

Pengenalan Plat Kendaraan Secara Waktu Nyata Menggunakan *FRAMEWORK AFORGE.NET*

Relung Satria D.^{*)}, Rizal Isnanto, and Ajub Ajulian Zahra

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}Email : relungsatria@gmail.com

Abstrak

Teknologi pengenalan sekarang ini sudah sangat berkembang. Salah satu aplikasi teknologi pengenalan yang menarik untuk dikembangkan adalah pengenalan plat kendaraan. Pengenalan plat kendaraan ini dapat dilakukan secara waktu nyata. Pengenalan ini menggunakan sebuah rangkaian metode. Sebelum citra dikenali, dilakukan prapengolahan. Prapengolahan ini terdiri dari pengubahan citra menjadi aras keabuan, pendeteksian tepi dan pengambangan. Setelahnya dilakukan penentuan lokasi plat pada citra. Metode yang digunakan untuk penentuan lokasi plat ini adalah analisis intensitas vertikal dan horizontal citra. Langkah berikutnya adalah memisahkan karakter pada plat menggunakan algoritma komponen terkoneksi dan membaca karakter pada plat tersebut dengan menggunakan algoritma K-Nearest Neighbors. Dari hasil pengujian, dapat disimpulkan bahwa sistem pengenalan plat kendaraan waktu nyata yang diujikan terhadap 30 data uji menghasilkan tingkat keberhasilan sebesar 83,33% pada $K = 1$.

Kata kunci : deteksi tepi, pengambangan, analisis intensitas, algoritma komponen terkoneksi, K-Nearest Neighbors.

Abstract

Recognition technology nowadays has been so advanced. One of its interesting application that can be developed is vehicle plate recognition. This recognition technology can run in real time. Vehicle plate recognition itself uses a series of methods. Before the system recognizes an image, it does some preprocessing. This preprocessing consist of grayscaleing, edge detection and thresholding and afterward vehicle plate on the image is being located. This system uses vertical and horizontal intensity analysis from the image to locate vehicle plate. Next step is character segmentation. It is done by using connected component algorithm. And finally, K-Nearest Neighbors algorithm reads all these character. The test results show that this vehicle plate recognition system provides 83,33% success rate from 30 test data.

Keywords : edge detection, thresholding, intensity analysis, connected component algorithm, K-Nearest Neighbors.

1. Pendahuluan

Pengenalan Plat Kendaraan merupakan salah satu aplikasi pengolahan citra yang memiliki banyak manfaat. Seperti sekuritas, pengidentifikasian kendaran di tempat parkir ataupun pada saat trafik. Tujuan dari penelitian ini adalah agar komputer dapat mengenali dan membaca plat kendaraan secara otomatis dalam waktu nyata. Untuk mengubah informasi dari dunia nyata menjadi informasi yang dapat dimengerti oleh komputer dibutuhkan metode khusus. metode yang dapat digunakan juga cukup beragam, tapi untuk mendapatkan hasil secara waktu nyata dibutuhkan metode yang cepat dan handal.

Beberapa penelitian telah dilakukan terkait dengan pengenalan plat kendaraan. Salah satunya menggunakan analisis intensitas citra untuk menentukan lokasi plat serta jaringan syaraf tiruan untuk pembacaan karakter^[1].

Kekurangan jaringan syaraf tiruan sendiri adalah membutuhkan pelatihan yang lama dan data yang cukup banyak sebelum jaringan tersebut dapat mengenali karakter. Meskipun begitu jaringan syaraf tiruan dinilai cukup cepat untuk mengenali karakter. Selain itu juga terdapat penelitian yang memanfaatkan horizontal vector density untuk penentuan lokasi plat^[2].

Metode pengenalan plat kendaraan yang diusulkan terbagi menjadi empat tahap, yaitu prapengolahan citra, penentuan lokasi plat pada citra, segmentasi karakter pada plat, dan pembacaan plat kendaraan. Pada prapengolahan terdapat pengubahan citra menjadi aras keabuan, deteksi tepi dan pengambangan. Dilanjutkan dengan penentuan lokasi citra dengan analisis intensitas dan segmentasi citra menggunakan algoritma komponen terkoneksi, dan terakhir pembacaan citra dengan menggunakan algoritma K-Nearest Neighbors.

2. Metode

2.1 Aras Keabuan

Aras keabuan mengacu pada sebuah gambar yang hanya memiliki ranah warna abu-abu atau keabuan. Pada gambar digital, sebuah gambar terdiri dari piksel-piksel kecil yang tiap pikselnya diwakili oleh nilai-nilai warna. Dalam RGB sebuah gambar digital terbagi menjadi 3 kanal warna, yaitu kanal Merah (*Red*), kanal Hijau (*Green*) dan kanal biru (*Blue*). Menggabungkan kanal-kanal warna ini dalam sebuah gambar menciptakan sebuah gambar berwarna utuh.

Dalam Aras keabuan, pada intinya 3 nilai kanal warna ini dijumlah menjadi satu buah nilai. Algoritma umum pada aras keabuan gambar adalah:

1. Mendapatkan nilai merah, hijau dan biru dari tiap piksel.
2. Gunakan perhitungan matematika untuk mengubah nilai-nilai tersebut menjadi sebuah nilai abu-abu.
3. Ganti nilai-nilai asli merah, hijau, biru dengan nilai abu-abu yang baru.

Formula umum dari perhitungan aras keabuan adalah;

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3 \quad (1)$$

Pada intinya jumlah nilai-nilai tersebut tidak boleh melebihi 100%.

Selain itu juga terdapat formula lain untuk perhitungan Aras keabuan yaitu BT-709, yang terkadang disebut Luma. Formula dari BT-709 itu adalah

$$\text{Gray} = (\text{Red} * 0.2126 + \text{Green} * 0.7152 + \text{Blue} * 0.0722)^{[3]} \quad (2)$$

2.2 Deteksi Tepi

Definisi dari tepi adalah perubahan yang signifikan secara lokal pada suatu citra. Tepi sendiri sering muncul di batas antar dua area yang berbeda pada citra. Tujuan dari deteksi tepi sendiri adalah memberikan gambaran garis dari citra, yang dapat menghasilkan ciri yang penting untuk kemudian diekstraksi. Ciri tersebut dapat digunakan untuk algoritma komputasi yang lebih tinggi.

Beberapa hal dapat menyebabkan perubahan intensitas. Batas pada objek, perbedaan kedalaman objek, perbedaan warna pada permukaan objek, batas permukaan, bayangan, dan pantulan cahaya.^{[3][5]}

Pendeteksian tepi sendiri dilakukan dengan cara mencari nilai maksimum dari perbedaan antara piksel yang berada di 4 arah di sekeliling piksel yang sedang di proses. Seumpama terdapat 3x3 elemen citra yang akan diproses, dan x adalah piksel yang sedang ditinjau dan p1 hingga p8 adalah piksel tetangga disekeliling piksel yang sedang ditinjau, maka dapat digambarkan

$$\begin{bmatrix} p1 & p2 & p3 \\ p8 & [x] & p4 \\ p7 & p6 & p5 \end{bmatrix} \quad (3)$$

Dengan,

x = piksel yang ditinjau

p1 – p8 = piksel tetangga.

Dan hasil dari piksel yang sedang diproses dapat dirumuskan dengan

$$\max(|p1 - p5|, |p2 - p6|, |p3 - p7|, |p4 - p8|) \quad (4)$$

2.3 Pengambangan

Pengambangan dalam pengolahan citra digital digunakan untuk memisahkan latar depan dan latar belakang citra dengan menggunakan citra masukan paling sedikit satu warna atau aras keabuan untuk mendefinisikan batas. Pengambangan sering digunakan sebagai langkah awal dalam pemrosesan citra untuk mencapai tujuan tertentu. Kegunaan segmentasi dari Pengambangan sendiri dapat untuk memisahkan satu objek dengan objek lainnya. meskipun begitu, pencahayaan, noise, intensitas warna dapat mempengaruhi hasil dari Pengambangan itu sendiri, karena itu dibutuhkan nilai Pengambangan yang tepat untuk mendapatkan hasil yang baik.

Cara yang paling sederhana untuk melakukan Pengambangan adalah dengan menentukan sebuah nilai Ambang. Semua nilai piksel yang sama atau lebih dari nilai Ambang akan diubah menjadi piksel hitam, sedang semua nilai piksel yang dibawah nilai Ambang diubah menjadi piksel putih.^{[6][7]}

2.4 Proyeksi Vertikal dan Horisontal

Pada citra dapat dilakukan sebuah perhitungan statistik berdasarkan intensitas gambar. Perhitungan ini dapat disebut proyeksi atau analisis intensitas. Proyeksi sendiri terdiri dari Proyeksi Vertikal dan Proyeksi Horisontal. Proyeksi vertikal merupakan jumlah semua magnitude berdasarkan posisi y pada gambar. Proyeksi horisontal merupakan jumlah semua magnitude berdasarkan posisi x pada gambar.

$$Px(X) = \sum_{j=0}^{h-1} f(x, j) \quad (5)$$

$$Py(Y) = \sum_{i=0}^{w-1} f(y, i) \quad (6)$$

Dengan,

Px(X) = nilai proyeksi dari posisi x pada citra.

Py(Y) = nilai proyeksi dari posisi y pada citra.

h = tinggi citra.

w = lebar citra.

f(x,j) = nilai piksel pada koordinat x,j

f(y,i) = nilai piksel pada koordinat y,i

2.5 Algoritma Komponen Terkoneksi

Algoritma komponen terkoneksi adalah sebuah algoritma yang berguna untuk mendeteksi region atau area yang

terdapat pada citra biner. Algoritma ini bekerja dengan 2 kali meninjau seluruh data pada citra 2 dimensi. Tinjauan pertama bertujuan untuk memberikan label sementara pada setiap pikselnya dan dicatat. Tinjauan kedua bertujuan untuk mengganti label sementara tadi dengan label terkecil dari kelas yang sama.

Algoritma ini mengecek konektifitas setiap piksel tetangga dari piksel yang sedang diproses. Piksel tetangga ini meliputi piksel pada diagonal kiri, diagonal kanan, horisontal dan vertikal dari piksel yang sedang diproses, atau biasa disebut 8-konektifitas. Alur dari pemberian label pada tinjauan pertama adalah sebagai berikut

1. Tinjau tiap piksel dari citra
2. Apabila piksel yang sedang ditinjau bukan piksel latarbelakang maka dilakukan pengecekan tetangga dari piksel tersebut.
3. Apakah piksel tetangga sudah mempunyai label, jika iya maka beri label pada piksel yang sedang ditinjau sesuai dengan label utama dari piksel tetangga, jika tidak, maka beri label baru pada piksel yang sedang ditinjau.
4. Jika terdapat label berbeda pada piksel tetangga maka berikan label utama pada piksel tetangga tersebut. Label utama didapat dengan mengambil nilai label terkecil.

Setelah itu dilakukan tinjauan kedua yang berguna untuk mengubah semua label sementara ke label utama yang telah diberikan.^{[6][8]}

2.6 K-Nearest Neighbors

K-Nearest Neighbors adalah sebuah algoritma yang berfungsi untuk melakukan klasifikasi data berdasarkan kedekatan lokasi data tersebut dengan data yang lain. Pada algoritma K-NN, data berdimensi q, jarak dari data tersebut ke data lain dapat dihitung. Nilai jarak tersebut digunakan sebagai nilai kedekatan antara data uji dan data latih. K pada K-NN berarti K-data terdekat dari data uji. Jika K bernilai 1, kelas dari satu data latih sebagai tetangga terdekat (terdekat pertama) dari data uji tersebut akan diberikan sebagai kelas untuk data uji, yaitu kelas 1. Jika K bernilai 2, akan diambil 2 tetangga terdekat dari data latih. Begitu pula dengan nilai K 3, 4 dan seterusnya. Jika dalam K-tetangga terdapat dua kelas yang berbeda, akan diambil kelas dengan jumlah data terbanyak (pengambilan suara mayoritas). Jika kelas dengan data terbanyak ada dua atau lebih, akan diambil kelas dari data dengan jumlah yang sama tersebut secara acak.

Algoritma prediksi pada K-NN dapat dijelaskan dengan

1. $Z = (x', y')$, adalah data uji dengan vektor x ; dan label kelas y' yang belum diketahui.
2. Hitung jarak $d(x; x)$, jarak diantara data uji z ke setiap vektor data latih, simpan dalam D
3. Pilih $D_z \subseteq D$, yaitu K tetangga terdekat dari z
4. $Y' = \arg \max_{(x_i, y_i) \in D_z} I(v = y_i)$

Masalah yang dihadapi dalam K-NN adalah pemilihan nilai K yang tepat. Cara pengambilan suara mayoritas dari K-tetangga untuk nilai K yang besar dapat mengakibatkan distorsi data yang besar.^[9]

Untuk menangani masalah pengambilan suara mayoritas tersebut biasanya ditambahkan penggunaan bobot untuk menghitung kandidat kelas yang sebaiknya diambil oleh data uji dari K-tetangga terdekat. Bobot dari setiap tetangga terdekat dihitung dengan formula

$$W_i = \frac{1}{d(x', x_i)^2} \quad (7)$$

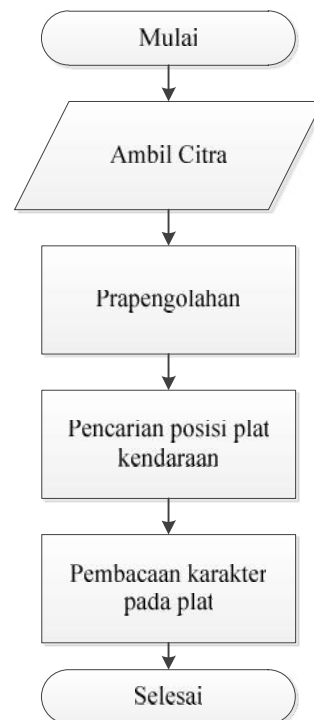
Formula yang bisa digunakan adalah^[9]

$$Y' = \arg \max \sum (x_i, y_i) \in D_z w_i x I(v = y_i) \quad (8)$$

3. Hasil dan Analisa

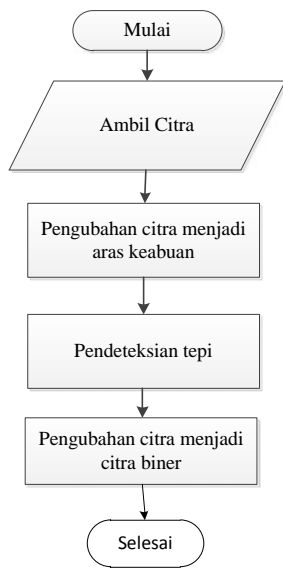
3.1 Algoritma Program

Pada sistem pengenalan plat ini data masukan adalah citra bergerak dari webcam atau dari berkas. Secara umum pembuatan program ini mengikut alur sesuai yang ditunjukkan dalam Gambar 1.



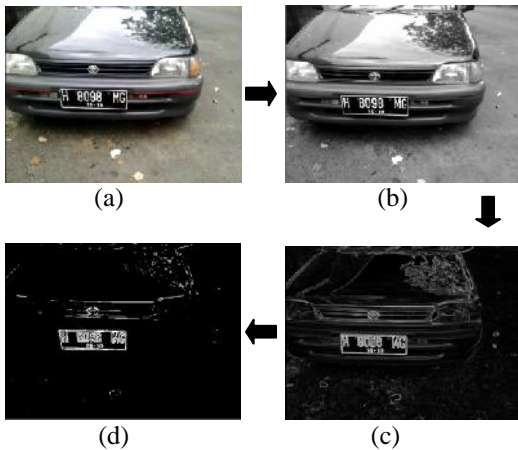
Gambar 1. Diagram alur sistem pengenalan plat

Tahap pertama dari sistem ini adalah prapengolahan. Prapengolahan ini terdiri dari beberapa proses yaitu mengubah citra menjadi aras keabuan, lalu penerapan pendeteksian tepi pada citra keabuan, dan terakhir thresholding untuk mengubah citra menjadi citra biner. Diagram alur dapat dilihat pada Gambar 2.



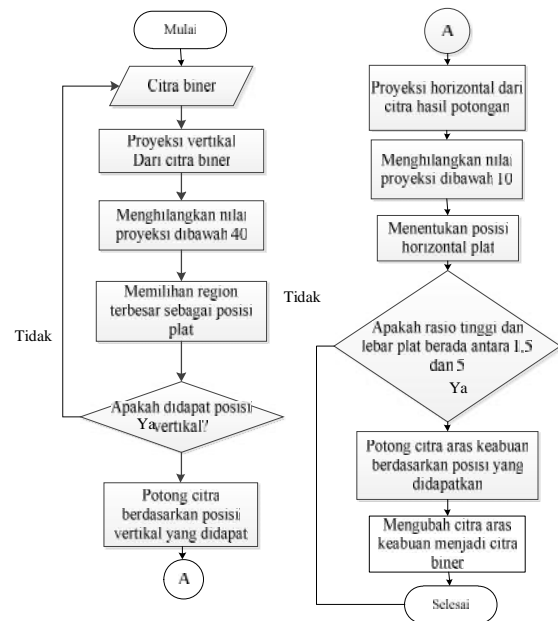
Gambar 2. Diagram alur prapengolahan

Ukuran yang digunakan adalah 640×480 piksel, karena dianggap sebagai ukuran yang cukup untuk mempertahankan kualitas citra untuk dilihat pengguna (*user*) serta cukup untuk pengolahan lanjut. Ilustrasi prapengolahan dapat dilihat pada gambar 3.



Gambar 3. (a) citra asli (b) citra aras abu-abu (c) citra hasil deteksi tepi (d) citra hasil pengambangan

Setelah prapengolahan dilanjutkan dengan pencarian posisi plat kendaraan. Metode yang digunakan untuk pencarian posisi plat kendaraan adalah menganalisa proyeksi horizontal dan proyeksi vertikal dari citra hasil prapengolahan. Diagram alur dapat dilihat pada Gambar 4.



Gambar 4. Diagram alur penentuan lokasi plat

Proyeksi vertikal didapatkan dengan cara menjumlahkan piksel putih tiap kolom pada citra biner hasil prapengolahan lalu hasilnya ditampilkan dalam bentuk grafik. Proyeksi horizontal pun sama, menjumlahkan piksel putih tiap baris pada citra biner hasil prapengolahan lalu hasilnya ditampilkan dalam bentuk grafik. Gambar 5 menunjukkan grafik hasil proyeksi vertikal. Bagian yang terdapat pada kotak merah merupakan bagian yang menunjukkan lokasi plat kendaraan.



Gambar 5. Bagian grafik proyeksi vertikal yang menunjukkan lokasi plat

Dari grafik hasil proyeksi vertikal dapat dianalisis untuk mendapatkan posisi vertikal dari plat kendaraan. Hal ini dilakukan dengan menapis nilai-nilai grafik hasil proyeksi vertikal untuk menghilangkan nilai poin yang kurang dari 40. Lalu setelah itu akan didapatkan beberapa region yang menunjukkan nilai diatas 40, dan kemudian dilakukan pemilihan region yang terbesar dari kumpulan region tadi, karena kemungkinan tertinggi plat kendaraan berada pada region terbesar. Pada Gambar 6 region terbesar ditunjukkan oleh lingkaran merah.



Gambar 6. Grafik intensitas setelah ditapis

Setelah itu dilakukan pemotongan citra berdasarkan posisi vertikal dari region terbesar tadi dan kemudian dianalisis kembali secara horizontal dari hasil potongan citra tersebut. Analisis dilakukan dengan melihat nilai dari proyeksi horizontal potongan citra tadi. Nilai tersebut ditapis untuk menghilangkan nilai dibawah 10. Kemudian diambil posisi horizontal pertama dari hasil tapis tadi sebagai posisi plat secara horizontal, seperti pada Gambar 7.



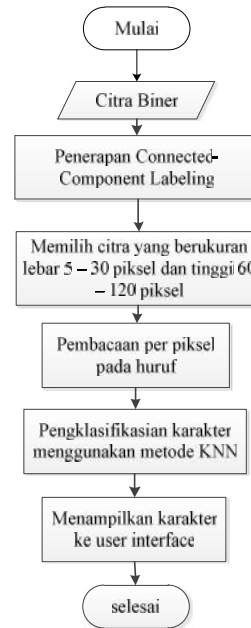
Gambar 7. Hasil pemotongan citra

Lalu kemudian di cek apakah rasio tinggi dan lebar dari hasil penentuan lokasi plat tadi berada antar 1.5 dan 5, jika iya maka dilakukan pemotongan citra aras keabuan hasil prapengolahan berdasarkan posisi plat yang telah didapatkan, kemudian citra aras keabuan tersebut diubah menjadi citra biner, seperti ditunjukkan pada Gambar 8.



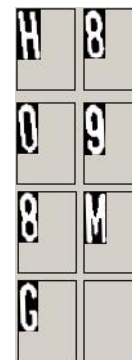
Gambar 8. Citra biner plat kendaraan.

Pembacaan karakter plat dilakukan per karakter, karena itu sebelum pembacaan dilakukan segmentasi pada plat untuk memisahkan citra menjadi per karakter. Diagram alur pembacaan plat dapat dilihat pada Gambar 9.



Gambar 9. Diagram alur pembacaan karakter

Segmentasi ini dilakukan dengan menggunakan metode *Connected-Component Labeling* atau biasa disebut *blob extraction*. *Blob* ini berfungsi memisahkan area-area yang terhubung pada citra. Dan hanya mengambil area yang memiliki lebar 5 - 30 piksel dan tinggi 30 - 65 piksel. Hasil segmentasi karakter dapat dilihat pada Gambar 10.



Gambar 10. hasil pemisahan tiap karakter

Lalu akan didapatkan citra masing-masing karakter pada plat dan dilakukan pengubahan ukuran citra karakter menjadi 10x10 piksel. Kemudian citra karakter tersebut dibaca per pikselnya. Hasil pembacaan tadi digunakan sebagai ciri dari karakter tersebut. Ciri karakter tadi kemudian dijadikan masukan untuk fungsi *k-Nearest Neighbors* (KNN), ciri tersebut dihitung dengan ciri data latih untuk kemudian diklasifikasikan ciri tersebut termasuk sebagai karakter apa. Setelah itu menampilkan

hasil pembacaan karakter tadi ke *User Interface*. Gambar 11 menunjukkan hasil pembacaan yang ditampilkan pada UI.



Gambar 11. Hasil pembacaan plat kendaraan

3.2 Pengujian Sistem

Pengujian pada sistem pengenalan plat kendaraan ini dilakukan dengan menggunakan data video dari 30 kendaraan yang berbeda. Parameter awal untuk pengujian sistem dapat dilihat pada tabel 1.

Tabel 1. Parameter pengujian

Parameter	Value
K	1
Ukuran frame	640x480

$$\% \text{Keberhasilan} = \frac{\sum \text{Benar}}{\sum \text{Data}} \times 100 \%$$

Hasil pengujian dari sistem ini dapat dilihat pada Tabel 2

Tabel 2. Hasil pengenalan plat nomor pada K = 1

No.	Gambar Plat	Nomor terbaca	Pengujian
1.		H9069GR	Benar
2.		AA7362DE	Benar
3.		H8367FS	Benar
4.		H8402EF	Benar
5.		H8529FE	Benar
6.		H8575HY	Benar
7.		H8583FF	Benar
8.		H8742DE	Benar
9.		H8427TR	Benar
10.		H8563GY	Benar
11.		K8479JF	Benar
12.		H8474LC	Benar

13.		H8635PR	Benar
14.		G8849FB	Benar
15.		K8947ZA	Benar
16.		R9027WB	Benar
17.		H8098MG	Benar
18.		Tidak Terbaca	Salah
19.		B2886YH	Benar
20.		H7835KS	Benar
21.		H8723HF	Benar
22.		H333YS	Benar
23.		Tidak Terbaca	Salah
24.		Tidak Terbaca	Salah
25.		H8984MB	Benar
26.		H9142vs	Benar
27.		H861ZZR1	Salah
28.		19187N	Salah
29.		K8813CB	Benar
30.		G8888YA	Benar

$$\% \text{Keberhasilan} = \frac{25 \text{ Citra yang benar}}{30 \text{ total citra yang diuji}} \times 100\% = 83,33\%$$

Hasil pengujian dengan K = 1 menghasilkan tingkat keberhasilan sebesar 83,33%.

3.3 Variasi Parameter

Table 3. Variasi parameter

Parameter	Value
K	3, 5, 7, 9
Ukuran frame	1280x720

Pengujian yang sama juga dilakukan dengan variasi nilai K. Dengan nilai K = 3, 5, 7, 9. Didapatkan hasil seperti Tabel 4.

Tabel 4. Hasil pengenalan plat dengan variasi nilai K

K	3	5	7	9
Hasil	53,33%	10%	0%	0%

Dilakukan pula pengujian perbandingan ukuran frame citra 640x480 dan 1280x720 dengan 10 data uji dan K = 1. Dengan catatan bahwa sistem ini saat perancangan dan

dibangun awal menggunakan ukuran frame 640x480. Tabel 5 merupakan hasil perba

Tabel 5. Hasil pengenalan plat dengan variasi ukuran frame

Ukuran frame	640x 480	1280x720
Hasil pengujian	100 %	40 %

3.4 Analisis Terhadap Hasil Pengujian

Kegagalan yang terjadi pada saat pembacaan citra terjadi karena beberapa faktor.



Gambar 12. Citra plat

Kasus 1

Yang pertama adalah karena bayangan. Pada Gambar 12, dapat dilihat pada huruf H dan R terjadi kesalahan pemotongan yang menyebabkan kedua huruf tersebut menjadi salah pembacaannya. Penyebab dari kesalahan pemotongan ini berasal dari plat yang kurang mendapat cahaya karena terkena bayangan mobil, seperti contoh yang ditunjukkan pada Gambar 13.

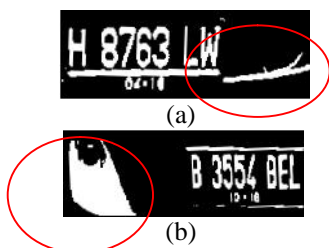


Gambar 13. Citra asli saat pengujian

Saat terjadi pengambangan, plat menjadi tidak begitu jelas dan hasil pembacaan intensitas horisontal menjadi kurang baik yang akhirnya menyebabkan kesalahan pemotongan plat.

Kasus 2

Yang kedua, Pantulan-pantulan cahaya yang terang pada badan mobil juga dapat mengacaukan penentuan lokasi plat. Seperti ditunjukkan pada Gambar 14



Gambar 14. Citra plat

Pada Gambar 14(a) terjadi pantulan cahaya yang terang pada bumper mobil. Sedang pada Gambar 14(b) terjadi pantulan cahaya terang pada lampu mobil. Semua pantulan cahaya ini berpengaruh sekali pada hasil penentuan lokasi plat.



Gambar 15. Citra saat prapengolahan

Intensitas pantulan cahaya yang tinggi akan memberikan informasi salah pada saat pendeteksian tepi dan pengambangan, seperti yang ditunjukkan pada Gambar 15.

Kasus 3

Yang ketiga adalah nilai K. Nilai K sangat berpengaruh pada hasil pembacaan karakter. Nilai K yang semakin besar tidak berarti pembacaan semakin bagus. Semakin besar nilai K semakin besar pula distorsi data. Sehingga pembacaan pun menjadi kacau. Nilai K yang efektif untuk melakukan pembacaan adalah 1. Ukuran frame juga mempengaruhi hasil pembacaan. Hal ini dikarenakan detail konfigurasi pada sistem tidak sesuai untuk frame 720p. selain itu terjadi lonjakan waktu perhitungan pada frame 720p lebih dari dua kali lipat. Hal ini dikarenakan semakin banyak piksel yang harus dihitung.

4. Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa pengenalan plat kendaraan waktu nyata sangat mungkin dilakukan. Untuk mencapai hal itu dilakukan beberapa langkah utama, yaitu, prapengolahan penentuan lokasi plat, segmentasi karakter pada plat, kemudian pembacaan karakter. Metode yang diusulkan menggunakan analisis intensitas untuk penentuan lokasi plat, algoritma komponen terkoneksi untuk segmentasi karakter, dan *k-nearest neighbors* untuk pembacaan karakter. Didapatkan hasil sebesar 83,33% untuk $K = 1$. Meskipun hasil yang didapat tidak memuaskan, tapi tetap menunjukkan sistem ini dapat digunakan untuk pengenalan plat kendaraan.

Referensi

- [1]. Martinsky, Ondrej. *Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems*. 2007. B.S.C. Thesis. BRNO University of Technology.

- [2]. Othman Khalifa, Sheroz Khan, Rafiqul Islam, Ahmad Suleiman. *Malaysian Vehicle License Plate Recognition*. 2007. International Islamic University Malaysia.
- [3]. Gonzalez, Rafael C. *Digital Image Processing Third Edition*. 2008. Pearson Education International.
- [4]. Helland, Tanner. *Seven Grayscale Conversion Algorithm (With Pseudocode and VB6 Sourcecode)*.
- [5]. Trucco, Chapt 4 AND Jain et al., Chapt 5
- [6]. Putra, Darma. *Pengolahan Citra Digital*. 2010. Penerbit ANDI. Yogyakarta.
- [7]. Russ, John C. *The Image Processing Handbook Fifth Edition*. 2007. Taylor & Francis Group. USA.
- [8]. Luigi Di Stefano, Andrea Bulgarelli. *A Simple and Efficient Connected Components Labeling Algorithm*. University of Bologna
- [9]. Prasetyo, Eko. *Data Mining: Konsep Dan Aplikasi Menggunakan MATLAB*. 2012. Penerbit ANDI. Yogyakarta.