

# IDENTIFIKASI PLAT NOMOR KENDARAAN MENGGUNAKAN METODE *K-NEAREST NEIGHBOR*

## IDENTIFICATION OF VEHICLE NUMBER PLATE USING *K-NEAREST NEIGHBOR METHOD*

Tirtha Dewanca I<sup>1</sup>, Anggunmeka Luhur Prasasti S.T., M.T.<sup>2</sup>, Randy Efra Saputra S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[tierhaatha@student.telkomuniversity.ac.id](mailto:tierhaatha@student.telkomuniversity.ac.id), <sup>2</sup>[anggunmeka@telkomuniversity.ac.id](mailto:anggunmeka@telkomuniversity.ac.id),

<sup>3</sup>[resaputra@telkomuniversity.ac.id](mailto:resaputra@telkomuniversity.ac.id)

---

### Abstrak

Tempat parkir adalah salah satu komponen penting pada fasilitas umum, dimana tempat parkir yang baik mendukung keoptimalan pelayanan fasilitas umum pada masyarakat. Tempat parkir yang baik biasanya dilengkapi dengan sistem parkir. Sistem parkir yang banyak diterapkan sekarang adalah sistem parkir konvensional dimana sistem tersebut memiliki beberapa kekurangan yakni adanya penggelapan uang parkir yang dilakukan oleh perorangan. Maka penting untuk dirancang sebuah otomasi dalam sistem parkir berbasis plat nomor sebagai identitas unik kendaraan untuk mengatasi kekurangan tersebut.

Pada tugas akhir ini membahas mengenai pengenalan plat nomor kendaraan menggunakan metode *K-Nearest Neighbor*. Hasil akurasi yang didapatkan dari penelitian ini adalah sebesar 96.6% dengan parameter optimal yang digunakan adalah *filter Gaussian*, nilai *threshold* 61 dan nilai  $k = 3$ . Hasil akurasi untuk keadaan dilapangan yang didapatkan dari penelitian ini adalah sebesar 97% dengan jarak optimal 50 cm, sudut 40° dan intensitas cahaya pada 1000 - 1500lx.

**Kata Kunci :** Klasifikasi, *K-Nearest Neighbor*, plat nomor kendaraan

---

### Abstract

*Parking space is one of the important components of public facilities, where a good parking lot supports the optimization of public facilities services in the community. A good parking lot is usually equipped with a parking system. The parking system that is widely applied now is a conventional parking system where the system has several shortcomings, namely the existence of embezzlement of parking money by individuals. So it is important to design an automation in the number plate-based parking system as a vehicle's unique identity to overcome these deficiencies.*

*In this final project discusses the introduction of vehicle number plates using the *K-Nearest Neighbor* method. Accuracy results obtained from this study amounted to 96.6% with optimal parameters used are *Gaussian filters*, 61 threshold values and  $k = 3$ . Accuracy results for field conditions obtained from this research are 97% with an optimal distance of 50 cm, angle 40° and intensity light at 1000 - 1500lx.*

**Keyword :** Classification, *K-Nearest Neighbor*, license plate

---

## 1. Pendahuluan

Tempat parkir adalah salah satu komponen penting pada fasilitas umum, dimana tempat parkir yang baik mendukung keoptimalan pelayanan fasilitas umum pada masyarakat. Tempat parkir yang baik biasanya dilengkapi dengan sistem parkir. Disisi lain menurut beberapa jurnal yang penulis baca, penulis menemukan kasus penggelapan uang parkir oleh beberapa oknum yang menyebabkan kerugian negara sebesar dua puluh satu milyar rupiah yang terjadi pada 2016 dan 2017[1]. Untuk mengatasi kedua masalah diatas penulis berinisiatif untuk membuat *Automatic Number Plate Recognition (ANPR)* dengan menggunakan metode *K-Nearest Neighbor*.

## 2. Landasan Teori

### 2.1 Plat Nomor Kendaraan

Lalu lintas dan kendaraan mempunyai fungsi strategis dalam mendukung pembangunan dan integrase nasional. Sehubungan dengan itu maka dibentuklah peraturan mengenai lalu lintas dan kendaraan. Peraturan tersebut memuat berbagai macam ketentuan yang harus dipatuhi oleh pengguna lalu lintas, khususnya bagi kendaraan bermotor yang dioperasikan di jalan raya. Selain memiliki Surat Tanda Nomor Kendaraan (STNK), setiap kendaraan bermotor baik roda dua atau lebih juga harus memiliki Tanda Nomor Kendaraan Bermotor (TNKB) yang diterbitkan oleh samsat yaitu plat yang sudah ditetapkan dalam ketentuan pasal 178

Peraturan Pemerintah Nomor 44 Tahun 1993 Tentang Kendaraan dan Pengemudi. Plat TNKB tersebut dibuat secara spesifik dan telah memenuhi syarat baik segi bentuk, jenis, angka dan huruf nomor polisi yang tertera dalam plat tersebut. Tanda Nomor Kendaraan Bermotor (TNKB) tersebut harus memuat kode wilayah, nomor registrasi dan masa berlaku. Tanda Nomor Kendaraan Bermotor ini sering disebut dengan Nomor Polisi (Nopol) atau Plat Nomor.

## 2.2 Citra Digital

Citra digital merupakan gambar pada pada bidang dua dimensi. Ditinjau dari sudut pandang matematis citra digital merupakan fungsi *continue* dari intensitas cahaya pada bidang dua dimensi tersebut. Pada umumnya citra pada komputer direpresentasikan menggunakan model warna RGB dengan *range* 0 – 255.

Citra digital sendiri merupakan sebuah *array* yang berisi nilai – nilai *real* dan kompleks yang diimplementasikan ke dalam deretan bit tertentu. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi.

## 2.3 RGB

RGB adalah gambar berwarna yang terdiri dari nilai tetap untuk setiap piksel, RGB memiliki nilai yang berbedanya mulai dari 0 hingga 255 [4]. RGB merupakan model warna pencahayaan (*additive color mode*) dipakai untuk *input devices* seperti *scanner* maupun *output devices* seperti *display monitor*. Warna-warna primernya (*Red, Green, Blue*) tergantung pada teknologi alat yang dipakai seperti CCD atau PMT pada *scanner* atau digital camera, CRT atau LCD pada *display monitor*. RGB memiliki spektrum yang sangat besar sehingga warna dapat di tampilkan bermacam – macam [2].

## 2.4 OpenCV

OpenCV (Open Source Computer Vision Library) adalah sebuah library open source yang dikembangkan oleh Intel yang fokus untuk menyederhanakan program terkait citra digital. Library ini terdiri dari fungsi-fungsi Computer Vision dan API (Application Programming Interface) untuk *image processing high level* maupun *low level* dan sebagai optimasi aplikasi *real-time* [3]. OpenCV dapat digunakan pada bahasa pemrograman C/C++ dan telah dikembangkan ke python, java, matlab.

## 2.5 Noise Reduction

*Noise Reduction* adalah proses menghilangkan *noise* dari suatu sinyal. Semua perangkat pemrosesan sinyal, baik analog maupun digital, memiliki sifat yang membuatnya rentan terhadap *noise*. *Noise* dapat berupa *random* atau putih dengan distribusi frekuensi genap. *Noise* bergantung pula pada mekanisme perangkat atau algoritma pemrosesan citra[4] .

Ada beberapa cara penggunaan *noise reduction* diantaranya menggunakan *Gaussian filter* (dikenal juga sebagai *Gaussian smoothing*) dan *Mean filter*, berikut penjelasannya:

1. *Mean filter* atau disebut dengan *filter* rata-rata berfungsi untuk mengaburkan gambar dan menghilangkan *noise* yang terdapat pada gambar. Ditentukan dengan nilai rata-rata piksel dalam kernel  $n \times n$ . Nilai piksel tengah kemudian diganti dengan nilai rata-rata. Berikut penjeleasan rumus untuk penerapan *mean filter*

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ & & & \dots & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix} \quad (1)$$

2. *Gaussian filter* lebih melibatkan rata-rata tertimbang piksel sekitarnya dan memiliki parameter sigma. Kernel disini mewakili perkiraan diskrit dan distribusi *Gaussian*. Secara matematis, menerapkan keburaman *Gaussian* ke gambar sama dengan menggabungkan gambar dengan dua fungsi *Gaussian*. *Gaussian* menyebabkan efek gambar yang lebih *smooth* sehingga *noise* tipe *Gaussian* yang tersebar di gambar dapat dihilangkan[5]. Untuk menghitung transformasi yang diterapkan ke setiap piksel dalam gambar dapat diterapkan rumus fungsi *Gaussian* dalam satu dimensi dan dua dimensi sebagai berikut

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Perbaikan citra juga sangat diperlukan untuk menghilangkan *noise*, karena pada dasarnya perbaikan citra adalah langkah pertama dalam pemrosesan citra untuk meningkatkan kualitas dengan memperkuat tepi dan menghaluskan area citra. Berikut beberapa teknik untuk perbaikan citra [6]:

#### 1. *Spatial Domain Method*

Dalam metode ini, perbaikan citra dapat dilakukan dengan memanipulasi setiap nilai piksel yang disebut juga dengan perlakuan titik

#### 2. *Frequency Domain Method*

Dalam metode ini terdapat dua frekuensi yang mempengaruhi yaitu frekuensi tinggi dan frekuensi rendah. Untuk menghasilkan gambar yang lebih halus dapat menghilangkan frekuensi tinggi dan untuk menghasilkan gambar yang lebih tajam dapat menghilangkan frekuensi rendah. Metode ini juga dapat disebut sebagai filter high-pass dan filter low-pass menggunakan transformasi Fourier

#### 3. *Hybrid Method*

Metode ini merupakan kombinasi metode domain spasial dan domain frekuensi yang dapat meningkatkan citra inframerah dengan mengatur kontras dengan menghaluskan garis tepi. Pendekatan ini menggunakan algoritma seperti fungsi *Gaussian* untuk memperbaiki detail, tepi atas, dan tepi bawah untuk menghaluskan konten gambar.

### 2.6 Citra Grayscale

Citra *grayscale* adalah suatu citra yang hanya memiliki warna tingkat keabuan. Penggunaan citra *grayscale* dikarenakan karena membutuhkan sedikit informasi pada tiap pixel dibandingkan dengan citra berwarna. Gambar *grayscale* dikenal juga sebagai hitam putih, dimana masing – masing nilai *pixel* hanya membuat informasi intensitas[7]. Rumus untuk konversi citra RGB pada tiap sel (x,y) menjadi skala keabuan adalah sebagai berikut[8]:

$$f(x,y) = 0,289 \times R + 0,5870 \times G + 0,1140 \times B (3)$$

### 2.7 Operasi Morfologis

Operasi morfologis dalam image processing bertujuan untuk menghilangkan ketidaksempurnaan dengan memperhitungkan bentuk dan struktur gambar. Ada beberapa jenis operasi morfologis diantaranya adalah operasi *top hat* dan *black hat*. Operator ini digunakan untuk mengisolasi citra yang dimana lebih terang atau lebih redup daripada tetangga terdekatnya. Operator *top hat* mengurangi pembukaan dari A ke A, sedangkan operator *black hat* menebalkan area yang lebih gelap dari wilayah sekitarnya[9].

### 2.8 Thresholding

Thresholding merupakan salah satu metode segmentasi citra dengan memisahkann antara objek dengan *background* dalam suatu citra berdasarkan pada perbedaan tingkat kecerahan atau gelap. Region citra yang memiliki nilai intensitas rendah akan dibuat semakin gelap atau hitam (0), sedangkan region citra yang memiliki nilai intensitas tinggi akan dibuat semakin terang atau putih (1).

### 2.9 Region Of Interest (ROI)

*Region of Interest* adalah suatu bagian dari citra yang dipilih untuk kemudian diproses [10]. *Region of Interest* adalah cara yang tepat untuk mengurangi tingginya waktu pemrosesan. ROI mampu menandai area tertentu sehingga dapat digunakan untuk mengoptimalkan kinerja sistem untuk mendeteksi, menghitung dan mengklasifikasi secara realtime [11].

Daerah dibedakan dengan menggunakan klasifikasi dan masking. Jika piksel pada mask tidak nol, maka pemrosesan citra dilakukan. Sebaliknya jika piksel pada mask sama dengan nol, proses tidak dijalankan. Setelah daerah yang diinginkan ditemukan, daerah tersebut ditandai dengan kotak untuk membatasi daerah yang akan dikenali.

### 2.10 Ekstraksi Ciri

Ekstraksi ciri adalah proses pengurangan dimensionalitas dimana set awal data mentah direduksi menjadi kelompok yang lebih mudah dikelola untuk diproses. Karakteristik dari kumpulan data besar ini adalah sejumlah besar variabel yang membutuhkan banyak sumber daya komputasi untuk diproses. Ekstraksi ciri adalah nama metode untuk memilih atau menggabungkan variabel menjadi ciri, secara efektif mengurangi jumlah data yang harus diproses secara akurat dan lengkap yang menggambarkan kumpulan data asli.

### 2.11 K-Nearest Neighbor

*K – Nearest Neighbor* merupakan salah satu metode pengklasifikasian. Metode ini mengklasifikasikan hasil kueri instance yang baru berdasarkan jarak terdekat antara data yang akan dievaluasi dengan nilai *k* terdekatnya[12]. Tujuannya untuk mengklasifikasikan objek baru berdasarkan atribut dan data training. Klasifikasi sendiri dilakukan tanpa menggunakan model, tetapi hanya berdasarkan memori.

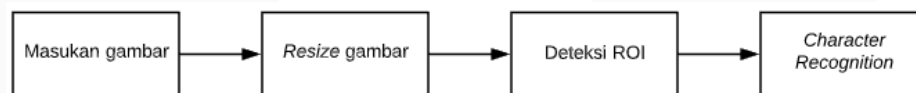
Cara kerja algoritma ini adalah data uji diklasifikasikan ke dalam kelas tertentu berdasarkan jumlah tetangga terdekat dengan menggunakan perhitungan jarak *Euclidian*. Saat menghitung, jarak antara data uji dan semua dataset dengan label kelasnya disimpan dalam vektor, kemudian diurutkan berdasarkan jarak terpendek[13]. Berikut adalah fungsi untuk mencari jarak tertentu dengan menggunakan rumus *Euclidean*.

$$\sqrt{\sum_{i=1}^p (X_{2i} - x_{1i})^2} \quad (4)$$

## 3. Perancangan Sistem

### 3.1 Gambaran Umum Sistem

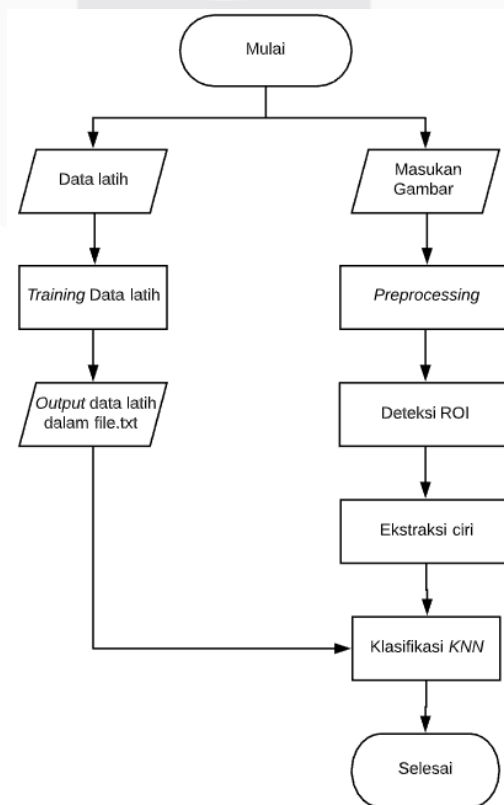
Secara umum sistem yang akan dibangun adalah sebuah program untuk mengenali plat nomor melalui video secara *real-time*. Pada pengerjaan tugas akhir ini pengenalan identifikasi plat nomor ini akan memiliki dua proses utama, yaitu proses training dan proses testing pada kedua skema program (*real-time*). Diagram blok yang akan dibuat pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Diagram Blok Sistem

### 3.2 Perancangan Sistem

Flowchart gambar 2 adalah gambaran mengenai alur proses yang akan dilakukan dengan proses pertama yaitu memasukan gambar data latih kedalam sebuah folder yang nantinya akan dilakukan training. Setelah didapatkan data latih, proses dilanjutkan dengan pengambilan gambar melalui kamera.

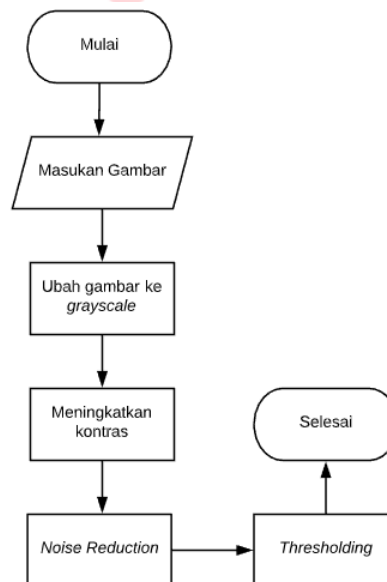


Gambar 2. Flowchart Perancangan Sistem

Terdapat *preprocessing* yaitu proses yang digunakan untuk mendeteksi plat nomor yang telah diambil gambarnya oleh kamera. Tentukan letak plat nomor kendaraan dengan deteksi ROI yang dilanjutkan dengan menentukan letak karakter yang ada dalam plat nomor tersebut. Ambil data training yang sudah dimasukkan kedalam file lalu dilakukan klasifikasi menggunakan KNN. Evaluasi nilai  $k$  yang terbaik untuk mendapatkan *output* maksimal.

### 3.2.1 PreProcessing

Tahapan ini adalah tahap pertama dimana dilakukan pemrosesan gambar. Gambar 3 merupakan urutan proses yang akan dilakukan pada tahapan ini.



Gambar 3. PreProcessing

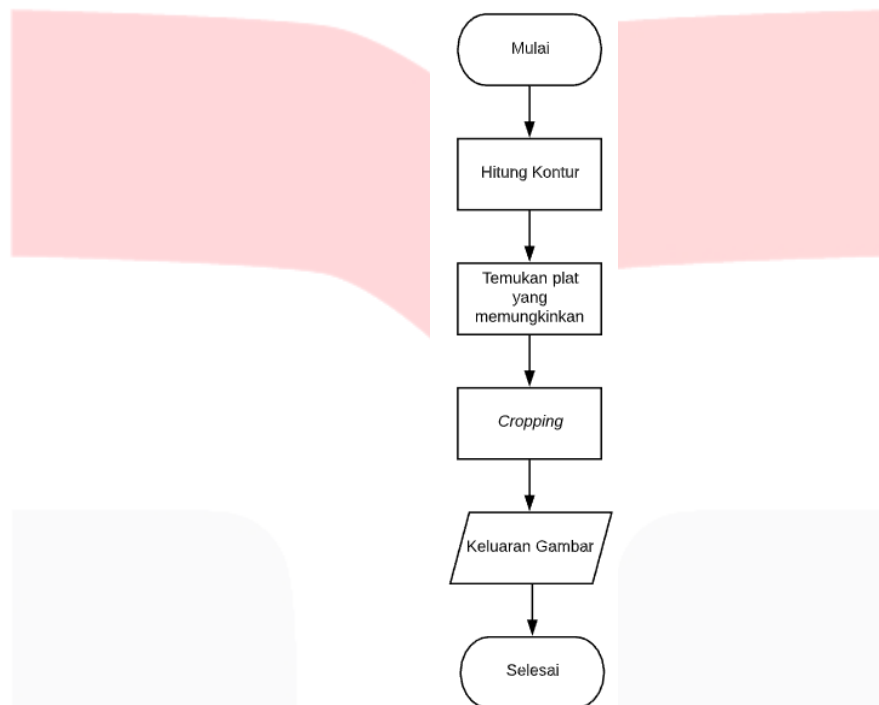
Dimulai dengan tahap mengubah gambar menjadi *grayscale*, citra yang sebelumnya berwarna RGB menjadi diubah HSV untuk mempermudah merepresentasikan warna. Dilanjutkan dengan meningkatkan kontras pada citra *grayscale*. Dalam penelitian ini digunakan operasi morfologis *TopHat* dan *BlackHat*. Operator ini digunakan untuk mengisolasi citra yang dimana lebih terang atau lebih redup daripada tetangga terdekatnya. Menghilangkan *noise* dengan menggaburkan citra. Dalam penelitian ini digunakan dua metode pengaburan sebagai parameter yaitu *Mean filter* dan *Gaussian filter* dengan kernel 5x5. Hasil dari pengaburan gambar diubah menjadi hitam putih dengan dengan metode *threshold* dimana prosesnya didasarkan pada derajat keabuan citra untuk memisahkan antara objek dan *background*. Pada penelitian ini digunakan *adaptive threshold* dikarenakan setiap citra memiliki kondisi pencahayaan yang berbeda di berbagai area. Nilai *threshold* yang akan diuji nantinya adalah  $T1 = 255$  dan  $T2 = 61$ .

### 3.2.2 Deteksi ROI

Pada Gambar 4 merupakan proses menentukan lokasi plat nomor. Hitung semua kontur dan periksa kontur pada citra menggunakan *findContours*. Setiap kontur yang terdeteksi akan diperiksa ukuran panjang, lebar dan rasionya apakah memenuhi ukuran sebagai berikut:

- Luas  $> 80$  & lebar  $> 2$  & panjang  $> 8$
- Aspek rasio (panjang dibagi lebar) antara 0.25-1

Hitung tiga point penting untuk menentukan letak plat nomor menggunakan fungsi *geometric image transformations* yaitu dengan menghitung titik tengah plat dengan menentukan nilai  $x$  dan  $y$  dilanjutkan dengan menghitung tinggi dan lebar plat dan terakhir menentukan sudut dari perhitungan sebelumnya. Setelah didapatkan *region of interest* lakukan *cropping* menggunakan fungsi *getRectSubPix* dengan mengambil nilai titik tengah plat serta tinggi dan lebar plat



Gambar 4. Deteksi ROI

### 3.2.3 Ekstraksi Ciri



Gambar 5. Ekstraksi Ciri

Pada Gambar 5, hasil dari deteksi ROI yang sudah di *crop* dilanjutkan dengan menghitung semua kontur dari gambar. Pada penelitian ini ciri yang diambil adalah kontur dari gambar. Perhitungan jarak antar karakter dengan menggunakan rumus *Euclidean* dan menghitung kemungkinan sudutnya dengan menggunakan *Trigonometri (SIN COS TAN)*. Jika didapatkan dua karakter yang tumpang tindih atau menutup satu sama lain untuk menjadi karakter yang terpisah, pisahkan karakter bagian dalam yang lebih kecil.

### 3.2.4 Klasifikasi KNN

Pada proses ini dilakukan pengenalan terhadap karakter yang sudah terdeteksi melalui proses sebelumnya. Karakter yang terdeteksi di *resize* dengan ukuran 20x30 dan di *reshape* kedalam bentuk 1D *numpy array*, konversi *array* yang sebelumnya integer kedalam bentuk *float*. Tentukan nilai *k* dan hitung jarak ketetanggaan terdekat. Masing-masing kandidat karakter dibandingkan dengan data latih.

## 4. Pengujian dan Analisis Sistem

### 4.1 Pengujian Performasi Sistem

Tabel 4.1 *Gaussian filter*, *threshold = 61*, *k = 3*

No	Plat	Hasil	Benar	Salah	Waktu Eksekusi	Akurasi
1	F1289LP	F1289LP	7	0	13.98ms	100%
2	K4232NK	K4232NK	7	0	14.52ms	100%
3	L1705JA	L1705JA	7	0	14.77ms	100%
4	D3398AAU	D3398AAU	8	0	13.98ms	100%
5	BK2850AND	BK28250AND	9	0	14.35ms	100%
6	BK70NA	BK70NA	6	0	14.71ms	100%
7	A4432VA	A4432VA	7	0	16.26ms	100%
8	B3443TUV	B3443TUV	7	1	20.51ms	87.50%
9	BG4698HG	BG4698HG	7	1	18.27ms	87.50%
10	D3713ZCS	D3713ZCS	8	0	15.90ms	100%
11	N2411HY	N2411HY	6	1	19.87ms	85%
12	B6572PZU	B6572PZU	8	0	15.32ms	100%
13	B2857UFR	B2857UFR	8	0	14.18ms	100%
14	D1745IG	D1745AG	6	1	19.63ms	85%
15	N1059AJ	N1059AJ	7	0	14.72ms	100%
16	B1255SSO	B1255SSO	8	0	13.76ms	100%
17	BK525VG	BK525VG	6	1	18.32ms	85%
18	DM234FA	DM234FA	7	0	17.48ms	100%
19	H8904TP	H8904TP	7	0	15.62ms	100%
20	P5534WD	P5534WD	7	0	14.29ms	100%
21	E1370BQ	E1370BD	6	1	16.38ms	85%
22	E2838YWB	E2838YWB	8	0	15.13ms	100%
23	H5760AMG	H5760AMG	8	0	15.42ms	100%
24	W6823UW	W6823WU	7	0	15.77ms	100%
25	H4660CH	H4660CH	7	0	16.33ms	100%
	<b>Rata-rata</b>					<b>96.6%</b>

Berdasarkan pengujian dengan tiga parameter pertama yang telah dilakukan, didapatkan hasil bahwa parameter *filter Gaussian*, nilai *threshold* sebesar 61, dan nilai *k = 3* adalah parameter optimal dengan tingkat akurasi yang dihasilkan sebesar 96.6%. *Filter Gaussian* lebih baik digunakan daripada *filter Mean* dikarenakan *filter Gaussian* bekerja dengan mendeteksi dan menghitung terlebih dahulu jumlah frekuensi pemakaian tiap warna yang sama, sedangkan *filter Mean* hanya mendeteksi sinyal tinggi dan sinyal rendah saja. Analisis nilai *threshold* diambil dari nilai RGB yang telah diubah ke dalam *grayscale* dengan ketentuan jika nilai *grayscale* dibawah 61 maka nilai tersebut akan diubah menjadi nilai 0 dan jika nilai *grayscale* diatas 61 maka akan diubah menjadi 255. Parameter *k* yang digunakan dalam penelitian ini bernilai *k = 1,3,5*. Ketiganya merupakan bilangan ganjil dikarenakan untuk menghindari munculnya jarak yang sama dalam proses pengklasifikasian. Didapatkan nilai *k* paling optimal adalah *k = 3* dikarenakan dengan nilai *k = 1* hasil yang didapatkan terlalu general untuk mengklasifikasikan 36 kelas, sedangkan dengan nilai *k = 5* tidak mencapai akurasi maksimum dikarenakan terlalu *overfitting* untuk mengklasifikasi 36 kelas.

## 4.2 Pengujian Keandalan Sistem

Tabel 4.2 Intensitas cahaya 1000 – 1500lx, jarak 50cm dan sudut 40°

No	Plat	Hasil	Benar	Salah	Waktu Eksekusi	Akurasi
1	H5439YQ	H5439YQ	7	0	13.87ms	100%
2	K3744MM	K3744MM	7	0	14.81ms	100%
3	H6472JF	H6472JF	7	0	14.88ms	100%
4	H5314ES	H5314ES	7	0	15.17ms	100%
5	G5270GH	G5270GH	7	0	17.41ms	100%
6	H3628GP	H3628GP	7	0	13.59ms	100%
7	K4333RG	K4333RG	7	0	13.75ms	100%
8	H6087AP	H6087AP	7	0	14.68ms	100%
9	H5215RH	H5215RH	7	0	16.12ms	100%
10	Z5737FA	Z5737FA	7	0	13.22ms	100%
11	H3284BH	H3284BH	7	0	13.52ms	100%
12	H3040AUG	H3040AUC	7	1	18.76ms	87%
13	D5919IF	D5919IF	6	1	34.21ms	85%
14	H5916AF	H5916AF	6	1	17.35ms	85%
	<b>Rata-rata</b>					<b>97%</b>

Berdasarkan pengujian yang dilakukan dilapangan dengan parameter intensitas cahaya, jarak dan sudut, didapatkan hasil paling optimal menggunakan intensitas cahaya pada 1000-1500x, jarak 50cm dan kemiringan sudut 40° dengan tingkat akurasi sebesar 97%. Intensitas cahaya sangat berpengaruh dalam pengenalan karakter dikarenakan jika nilai intensitas cahaya rendah maka program kesulitan mendeteksi dimana letak plat berada. Berkaitan dengan jarak, semakin dekat letak objek dengan kamera menyebabkan fokus pada kamera berkurang dimana hal tersebut berpengaruh menambah noise pada citra saat pengambilan gambar, maka diberi jarak renggang sepanjang 50cm.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil pengujian identifikasi plat nomor kendaraan dengan menggunakan metode *K-Nearest Neighbor* didapatkan kesimpulan sebagai berikut:

1. Jarak pengambilan gambar sangat berpengaruh dengan tingkat akurasi yang didapatkan. Dengan pengambilan jarak 40cm, 50cm, dan 60cm didapatkan rata – rata presentase akurasi sebesar 77.5% untuk jarak 40cm, 93,5% untuk jarak 50cm dan 90.5% untuk jarak 60cm.
2. Intensitas cahaya dan kualitas pada saat pengambilan gambar sangat berpengaruh dengan tingkat akurasi yang didapatkan. Dengan nilai intensitas cahaya yang didapatkan 1000-1500lx dan 10-20lx didapatkan rata – rata presentase akurasi sebesar 88.6% untuk nilai intensitas 1000-1500lx dan 85.6% untuk nilai intensitas 10-20lx.
3. Terjadi beberapa kesalahan pada saat pengenalan karakter yang hampir mirip seperti huruf I dan angka 1 dengan presentase kesalahan sebesar 0,09%, huruf B dan angka 8 dengan presentase kesalahan sebesar 0,1% dan beberapa karakter lainnya.
4. Presentase akurasi pengenalan dengan metode K-Nearest Neighbour memiliki hasil yang cukup baik dengan parameter filter Gaussian, nilai threshold 61 dan nilai k = 3 yang memiliki presentase terbaik sebesar 96.6% dan pada keadaan dilapangan memiliki hasil akurasi yang baik yakni sebesar 97% dengan parameter jarak 50cm, sudut 40° dan intensitas cahaya pada 1000-1500lx.

### 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut saran pengembangan dalam penelitian ini:

1. Perlunya pengembangan sistem dengan metode deep learning, sehingga dapat membandingkan hasil dari metode deep learning dan machine learning.
2. Variasi data latih seperti ukuran *font*, bentuk *font*, warna, dan intensitas dari berbagai cahaya yang diharapkan dapat meningkatkan akurasi sistem.
3. Pastikan pencahayaan yang cukup saat pengambilan gambar.



**Daftar Pustaka**

- [1] Surabaya.tribunnews, “Kota Malang Temukan Kerugian Negara Rp 21 M dari Uang Retribusi Parkir,” *surabaya.tribunnews*, 2018. [Online]. Available: <https://surabaya.tribunnews.com/2018/07/14/kejari-kota-malang-temukan-kerugian-negara-rp-21-m-dari-uang-retribusi-parkir>. [Accessed: 06-Aug-2020].
- [2] D. McGavin, B. Stukenborg, and M. Witkowski, “Color figures in BJ: RGB versus CMYK,” *Biophys. J.*, vol. 88, no. 2, pp. 761–762, 2005.
- [3] A. Ayubi, “Teori Penunjang,” *Its*, vol. 1, no. 1, pp. 7–23, 2017.
- [4] Y. Chen and S. Fomel, “Random noise attenuation using local signal and noise orthogonalization,” *77th EAGE Conf. Exhib. 2015 Earth Sci. Energy Environ.*, vol. 80, no. 6, pp. 3582–3586, 2015.
- [5] A. L. PRASASTI, B. IRAWAN, S. E. FAJRI, A. RENDIKA, and S. HADIYOSO, “Perbandingan Ekstraksi Fitur dan Proses Matching pada Autentikasi Sidik Jari Manusia,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 8, no. 1, p. 95, 2020.
- [6] M. N. Aziz, “A Survey on the Implementation of Image Enhancement,” vol. 12, no. 21, pp. 11451–11459, 2017.
- [7] S. Jeyalakshmi and S. Prasanna, “Measuring distinct regions of grayscale image using pixel values,” *Int. J. Eng. Technol.*, vol. 7, no. 1.1, p. 121, 2017.
- [8] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” in *IEEE*, 1979, pp. 62–66.
- [9] G. B. Adrian Kaehler, *Learning OpenCV*. 2008.
- [10] library.binus, “Region of Interest (ROI),” *library.binus.ac.id*. .
- [11] S. M. Awang Hendrianto Pratomo, Wilis Kaswidjanti, “Implementasi Algoritma Region of Interest (ROI) untuk Meningkatkan Performa Algoritma Deteksi dan Klasifikasi Kendaraan,” *jtiik.ub.ac.id*. .
- [12] & Y. Z. Lili, L., Y. Zhang, “K-Nearest Neighbours for Automated Classification of Celestial Objects,” *Sci. China Ser. G-Phys Mech Astron*, 5(7), pp. 916–922, 2008.
- [13] S. P. Dewi, A. L. Prasasti, and B. Irawan, “Analysis of LFCC feature extraction in baby crying classification using KNN,” *Proc. - 2019 IEEE Int. Conf. Internet Things Intell. Syst. IoTaIS 2019*, no. November, pp. 86–91, 2019.