



# TELNECT

Journal homepage: <http://ejournal-purwakarta.upi.edu/telnect>



## Perancangan dan Analisis *Optical Character Recognition* (OCR) Untuk Mencocokkan Pelat Kendaraan Pada *Data Base*

Alan Suryawinata<sup>1</sup>, Dede Irawan Saputra<sup>2\*</sup>

Program Studi Teknik Elektro Universitas Jenderal Achmad Yani, Indonesia <sup>1,2</sup>

\*Corresponding Author: E-mail: [dedeirawan.saputra@lecture.unjani.ac.id](mailto:dedeirawan.saputra@lecture.unjani.ac.id)

### ABSTRACTS

*This study aims to design a system that can detect license plates of motor vehicles that are located at the parking location of PT Tirta Gemah Ripah. The vehicle license plate detection system is designed using the OCR method which combines the haar cascade algorithm, gaussian filter and K-Nearest Neighbor (KNN) and equates it to the vehicle database. KNN who will read the characters on the plate by using a training dataset of 500 characters. System accuracy testing uses several methods including differences in the number of training datasets, differences in the value of K, the elimination of the haar cascade algorithm, and the removal of the gaussian filter algorithm to find the best condition of the system when reading characters on the plates. The haar cascade and gaussian filter algorithm are algorithms that make the readings focused on the plate only and eliminate the object of interference in the image. Research data shows the amount of training data and K values for the KNN algorithm are very important with the results of the system data, the more training data, and the K values, the increase will also increase. The final results of the study using the value of K = 5 has 90.26% for the calculation of the accuracy of each character and 60% for the calculation of the accuracy of the entire plate.*

### ARTICLE INFO

**Article History:**

Received 1 June 2021

Revised 15 June 2021

Accepted 28 June 2021

Available online 30 June 2021

**Keyword:**

Plate recognition  
Optical Character  
Recognition,  
KNN

## 1. PENDAHULUAN

Sepanjang tahun 2018, Polda Jabar berhasil mengungkap 2.421 kasus curanmor. Lokasi yang menjadi sasaran pelaku curanmor mayoritas berada di halaman rumah (753 kasus), jalan umum (268 kasus) dan tempat parkir (192 kasus). Minimnya penjagaan dan lemahnya sistem parkir, dimanfaatkan sindikat pencurian kendaraan bermotor. Bermodal struk parkir dan pelat nomor palsu, komplotan pencuri kendaraan bermotor mengambil mobil dalam kurun waktu 20 menit.

Dalam kasus ini, keberadaan kamera pengawas atau CCTV yang diintegrasikan dengan sistem pendeteksi pelat nomor merupakan perangkat yang dapat digunakan untuk meminimalisir terjadinya penyalahgunaan kendaraan [1]. Pelat nomor polisi kendaraan atau TNKB merupakan identitas utama dari sebuah kendaraan bermotor yang layak jalan/beroperasi di jalan raya, sehingga pelat nomor polisi umumnya diletakkan pada posisi yang mudah dilihat [2]. Dengan sistem pendeteksi pelat nomor, penjaga parkir dapat menyamakan identitas kendaraan dari yang tercetak di karcis parkir dengan yang tersimpan dalam sistem.

Metode pembacaan karakter pada pelat menggunakan metode *connected componen* dan *SVM* [3], metode *edge detection* [4], dan metode *transformasi hough* dan *hit or miss* [5] memiliki tujuan untuk mendeteksi pelat nomor pada kendaraan namun belum mendeteksi setiap karakter yang terdapat pada pelat. Maka dari itu diperlukan metode yang berfungsi untuk membaca karakter pada pelat seperti *Optical Character Recognition (OCR)* [6]. *OCR* adalah metode yang berfungsi untuk membaca citra kemudian merekonstruksi menjadi teks, dan metode *OCR* juga dapat menjadi *support* algoritma tambahan untuk *scanner*[7]. Pada penelitian sebelumnya *OCR* diimplmentasikan pada aplikasi pendeteksi plat nomor negara indonesia menggunakan *OpenCV* dan *Tesseract OCR* pada *Android Studio* memiliki hasil yang baik namun tidak memiliki metode yang meminimalisir *noise* pada gambar [8] dan analisis performansi metode *KNN (K-Nearest Neighbor)* untuk pengenalan karakter pada plat nomor kendaraan di *Raspberry Pi* memiliki hasil yang cukup baik dalam pendeteksian karakter akurasi 77,98% [9].

Pada penelitian penulis menggunakan metode *OCR* dengan algoritma *KNN*, *Gaussian Filter* dan algoritma *Haar Cascade Classification* untuk mendeteksi citra pelat nomor menggunakan kamera sehingga pembacaan pelat dapat terealisasi dan dapat mengkonstruksi citra ke teks. Lokasi di PT Tirta Gemah Ripah (TGR) dapat menjadi contoh kasus implementasi sistem pendeteksi pelat nomor kendaraan bermotor dengan metode *OCR*. Kendaraan di PT TGR terbagi menjadi dua yaitu kendaraan perusahaan dan kendaraan karyawan sedangkan jumlah personil keamanan yang tersedia sangat terbatas dalam penjagaan. Hal tersebut menyebabkan beberapa masalah diantaranya sulitnya mengidentifikasi informasi kepemilikan kendaraan yang keluar dan masuk perusahaan, serta kemungkinan penyalahgunaan kendaraan. Setiap kendaraan di Perusahaan akan dimasukkan kedalam *database* untuk dicocokkan dengan hasil pembacaan karakter pelat nomor untuk identitas kendaraan yang keluar atau masuk.

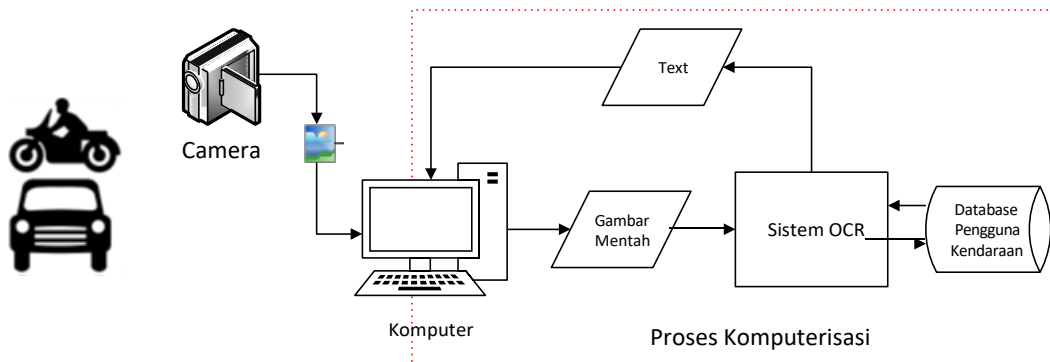
## 2. METODE

### 2.1 Desain dan Model Sistem

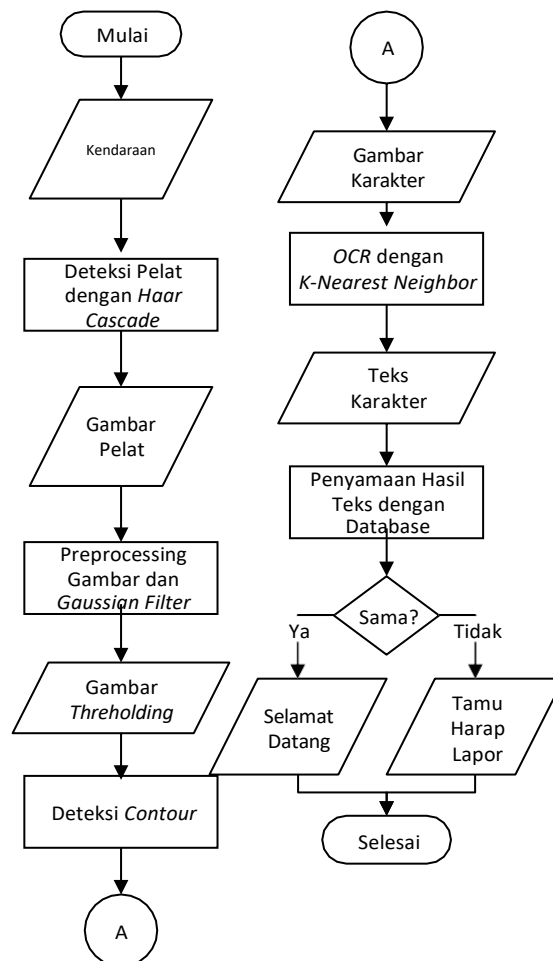
Penjelasan dari diagram blok sistem ditunjukkan pada Gambar 1. Kendaraan merupakan faktor utama dalam data masukan (input). Kendaraan yang dimaksudkan ialah kendaraan yang memiliki TNKB/Pelat Nomer dengan karakter yang menggunakan *font* Indonesia. Kamera akan selalu membaca gambar setiap saat secara live dengan mengirimkan data gambar ke laptop/25ahasa25r. Laptop/25ahasa25r merupakan komponen yang menjadi tempat penyimpanan gambar dari kamera, sistem *OCR* dan *database* kendaraan. Laptop/25ahasa25r akan menjalankan sistem terus menerus. *OCR* merupakan sistem yang digunakan untuk memproses gambar masukan (input) dengan berbagai algoritma dan memiliki keluaran berupa teks yang akan disamakan dengan *database* kendaraan. Cara kerja sistem *OCR* akan dijelaskan dalam Gambar 1 Hasil dari sistem pendeteksi pelat nomer merupakan teks yang dapat dibaca oleh pengguna. Berikut diagram alir cara kerja sistem *OCR* pada Gambar 2.

Diagram alir dari cara kerja sistem *OCR* pada Gambar 2 memiliki penjelasan seperti berikut. Kendaraan merupakan masukan untuk sistem yang dari kamera yang akan direkam dalam bentuk video. Video tersebut akan dibaca oleh sistem dalam bentuk *frame-frame* yang telah diekstraksi yang selanjutnya akan diimplementasikan pada metode yang diterapkan. Terdapat algoritma khusus yang digunakan dengan fungsi yang berbeda dalam sistem yaitu algoritma *haar cascade classification*, *gaussian filter* dan *K-Nearest Neighbor (KNN)* menggunakan 25ahasa pemrograman *python* 3. Algoritma *Haar Cascade classifier* berfungsi untuk mencari kandidat pelat nomor pada seluruh gambar,

pelat yang terdeteksi pada frame video akan dipotong dan menghasilkan keluaran berupa gambar pelat. Gambar pelat yang telah didapatkan oleh sistem akan melalui proses preprocessing untuk merubah warna gambar menjadi grayscale, blur dan bentuk threshold.

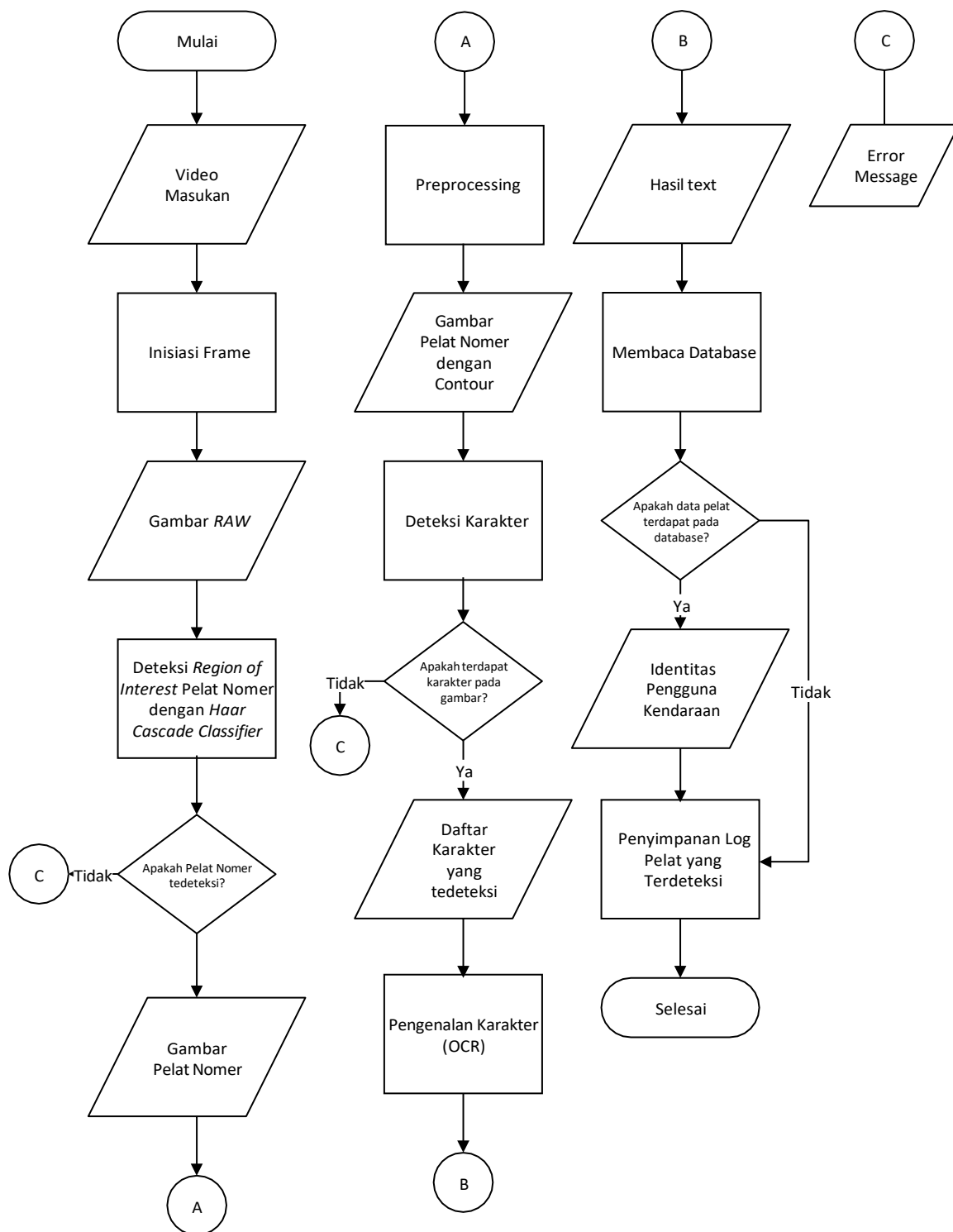


**Gambar 1.** Diagram blok sistem keseluruhan



**Gambar 2.** Diagram alir cara kerja sistem OCR

Algoritma Gaussian Filter merupakan algoritma yang akan digunakan dalam proses blur dan berfungsi untuk menghapus objek kecil yang mengganggu pada gambar (noise) lalu dilakukan proses konversi ke thresholding untuk mendapatkan gambar hitam dan putih. Gambar hitam dan putih akan dideteksi setiap karakternya dengan contour kemudian karakter yang terdeteksi akan dipotong menjadi satu gambar kecil berisikan 1 karakter. Gambar setiap karakter akan dibaca dan disamakan dengan training dataset pada algoritma KNN yang akan memiliki keluaran berupa teks. Teks hasil pembacaan KNN akan disamakan dengan database lalu pelat-pelat yang terdeteksi akan tersimpan dalam rekam jejak sistem. Diagram perancangan sistem pembacaan identitas kendaraan ditunjukkan pada Gambar 3.



Gambar 3. Diagram alir perancangan sistem aplikasi OCR

## 2.2 Pelatihan Data set

Training dataset merupakan data sample yang akan digunakan untuk menyamakan gambar karakter keluaran dari sistem. Training dataset merupakan kumpulan gambar dari berbagai variasi karakter yang akan digunakan dalam sistem OCR. Gambar yang digunakan dalam training data set merupakan 50 gambar pelat dengan format Indonesia dan dibagi menjadi 3 rincian total karakter yaitu training dataset 1 memiliki 72 gambar keseluruhan karakter dan training dataset 2 memiliki 521 gambar keseluruhan karakter. Gambar keluaran dari sistem akan disamakan dengan Training dataset menggunakan algoritma KNN dengan rincian gambar karakter yang ditunjukkan pada Tabel 1 dan 2.

TABEL 1. TRAINING DATASET KARAKTER 0-H

Karakter	Gambar	Jumlah	Karakter	Gambar	Jumlah
0		24	9		20
1		24	A		16
2		24	B		24
3		24	C		10
4		24	D		16
5		24	E		9
6		24	F		12
7		13	G		11
8		24	H		10

TABEL 2. TRAINING DATASET KARAKTER I-Z

Karakter	Gambar	Jumlah	Karakter	Gambar	Jumlah
I		17	R		8
J		9	S		10
K		10	T		9
L		9	U		9
M		9	V		9
N		12	W		13
O		20	X		6
P		16	Y		10
Q		6	Z		6

### 3. HASIL DAN PEMBAHASAN

Deteksi pelat kendaraan merupakan proses pertama yang dilakukan oleh sistem untuk mencari pelat dari keseluruhan objek gambar. Dalam menjalankan sebuah video sistem memerlukan kode untuk menginisiasi setiap frame yang ada dalam video. Hasil keluaran dari inisiasi frame merupakan gambar mentah yang siap diolah oleh sistem. Video yang ditampilkan akan memiliki kotak merah pada setiap framenya yang menandakan bahwa adanya potensi pelat pada frame tersebut yang didefinisikan oleh algoritma cascade classification dan haar classifier. Perintah Algoritma haar feature-based cascade classifier yang digunakan untuk mendeteksi objek adalah detect multiscale. DetectMultiScale berfungsi untuk mendeteksi objek yang memiliki ukuran yang berbeda dengan data yang telah ditraining dalam classifier. Objek yang terdeteksi akan ditampilkan dalam beberapa persegi dalam gambar. Dalam melakukan proses deteksi, ada beberapa tahapan yang harus dilakukan untuk mendapatkan gambar yang hanya berisi gambar pelat nomor.

#### 3.1 Training dan Membaca Data Classifier

Sebelum melakukan proses mendeteksi wilayah pelat dengan cascade classifier, sistem memerlukan data pembandingan untuk mencari pelat agar sistem tidak meleset dalam pemotongan gambarnya. Masukan training data classifier yang berbentuk file .xml, data tersebut akan menjadi pembandingan untuk semua gambar yang memiliki ciri-ciri yang sesuai.

#### 3.2 Deteksi Pelat Kendaraan

Deteksi Pelat Kendaraan dengan algoritma Haar Cascade merupakan proses dimana pelat nomor terdeteksi dari sekian banyak object pada gambar dengan menggunakan fungsi *detect multiscale* yang terdapat pada OpenCV. Terdapat beberapa parameter yang digunakan dalam algoritma tersebut antara lain scalefactor, minNeighbor, dan minSize. scaleFactor merupakan parameter yang menentukan seberapa banyak gambar akan diperkecil setiap perskala gambar. minNeighbor merupakan parameter yang menentukan seberapa banyak tetangga pada setiap persegi dalam penyimpanannya. minSize merupakan parameter yang menetapkan ukuran minimum pada objek yang akan dideteksi, objek dengan ukuran dibawah value akan diabaikan seperti pada Gambar 4.



Gambar 4. Gambar deteksi pelat

### 3.3 Pemotongan Gambar

Setelah region of interest (ROI) gambar telah ditemukan, maka akan dipotong pelatnya saja untuk mempercepat dan meningkatkan akurasi dalam pembacaan angka pada pelat. Apabila pelat tidak dipotong, sistem akan membaca seluruh gambar dan hasil keluaran akan banyak yang tidak sesuai. Algoritma pemotongan pelat yang digunakan adalah numpy arrays (slicing). Numpy Arrays (Slicing) merupakan Teknik pemotongan berdasarkan data array yang terdapat pada objek, objek dalam sistem ini ialah kotak merah hasil Region of Interest (ROI). Berikut hasil dari pemotongan pelat dari gambar kendaraan yang ditunjukkan pada Gambar 5.



Gambar 5. Hasil pemotongan gambar

#### 1. *Preprocessing* Gambar Pelat

*Preprocessing* gambar memiliki beberapa tahapan dalam pengolahan gambarnya untuk mencari keluaran berupa gambar grayscale, gambar blur dan gambar thresholding yang diperlukan untuk proses KNN. Berikut adalah beberapa tahapan yang harus dilakukan dalam proses *pre-processing*:

##### a. Konversi gambar pelat nomor ke gambar *grayscale*.

Dalam tahap ini citra input berwarna dalam RGB diubah menjadi citra *grayscale*. Tujuan dari tahap ini adalah untuk mereduksi ukuran data. Dalam teknik ini data yang akan digunakan dalam proses pengenalan adalah intensitas cahaya dari sebuah citra sehingga kombinasi warna *Red*, *Green*, dan *Blue* tidak dibutuhkan. Proses pengubahan citra RGB menjadi citra *grayscale* adalah sebagai berikut: Misalkan suatu citra mata memiliki nilai:  $R = 152,75$   $G = 132$   $B = 133,81$  Maka nilai *grayscale* dari citra tersebut dapat dihitung seperti di bawah ini:

$$w = (0,2989 \times 152,75) + (0,5870 \times 132) + (0,1140 \times 133,81) = 138,395$$

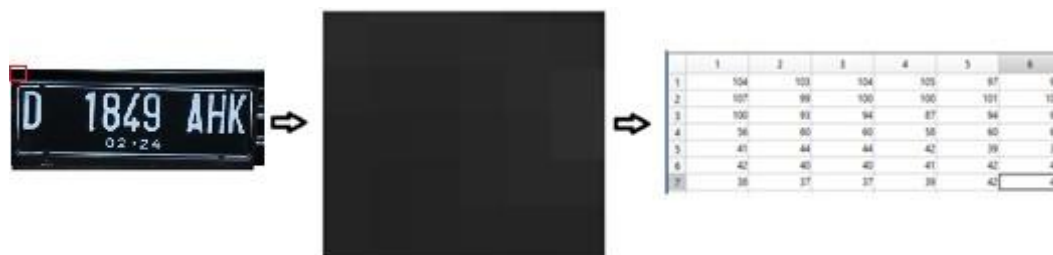
Hasil gambar yang telah dikonversikan bentuk warnanya menjadi *greyscale* ditunjukkan pada Gambar 6.



Gambar 6. Hasil konversi gambar RGB ke *grayscale*

b. Menghilangkan *noise* gambar dengan mengaburkan (*blur*) gambar.

Metode pengaburan (*blur*) yang digunakan yaitu *gaussianblur*. Dengan *gaussianblur*, wilayah yang akan dikaburkan hanya wilayah samping dari gambar dan *noise-noise* pada gambar akan dikaburkan seperti *background*. Berikut contoh perhitungan proses *gaussian blur* seperti pada Gambar 7.



Gambar 7. Ekstraksi citra pelat 6 x 7 pixel

Pada proses ini akan digunakan nilai deviasi = 1 dan kernel sebesar 5x5, didapat kernel distribusi *gaussian 2-D* seperti yang ditunjukkan pada Gambar 8.

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Gambar 8. Distribusi *gaussian 2-D*

Langkah berikutnya adalah melakukan *image filtering* dengan menerapkan operasi *gaussian* seperti berikut:

$$h(m, n) = \frac{[(1 \cdot 104) + (4 \cdot 103) + (7 \cdot 104) + (4 \cdot 105) + (1 \cdot 97) + (4 \cdot 107) + (16 \cdot 99) + (26 \cdot 100) + (16 \cdot 100) + (4 \cdot 101) + (7 \cdot 100) + (26 \cdot 93) + (41 \cdot 94) + (26 \cdot 87) + (7 \cdot 94) + (4 \cdot 56) + (16 \cdot 60) + (26 \cdot 60) + (16 \cdot 58) + (4 \cdot 60) + (1 \cdot 41) + (4 \cdot 44) + (7 \cdot 44) + (4 \cdot 42) + (1 \cdot 39)]}{273} = 84$$

Dan hasil dari proses gaussian filtering ditunjukkan pada Gambar 8.

The diagram shows the result of Gaussian filtering. On the left, a black square represents the blurred 6x7 pixel region. An arrow points to a 7x7 grid representing the filtered image values.

	1	2	3	4	5	6
1	104	103	102	101	100	100
2	100	98	97	96	96	96
3	87	85	84	83	84	84
4	65	65	65	64	64	64
5	48	49	49	48	48	47
6	41	41	41	42	42	42
7	38	38	38	40	41	43

(a)



(b)

Gambar 8. (a) Hasil perhitungan gaussian filtering (b) Gambar hasil gaussian filtering  
c. Thresholding

Selanjutnya adalah tahap *thresholding* yang digunakan untuk mengubah gambar hasil *grayscale* menjadi gambar biner. Nilai *Threshold* dihitung dengan membagi nilai hasil *grayscale* pada tahap sebelumnya dengan nilai jumlah derajat keabuan (0 sampai 255 = 256) dibagi dengan 256 (nilai derajat keabuan yang diinginkan). Proses penghitungan nilai *threshold* untuk citra mata seperti yang ditunjukkan pada persamaan 1.

$$x = \frac{w}{b} \tag{1}$$

Keterangan:

x = nilai pembandingan *threshold*

w = nilai hasil *grayscale*

b = 256/a (a = 256)

Proses mengubah citra RGB menjadi citra biner menggunakan aturan sebagai berikut: (1) Jika nilai piksel citra  $\geq x$  maka nilai piksel menjadi 1, (2) Jika nilai piksel citra  $\leq x$  maka nilai piksel menjadi 0. Berikut proses pengubahan citra RGB menjadi citra biner ditunjukkan pada Gambar 9. Gambar hasil pengubahan citra *grayscale* menjadi biner ditunjukkan pada Gambar 10.

Citra asli								Citra biner							
121	159	205	88	71	103	231	134	0	1	1	0	0	0	1	0
211	177	67	54	81	92	233	146	1	1	0	0	0	0	1	1
183	246	199	56	61	191	211	113	1	1	1	0	0	1	1	0
98	72	45	153	215	123	222	100	0	0	0	0	1	1	1	0
123	111	232	152	63	44	119	211	0	0	1	1	0	0	0	1
186	191	124	45	111	100	206	99	1	1	0	0	0	0	1	0

Gambar 9. Contoh proses pengubahan citra grayscale menjadi citra biner



Gambar 10. Hasil thresholding

Tujuan dari konversi gambar pelat menjadi berbentuk biner adalah untuk proses pembacaan karakter yang akan menggunakan *contour*. *Contour* merupakan teknik yang berfungsi untuk membaca garis keluaran gambar dengan menyambungkan titik-titik yang memiliki nilai RGB yang sama.

Hasil dari Teknik *contour* berupa garis yang mengikuti bentuk karakter pada pelat. Setiap kontur yang terdeteksi akan disimpan sebagai satu gambar karakter yang telah dipotong dan akan disamakan dengan *training dataset* yang ada pada *database* menggunakan algoritma KNN seperti yang ditunjukkan pada Gambar 11.



Gambar 11. Hasil deteksi contour

Setelah contour pada karakter terdeteksi maka sistem akan memberi tanda pada setiap karakter yang terbaca dan akan digunakan untuk disamakan dengan *training dataset* menggunakan algoritma

KNN. Berikut gambar dari setiap karakter yang terbaca dan ditandai oleh sistem seperti pada Gambar 12.



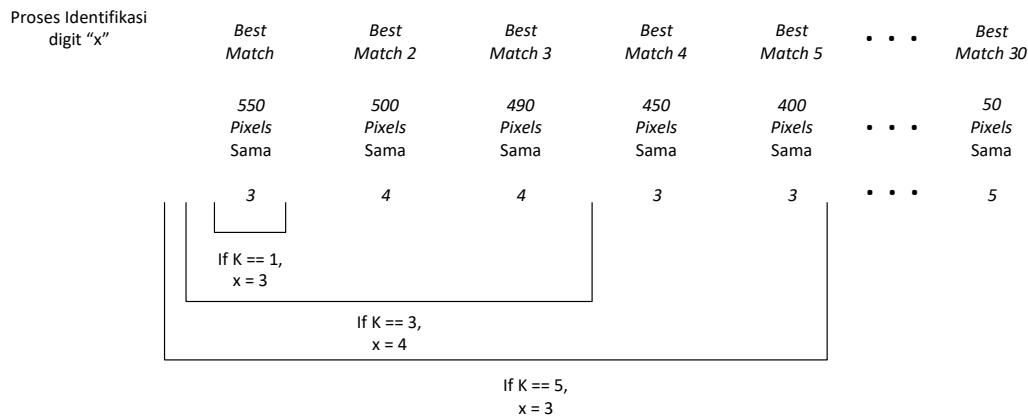
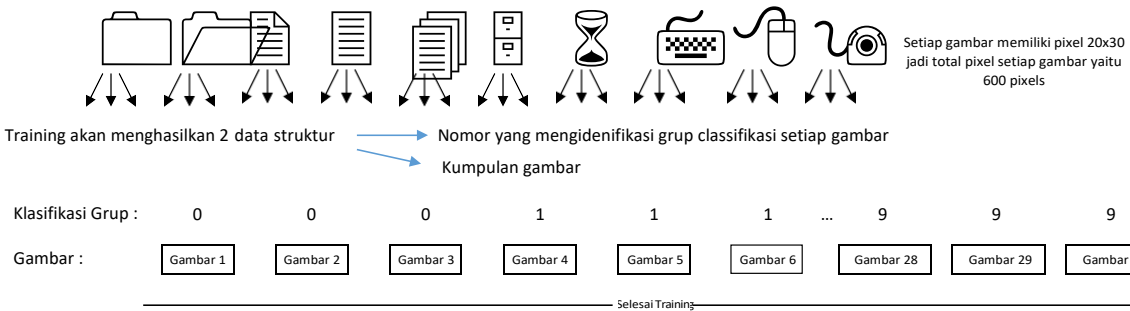
Gambar 12. Contoh hasil karakter yang terbaca

d. KNN

Mekanisme algoritma KNN yang terdapat pada sistem ditunjukkan pada Gambar 13.

Cara Kerja KNN

Sebagai contoh digit 0 – 9 akan memiliki 3 gambar training setiap digit maka diperlukan 30 gambar training



Gambar 13. Cara kerja KNN pada sistem

Algoritma KNN pada sistem bekerja dengan menyamakan nilai pixel yang terdapat pada gambar masukan dengan semua nilai pixel yang terdapat pada training dataset. Semakin banyak pixel yang sama, maka kemungkinan besar digit tersebut akan menjadi teks keluaran sistem. Nilai K akan menentukan jumlah digit yang akan diambil sebagai pertimbangan sistem menyimpulkan keluaran. Dan hasil dari penyamaan tersebut akan menjadi teks keluaran serta karakter pada gambar pun akan dipotong dan dinamakan sesuai teksnya seperti pada Gambar 14.



Gambar 14. Hasil pemotongan dan pembacaan karakter

Hasil yang diperoleh setelah menerapkan algoritma Haar-like Cascade classifier dan K-Nearest Neighbor dalam melakukan metode OCR, diperoleh tingkat akurasi yang beragam dari setiap digit karakter pelat (A sampai Z dan 0 sampai 9) yang terdeteksi. Presentase keberhasilan pendeteksian karakter pada pelat terpengaruhi oleh beberapa faktor antara lain posisi (kemiringan dan jarak) dari pelat, warna pelat, dan intensitas cahaya pada pelat. Semua faktor tersebut dapat merubah bentuk karakter yang terdapat pada pelat menjadi karakter lain dan membuat penyamaan antara contour gambar dan contour training dataset berbeda dari seharusnya.

Rekognisi atau pengenalan karakter teks pada pelat merupakan salah satu bentuk pengujian yang ada pada sistem. Ketepatan pengenalan karakter teks pada pelat dapat dilihat dari keluaran video hasil implementasi masing-masing algoritma. Persentase akurasi sistem akan diukur dengan 2 cara yaitu membandingkan karakter yang terdeteksi benar pada pelat dengan karakter yang ada pada pelat dan membandingkan jumlah setiap karakter yang terdeteksi dengan total jumlah setiap karakter. Membandingkan karakter yang terdeteksi pada pelat berfungsi untuk melihat bentuk pelat seperti apa yang dapat mendeteksi karakter dengan baik. Dan membandingkan jumlah karakter berfungsi untuk melihat karakter seperti apa yang dapat terdeteksi pada sistem dengan Training dataset yang telah disediakan. Berikut cara menghitung akurasi perbandingan antara karakter pada pelat:

$$\text{Akurasi Keseluruhan Pelat} = \frac{JPB}{JPK} \times 100\% \quad (2)$$

Keterangan:

JPB = Jumlah Pelat Benar

JPK = Jumlah Pelat Keseluruhan

$$\text{Akurasi Kesamaan Karakter} = \frac{JDB}{JDK} \times 100\% \quad (3)$$

Keterangan

JPB = Jumlah Digit Benar

JPK = Jumlah Digit Keseluruhan

*Training Dataset* memiliki total 521 gambar karakter dengan nilai K = 5, hasil dari pembacaan keseluruhan pelat yaitu 15 pelat terdeteksi dengan benar dari 25 pelat maka akurasinya 60%.

$$\text{Akurasi Keseluruhan Pelat} = \frac{15}{25} \times 100\% = 60.00\%$$

*Hasil* dari kesamaan karakter yang terbaca benar yaitu 165 karakter dari 183 karakter keseluruhan maka akurasinya 90.26%.

$$\text{Akurasi Kesamaan Karakter} = \frac{165}{183} \times 100\% = 90.26\%$$

Kesimpulan yang dapat diambil dari penggunaan nilai K = 5 adalah sistem mengambil 5 karakter terdekat sebagai pertimbangan dalam menyimpulkan teks karakter sehingga akurasi kesamaan karakter cukup tinggi. Pada akurasi keseluruhan pelat tidak semua karakter sesuai dengan karakter sesungguhnya dan apabila terdapat 1 karakter yang berbeda tetap akan disebut gagal.

Tabel 3 merupakan hasil akurasi dari pembacaan pelat menggunakan training dataset yang memiliki keseluruhan gambar yang telah didapatkan dengan nilai K sebesar 5. Nilai 5 merupakan jumlah maksimum K yang digunakan oleh sistem karena karakter Q, X dan Z memiliki 6 gambar pada training dataset dan nilai K harus ganjil. Nilai K = 5 akan sangat berpengaruh terhadap hasil keluaran sistem karena nilai K menentukan jumlah karakter dengan nilai pixel yang sama paling banyak yang akan diambil sistem dalam menyimpulkan teks keluaran. Hasil ini merupakan hasil tertinggi yang penulis dapatkan dan faktor yang menjadi sebab kegagalan dalam Tabel 4.4 yaitu walaupun banyak karakter yang terbaca dengan benar namun apabila ada 1 karakter yang berbeda maka pembacaan keseluruhan pelat akan tetap disebut gagal.

Terdapat beberapa pelat yang masih tetap gagal walaupun sudah menggunakan 5 metode pembacaan seperti pelat D6060R, B4707THE, D3572SAC, D4508MW, D2426SA, dan lain-lain. Untuk pelat D6060R terbaca gagal karena huruf pertama dari pelat yaitu D dilipat dan tidak dapat terambil oleh kamera, Pelat D3572SAC memiliki penutup pelat yang berwarna putih, cover berbahan akrilik dan menghadap sedikit keatas membuat karakter tidak dapat dibaca dengan benar karena cover berwarna putih membuat karakter seperti menempel dengan cover serta bahan akrilik dan menghadap ke atas membuat intensitas cahaya pada pelat cukup tinggi yang membuat background menjadi putih. Dan

untuk pelat lain gagal karena kurangnya karakter pada training dataset sehingga terdapat 1 atau 2 karakter yang memiliki hasil teks keluaran yang berbeda.

**TABEL 3.** HASIL AKURASI PEMBACAAN *TRAINING DATASET* DENGAN KNN = 5 PADA DATA BASE

No	Original	Terdeteksi	Hasil Deteksi	Karakter Sama	Jumlah Digit	Waktu Rekognisi (s)
1	D1031AEU	D1031AEU	Berhasil	8	8	0.7514
2	D6060R	6060R	Gagal	5	6	0.8118
3	D1213AGN	D1213AGN	Berhasil	8	8	0.8866
4	D1549IG	D1549IG	Berhasil	7	7	0.6832
5	D1162ADH	D1162ADH	Berhasil	8	8	0.9395
6	D1849AHK	D1849AHK	Berhasil	8	8	0.4149
7	D2655TP	D2655TP	Berhasil	7	7	0.5206
8	B4707THE	B470YTNE	Gagal	6	8	0.4927
9	D3682JX	D3682JX	Berhasil	7	7	0.5076
10	D3572SAC	3572	Gagal	4	8	0.6393
11	D6302ABW	D6302ABW	Berhasil	8	8	0.7111
12	D5757RD	05757RD	Gagal	6	7	0.6718
13	R2464BF	R2464BF	Berhasil	7	7	0.6663
14	R5944SS	R5944SS	Berhasil	7	7	0.5206
15	D5065JM	D5065JM	Berhasil	7	7	0.7450
16	D4508MW	D4508MU	Gagal	6	7	0.7211
17	D2426SA	D2T26S7	Gagal	5	7	0.8268
18	D5882VEB	D588ZNE6	Gagal	5	8	0.6722
19	D3904ZTO	D3904ZTO	Berhasil	8	8	0.7813
20	D3298GO	D3298GO	Berhasil	7	7	0.3810
21	D3693WY	D3693WY	Berhasil	7	7	0.8188
22	D2808MO	02808O	Gagal	5	7	0.5027
23	D3007OK	03007OK	Gagal	6	7	0.6832
24	B313EEK	B33EEK	Gagal	6	7	0.8278
25	D1371SQ	D1371SQ	Berhasil	7	7	0.5316

#### 4. KESIMPULAN

Dari hasil pembahasan dapat ditarik kesimpulan bahwa solusi terbaik dalam melakukan pendeteksian karakter pada gambar pelat menggunakan metode OCR di PT TGR yang sudah diimplementasikan antara lain perancangan sistem yang dapat mendeteksi pelat nomor kendaraan bermotor dengan metode OCR menggunakan algoritma *haar cascade*, *gaussian filter* dan KNN dapat diimplementasikan. Penggunaan *haar cascade* membuat objek yang dideteksi oleh sistem terfokuskan ke pelat, *gaussian filter* dapat menghapuskan noise yang terdapat pada gambar untuk meminimalisir adanya tambahan karakter lain atau merubah bentuk karakter dan KNN merupakan algoritma yang menyamakan bentuk karakter terdeteksi dengan training dataset, semakin banyak *training dataset* maka semakin banyak variasi karakter yang didapatkan. Sistem mampu menyamakan teks keluaran hasil metode OCR dengan data pada *database* serta mencatat *log* keluaran ke file teks dokumen.

Akurasi dari pembacaan keseluruhan karakter pada setiap pelat dengan nilai  $K=5$  memiliki hasil akurasi sebesar 60% dalam pendeteksian pelat keseluruhan dan 90.26% dalam pembacaan karakter. Namun tidak semuanya berhasil karena apabila terdapat 1-2 karakter yang berbeda maka tetap akan dikatakan gagal dan untuk akurasi pembacaan setiap karakter memiliki hasil terbaik apabila training dataset memiliki karakter banyak variasi dengan memadukan nilai  $K$  yang tinggi serta adanya tambahan algoritma yang membuat pembacaan terfokus ke pelat saja.

Dalam pendeteksian pelat nomor dalam keseluruhan gambar selain menggunakan haar cascade classifier terdapat algoritma lain seperti *connected component*, *edge detection*, *transformasi hough* atau *hit or miss* yang dapat diimplementasikan pada penelitian berikutnya serta kombinasi algoritma yang dapat mengurangi noise atau gangguan pada gambar selain menggunakan gaussian filter terdapat algoritma lain seperti *mean filtering* atau *median filtering*.

## 5. REFERENSI

- [1] D. Gunawan, W. Rohimah, and R. F. Rahmat, "Automatic Number Plate Recognition for Indonesian License Plate by Using K-Nearest Neighbor Algorithm," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 648, no. 1, doi: 10.1088/1757-899X/648/1/012011.
- [2] Y. Mansyur, "Optical Character Recognition Untuk Deteksi Pelat Mobil Dan Motor Kendaraan Pada Kampus Teknik Gowa," *Skripsi. Univ. Hasanuddin*, 2018.
- [3] A. Budianto, T. B. Adji, and R. Hartanto, "Deteksi Nomor Kendaraan Dengan Metode Connected Component Dan Svm," *J. Teknol. Inf. Magister Darmajaya*, vol. 1, no. 01, pp. 106–117, 2015.
- [4] P. P. Khare, R. Dudhe, A. Chungade, and A. Naykinde, "Advanced License Number Plate Recognition System," *Int. J. Eng. Res. Technol.*, vol. 3, no. 06, pp. 1–9, 2015, [Online]. Available: [www.ijert.org](http://www.ijert.org).
- [5] Y. Puspito, F. X. A. Setyawan, and H. Fitriawan, "Deteksi Posisi Plat Nomor Kendaraan Menggunakan Metode Transformasi Hough Dan Hit Or Miss," *Electr. - J. Rekayasa dan Teknol. Elektro*, vol. 12, no. 3, 2018.
- [6] A. P. Wahyu and Suhendri, "Peningkatan Sistem Keamanan Parkir dengan Teknologi Artificial Intelligence Imaging," *J. Inf. Technol.*, vol. 01, no. 02, pp. 39–42, 2019.
- [7] K. Kusumawati and D. W. Cahyadi, "Penerapan Teknologi Optical Character Recognition Untuk Mendeteksi Plat Nomor Kendaraan," *Pros. Semin. Nas. Inov. Teknol.*, pp. 12–20, 2017.
- [8] S. Wira, "Aplikasi Pendeteksi Plat Nomor Negara Indonesia Menggunakan Android Studio," *Nasional*, no. December, p. 6, 2017, [Online]. Available: [https://www.researchgate.net/publication/321797435\\_Aplikasi\\_Pendeteksi\\_Plat\\_Nomor\\_Negara\\_Indonesia\\_Menggunakan\\_Android\\_Studio](https://www.researchgate.net/publication/321797435_Aplikasi_Pendeteksi_Plat_Nomor_Negara_Indonesia_Menggunakan_Android_Studio).
- [9] N. A. Tauchid, "Analisis Performansi Metode KNN (K-Nearest Neighbor) Untuk Pengenalan Karakter Pada Plat Nomor Kendaraan di Raspberry Pi," Telkom University, 2015.