuk menghasilkan citra seperti pada gambar 3.



Gambar 3 Binary Thresholding

Dimana akan dicari *bounding box* setiap grup *pixel* berwarna putih, dan setiap bounding box yang memenuhi kriteria persamaan (5) berikut:

$$15 \geq w \geq 80, 50 \geq h \geq 110 \dots (5)$$

Dimasukan kedalam modul klasifikasi karakter.

## 2.3 Modul Klasifikasi Karakter

Modul ini mengklasifikasikan citra karakter yang masuk kedalam 36 kategori karakter. Citra karakter yang masuk berupa citra dengan ukuran 28\*28 *pixel* seperti gambar 4.

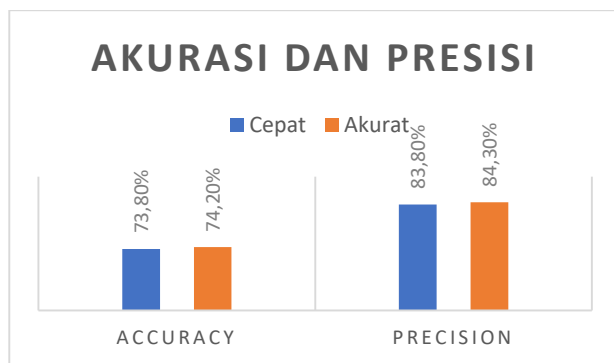


Gambar 4 Citra Karakter

Modul ini merupakan sebuah *neural network* dengan arsitektur *CNN*, dengan 2 *layer convolution* yang menggunakan aktivasi *ReLU*, dengan lapisan *Max Pooling*, yang diakhiri dengan *dense layer* dengan input 2450 neuron dan keluaran 1000 neuron, dan diakhiri dengan *fully connected layer* dengan keluaran 36 neuron, dengan fungsi aktivasi *softmax* untuk mengklasifikasikan citra kedalam 1 kelas karakter.

### 3. Hasil Percobaan

Program dibagi menjadi 2 buah modul, yaitu modul cepat dan akurat. Modul akurat melaksanakan cropping secara otomatis sebelum memasukan citra kedalam modul pengenalan. Akurasi yang didapatkan dari kedua modul tersebut adalah sebagai berikut (Gambar 5):



Gambar 5 Grafik Akurasi dan Presisi Algoritma

Dengan perbedaan performa diantara kedua modul tersebut adalah sebagai berikut (Tabel 1):

Tabel 1 Perbandingan algoritma A dan algoritma B

Algoritma	Waktu Proses
Cepat	928 ms
Akurat	1684 ms

Hasil tersebut didapatkan dari dataset validasi yang terdiri dari 76 citra dengan total 622 karakter. Percobaan dilakukan dengan perangkat keras sebagai berikut:

1. CPU Intel® Core™ i5-8250U
2. GPU NVIDIA® GeForce® MX150 2GB
3. 8 GB DDR4 RAM

### 4. Kesimpulan

Program dapat mengenali plat nomor kendaraan dengan tingkat keakurasian yang mencapai 74,2% untuk modul akurat, dan 73,8% untuk modul cepat.

Program juga dapat mengenali plat nomor yang tidak tegak tegak lurus terhadap kamera dalam citra dengan cukup baik dan modul juga bekerja dengan baik dalam kondisi dimana kontras antara kendaraan dengan plat nomor rendah (kendaraan berwarna hitam) dibandingkan metode sebelumnya [5].

Untuk memperbaiki akurasi dari program ini, jumlah dari data pelatihan dapat ditingkatkan sehingga *Mask R-CNN* lebih akurat dalam menentukan citra plat nomor kendaraan. Lalu perangkat yang digunakan juga dapat lebih ditingkatkan sehingga citra yang masuk kedalam modul *Mask R-CNN* tidak perlu melewati tahapan *cropping* untuk mendapatkan hasil yang lebih memuaskan.

### REFERENSI

- [1] Erlinda, Kristina. RLSA Untuk Segmentasi Plat Mobil Dan Proses Pengenalan Dengan Backpropagation. Jurnal Ilmu Komputer Dan Sistem Informasi Fakultas Teknologi Informasi Universitas Tarumanagara 3, no. 1 (Jakarta: FTI Untar, 2015).
- [2] Ke, Yan, and Rahul Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2004. <https://doi.org/10.1109/cvpr.2004.1315206>.
- [3] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence 42, no. 2 (2020). <https://doi.org/10.1109/TPAMI.2018.2844175>.
- [4] Gonzalez, Rafael C, and Richard E Woods. Digital Image Processing, Global Edition. Digital Image Processing, Global Edition. 4th Edition. Vol. 19. (New York, NY: Pearson, 2017).
- [5] Kristina, loc. cit.

**Anthony Mesakh**, mahasiswa program Teknik Informatika di Universitas Tarumanagara, Indonesia. Selama perkuliahannya aktif dalam organisasi Robotics Untar dan menjabat sebagai kepala divisi berkaki pada tahun ajaran 2020/2021 dan sebagai ketua umum pada tahun ajaran 2021/2022.