

Ekstraksi Karakter Plat Nomor Kendaraan Menggunakan Google Vision dan Regex Pattern

Vehicle License Plate Character Extraction Using Google Vision and Regex Patterns

Condro Wibawa ^{*1}, Dessy Tri Anggraeni ²

Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Gunadarma, Indonesia

*E-mail : condro_wibawa@staff.gunadarma.ac.id ^{*1}, dessytri@staff.gunadarma.ac.id ²*

**Corresponding author*

Received 16 April 2025; Revised 29 April 2025; Accepted 30 April 2025

Abstrak

Di era serba AI (*Artificial Intelligence*) seperti saat ini, penggunaan teknologi *Optical Character Recognition* (OCR) adalah hal yang lumrah. Salah satunya adalah pada *Manless Parking System* atau sistem parkir otomatis. Pada sistem ini teknologi OCR digunakan untuk mendeteksi plat nomor kendaraan secara otomatis. Namun begitu tidak semua metode memberikan hasil yang optimal. Pada penelitian-penelitian sebelumnya rata-rata tingkat akurasi dari penerapan OCR adalah sebesar 90%. Pada penelitian ini digunakan metode OCR dari Google yaitu Google Vision dan Pattern Regex untuk mendeteksi plat nomor kendaraan pada citra digital. Google Vision dipilih karena kemampuannya yang relatif lebih baik dibandingkan metode OCR lainnya seperti Tesseract dan EasyOCR. Sedangkan Regex Pattern digunakan untuk melakukan filtering terhadap teks hasil ekstraksi OCR yang tidak diperlukan. Penelitian ini diujikan terhadap 150 data sample berformat JPG dan JPEG. Ukuran file bervariasi antara 50 KB – 2 MB. Sudut pengambilan gambar juga dibuat bervariasi. Hasil penelitian menunjukkan bahwa penggunaan metode OCR dengan Google Vision dan Regex Pattern memberikan hasil yang sangat baik dengan tingkat akurasi mencapai 99,50%, jauh lebih baik dibandingkan penelitian sebelumnya. Hasil penelitian ini bisa dijadikan landasan penerapan teknologi OCR pada pembacaan plat nomor kendaraan, khususnya pada *Manless Parking System*. Sehingga diharapkan dapat meningkatkan efisiensi dan keamanan sistem.

Kata Kunci : Plat Nomor Kendaraan, Optical Character Recognition, Google Vision, Regex Pattern

Abstract

In the current era of Artificial Intelligence (AI), the use of Optical Character Recognition (OCR) has become increasingly common, particularly in applications such as Manless Parking Systems. These systems utilize OCR to automatically detect vehicle license plates. However, previous studies have shown that the average accuracy rate of OCR implementations is around 90%. This research applies Google's OCR technology, Google Vision, combined with Regex Pattern filtering, to improve license plate detection accuracy in digital images. Google Vision was selected for its superior performance compared to other OCR methods like Tesseract and EasyOCR, while Regex Pattern was used to eliminate irrelevant extracted text. The method was tested on 150 image samples in JPG and JPEG formats, with varying file sizes (50 KB – 2 MB) and image angles. The results demonstrate that the integration of Google Vision OCR with Regex Pattern yields excellent performance, achieving an accuracy rate of 99.50%, significantly outperforming previous approaches. These results can be used as a basis for the application of OCR technology in reading vehicle license plates, especially in the Manless Parking System. So it is expected to increase system efficiency and security.

Keywords : Vehicle License Plates, Optical Character Recognition, Google Vision, Regex Pattern

1. PENDAHULUAN

Teknologi *Optical Character Recognition* atau lebih dikenal dengan OCR merupakan teknologi yang mampu membaca atau mentranslasi karakter teks pada citra digital menjadi bentuk karakter digital [1] [2]. Teknologi ini sering digunakan pada berbagai bidang seperti pada pembacaan plat nomor kendaraan untuk parkir [3], ekstraksi dokumen digital [4], ekstraksi dokumen hasil pindai [5] dan lain-lain.

Penggunaan teknologi OCR pada deteksi plat nomor kendaraan mulai banyak diterapkan terutama di area parkir yang menerapkan sistem *Manless Parking System* atau sistem parkir tanpa bantuan manusia sebagai operator. *Manless parking system* menggunakan teknologi OCR untuk mengidentifikasi kendaraan yang masuk dengan bantuan kamera. Hasil pembacaan ini kemudian dicatat di dalam basis data dan akan dicocokkan kembali ketika kendaraan keluar dari area parkir. Permasalahan utama dari sistem ini adalah keakuratan dan kecepatan sistem dalam membaca plat nomor kendaraan. Kesalahan pembacaan plat nomor akan menimbulkan beberapa masalah. Masalah utama adalah palang pintu parkir tidak bisa terbuka jika hasil pembacaan plat nomor ketika masuk area parkir tidak sesuai dengan hasil pembacaan ketika keluar. Masalah lainnya adalah waktu pembacaan yang lama akan menimbulkan antrian parkir.

Permasalahan ini bisa diantisipasi dengan menerapkan metode yang tepat diantaranya adalah dengan melakukan *preprocessing* citra terlebih dahulu [6]. Penerapan metode *preprocessing* citra pada konteks OCR sudah pernah dibahas pada jurnal "Perbaikan Citra Dokumen Hasil Pindai Menggunakan Metode Simple, Adaptive-Gaussian, dan Otsu Binarization Thresholding" dan "Comparison of Image Segmentation Method in Image Character Extraction *Preprocessing* Using Optical Character Recognition". Dari kedua jurnal ini dapat disimpulkan bahwa penerapan metode *preprocessing* pada teknologi OCR Tesseract memberikan hasil yang lebih buruk dibandingkan langsung dengan citra aslinya [7] [8]. Sedangkan metode pembacaan OCR juga sudah pernah dibahas pada jurnal "Penerapan Teknologi OCR Plat Nomor Untuk Meningkatkan Efisiensi dan Keamanan Akses Kendaraan". Pada jurnal ini metode yang digunakan adalah EasyOCR tanpa filtering dan menghasilkan akurasi sebesar 72% [9]. Pada penelitian "Deteksi Plat Nomor Kendaraan Otomatis Dengan Convolutional Neural Network dan OCR Pada Tempat Parkir ITB Ahmad Dahlan Lamongan" menghasilkan akurasi sebesar 88% [10]. Penelitian lain yaitu "Implementasi Algoritma YOLO dan Tesseract OCR Pada Sistem Deteksi Plat Nomor Otomatis" menghasilkan tingkat akurasi yang lebih tinggi yaitu 92,3% [11]. Sedangkan pada penelitian "Implementasi Teknologi OCR Untuk Mendeteksi Nomor Plat Mobil" menyinggung jarak pengambilan gambar, dimana jika jaraknya lebih dari 80cm tingkat akurasinya mencapai 88% [12].

Pada penelitian kali ini akan digunakan metode OCR yang relatif jarang digunakan yaitu Google Vision. Namun meskipun kurang populer, dibandingkan Tesseract dan EasyOCR, akurasi Google Vision tergolong tinggi. Hal ini terlihat pada penelitian yang berjudul "Comparative Analysis of Google Vision OCR with Tesseract on Newspaper Text Recognition". Pada penelitian ini OCR menggunakan Google Vision lebih unggul dengan tingkat akurasi 97,89% dibandingkan Tesseract dengan tingkat akurasi 90,16%. Namun begitu, Google Vision memiliki kelemahan yaitu kecepatan ekstraksi yang lebih lambat yaitu rata-rata per dokumen 30,08 detik dibandingkan metode Tesseract sebesar 12,96 detik per dokumen [13].

Sementara itu proses *preprocessing* yang akan dilakukan adalah deteksi tepi menggunakan metode Canny dan Contour menggunakan library OpenCV. Pada proses *preprocessing* ini tidak dilakukan binerisasi citra secara tersendiri, mengingat hasil yang kurang baik pada penelitian-penelitian sebelumnya [7] [8].

2. METODE PENELITIAN

Tahapan penelitian disusun seperti pada gambar 1 di bawah ini, agar penelitian dapat berjalan dengan terstruktur.



Gambar 1. Tahapan penelitian

2.1 Studi Pustaka

Studi pustaka dilakukan dengan mencari informasi terkait metode-metode yang akan digunakan pada penelitian ini. Sumber pustaka sebagian besar diambil dari jurnal yang membahas penelitian-penelitian serupa sebelumnya.

Metode Canny

Metode Canny merupakan metode yang dikenal sebagai operator deteksi tepi optimal karena tingkat kesalahannya rendah. Jarak piksel yang ditemukan pada deteksi tepi Canny sangat pendek sehingga menghasilkan satu jawaban dan satu tepi [14]. Pada penelitian ini juga menunjukkan hasil dimana metode Canny lebih baik dibandingkan metode Sobel. Pada jurnal “Implementasi Metode Canny untuk Deteksi Tepi Mutu Daun Tembakau” dikatakan bahwa metode Canny memiliki tiga kriteria yaitu *good detection*, *good localization*, dan *only one response to a single* [15]. Ketiga kriteria ini menjadikan hasil deteksi tepi Canny relatif lebih baik dibandingkan metode lainnya.

Metode Contour

Kontur (contour) adalah suatu kumpulan point atau titik yang didapatkan dari proses komputasi yang mewakili bentuk dari suatu batas objek pada suatu citra yang nantinya dihubungkan dengan garis dari setiap titik atau point yang telah didapat, sehingga menghasilkan kontur dari objek tersebut. Kontur mirip dengan tepi (*edge*) hanya saja lebih detil, tegas dan jelas [16]. Pada penelitian ini contour digunakan untuk memperjelas bentuk obyek/gambar.

Teknologi OCR

Teknologi OCR (*Optical Character Recogintion*) adalah teknologi yang mampu membaca atau mentranslasi karakter teks pada citra digital menjadi bentuk karakter digital [1]. Sementara itu menurut laman resmi Google Cloud (2025), Pengenalan Karakter Optik (OCR) adalah teknologi dasar yang melandasi konversi teks ketik, tulisan tangan, atau cetak dari gambar menjadi teks yang diencode mesin [17].

Dalam dunia digital *Optical Character Recognition* (OCR) adalah salah satu topik yang menarik dan menantang dalam pengenalan pola [6]. Pengenalan karakter dicapai melalui banyak proses meliputi : Input Image, Optical Scanning, Location Segmentation, *Pre-Processing*, *Segmentation*, *Representation*, *Feature Extraction*, *Training and Recognition*, *Post-Processing*, *Output Text* : proses segmentasi, fitur ekstraksi dan klasifikasi [6].

Dalam kehidupan sehari-hari, teknologi OCR banyak digunakan untuk berbagai kebutuhan khususnya yang berkaitan dengan teknologi AI [6]. Beberapa diantaranya adalah ekstraksi teks pada dokumen, deteksi plat nomor, nomor induk kependudukan, dan lain-lain.

Google Vision

Google Vision adalah teknologi kecerdasan buatan (AI) yang dapat mengenali dan memahami konten visual dalam gambar dan video. Google Vision dapat diakses melalui Google Cloud Vision API, yang merupakan layanan cloud yang memungkinkan pengembang untuk menganalisis dan memproses gambar secara otomatis. Google Cloud Vision API adalah API (REST dan RPC) yang siap digunakan dan memungkinkan developer mengintegrasikan fitur deteksi penglihatan umum dengan mudah dalam aplikasi, termasuk pelabelan gambar, deteksi wajah dan struktur, pengenalan karakter optik (OCR), dan pemberian tag konten vulgar [17].

Google Vision adalah layanan berbayar. Meski demikian, Google memberikan layanannya secara gratis untuk penggunaan maksimal 1000 kali selama sebulan [17]. Merujuk berbagai macam penelitian [11] [12] [13] kemampuan Google Vision dalam melakukan ekstraksi karakter memiliki keakuratan yang cukup tinggi, bahkan di atas 97% . Tingkat akurasi ini lebih baik dibandingkan teknologi OCR lainnya seperti Tesseract [13].

Regex

Regex (Reguler Expression) adalah ekspresi/format/bentuk yang digunakan untuk mencocokkan pola teks [18]. Penggunaan Regex sangat luas seperti pada pencocokan nilai pada protokol jaringan, format penomoran, bahkan DNA [19]. Pada Python, regex bisa digunakan dengan menggunakan library `re` . Beberapa fungsi yang sering digunakan pada *library* ini adalah `findall`, `search`, `split`, dan `sub` [20].

2.2 Preprocessing Citra : Edge Detection dan Contour

Proses preprocessing citra dilakukan dengan menerapkan dua langkah filtering yaitu Edge Detection menggunakan metode Canny dan segmentasi menggunakan metode Contour. Penerapan filtering Edge Detection bertujuan untuk mendeteksi tepi obyek yang dalam hal ini adalah karakter pada plat nomor. Akan tetapi pada edge detection tidak ada kesinambungan antar titik sehingga bentuk obyek kurang jelas. Agar obyek terlihat lebih jelas sebagai satu kesatuan maka ditambahkan filtering menggunakan metode contour. Kedua langkah *preprocessing* ini dilakukan menggunakan library OpenCV.

2.3 Ekstraksi Karakter Dengan Google Vision

Langkah selanjutnya adalah mengekstraksi karakter pada citra plat nomor yang sudah dilakukan *preprocessing* . Proses ekstraksi ini menggunakan teknologi OCR dengan memanfaatkan metode dari Google yaitu Google Vision.

2.4 Penerapan Regex Pattern

Seringkali di dalam plat nomor kendaraan terdapat karakter lain seperti nama kendaraan, tanggal aktif plat nomor, dan lain-lain. Teknologi OCR akan membaca semua karakter ini, sehingga output yang dihasilkan kurang sesuai. Untuk itu hasil ekstraksi perlu difilter lagi. Salah satu cara untuk menentukan teks plat nomor dalam sebuah kumpulan teks adalah dengan

membandingkan dengan pola/format yang sudah baku. Untuk melakukan proses ini digunakan fungsi regex. Fungsi regex digunakan untuk memberikan pattern/format, sehingga teks yang diambil hanyalah yang sesuai format saja.

Seperti kita tahu bahwa plat nomor di Indonesia memiliki format yang baku yaitu :

- a. Blok Huruf I
Diawali dengan blok huruf dari A-Z yang bisa terdiri dari 1 atau 2 karakter
- b. Spasi
- c. Blok Angka
Karakter angka dari 0-9 yang terdiri dari 4 karakter.
- d. Spasi
- e. Blok Huruf II
Diakhiri dengan blok huruf dari A-Z yang bisa terdiri dari 1 hingga 3 karakter

Contoh plat nomor di Indonesia adalah sebagai berikut : F 3363 MP, B 2036 KKY, dan lain-lain.

2.5 Analisa Hasil

Langkah terakhir yang akan dilakukan adalah analisa hasil. Pada langkah ini, hasil keluaran OCR menggunakan Google Vision akan dibandingkan dengan karakter teks asli yang akan dibaca secara manual. Hasil perbandingan ini bisa disebut dengan tingkat kecocokan (*match rate*). Rumusan untuk menghitung tingkat kecocokan (*match rate*) adalah sebagai berikut:

$$r = \left(\frac{n}{m} \right) * 100\%$$

Deskripsi :

- r : tingkat kecocokan / *match rate*
n : jumlah karakter hasil OCR yang sesuai
m : jumlah karakter asli

Hasil perhitungan *match rate* berupa angka dalam persentase, dimana tingkat kecocokan atau keberhasilannya berbanding lurus dengan nilai *match rate*. Semakin tinggi nilai *match rate* maka semakin tinggi pula tingkat kecocokan atau keberhasilan metode OCR. Sebaliknya, nilai yang rendah menunjukkan metode OCR yang digunakan tidak optimal.

Penelitian ini menggunakan total 150 gambar plat nomor. Format gambar adalah .jpg dengan ukuran gambar bervariasi antara 1-2 MB. Hasil ekstraksi OCR akan dirangkum dalam sebuah tabel agar lebih mudah dianalisa dan dipahami. Tabel juga dilengkapi dengan nilai *match rate*.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini dibuat program dengan menggunakan bahasa pemrograman Python. Library yang digunakan adalah OpenCV dan Google Vision API. Sample citra yang digunakan berjumlah 150 citra plat nomor dengan variasi background hitam maupun putih. Ukuran citra juga bervariasi antara 1-2 MB.

3.1 Implementasi Preprocessing Citra : Edge Detection dan Contour

Dalam penelitian ini, proses *preprocessing* dan implementasi metode OCR dilakukan menggunakan program yang dibangun dengan bahasa pemrograman Python. Library yang digunakan adalah OpenCV. OpenCV sudah menyediakan method yang langsung bisa digunakan untuk melakukan proses Edge Detection dan Contour.

Hasil dari proses ini bisa dilihat pada gambar 2 hingga gambar 5 berikut ini.



Gambar 2. Gambar Plat Nomor Asli



Gambar 3. Gambar Plat Nomor Setelah Diterapkan Metode Canny

Ketika program dijalankan, program akan melakukan filtering untuk menemukan tepi dari gambar. Metode yang digunakan adalah metode Canny. Hasil dari filtering ini terlihat pada gambar 3 di atas.

Selanjutnya, citra dari hasil proses edge detection diolah kembali untuk menemukan contour pada setiap karakter pada citra plat nomor. Metode yang digunakan adalah metode contour dan sudah disediakan oleh library OpenCV. Hasil dari proses contour seperti pada gambar 4 dan 5 berikut ini. Gambar 4 menunjukkan penerapan metode contour untuk semua citra plat nomor. Sedangkan gambar 5 adalah bentuk zoom in dari gambar 4 agar garis tepi pada karakter terlihat lebih jelas.



Gambar 4. Gambar Plat Nomor Setelah Diterapkan Metode Contour



Gambar 5. Gambar Plat Nomor Setelah Diterapkan Metode Contour dan diperbesar

3.2 Ekstraksi Karakter dengan Google Vission

Setelah gambar selesai dilakukan proses *preprocessing* , maka citra plat nomor siap diekstraksi. Proses ekstraksi dilakukan dengan menggunakan library dari Google Vision API.

Untuk mengakses API OCR dari Google Vision, pengguna harus membuat akun di Google Cloud Platform (GCP), mengaktifkan Cloud Vision API, dan mengunduh Google Application Credentials dalam format JSON. Google Application Credentials ini digunakan untuk autentikasi akses API dan memastikan aplikasi dapat berinteraksi dengan layanan Google secara

aman. Setelah credentials diunduh, pengguna dapat mengatur variabel lingkungan pada sistem untuk menunjuk ke file JSON tersebut, atau memasukkannya langsung ke dalam kode program.

Selanjutnya akan dibuat program python untuk menggunakan melakukan OCR dengan library Google Vision ini. Contoh output dari program ini adalah sebagai berikut.

```
=== HASIL OCR ===  
  
B 4183 SEW  
HONDA  
04-29
```

Gambar 6. Hasil Ekstraksi dengan Google Vision Pada Plat Nomor B 4183 SEW

Hasil pada gambar 6 di atas merupakan hasil ekstraksi dari gambar plat nomor pada gambar 4 sebelumnya. Pada hasil ini terlihat bahwa ekstraksi OCR tidak hanya menghasilkan plat nomor saja, akan tetapi juga karakter lain seperti tulisan “HONDA” dan “04-29”. Padahal pada konteks ini, kedua tulisan tersebut tidak dibutuhkan.

Contoh lain yang lebih ekstrim adalah seperti ditunjukkan pada gambar 7.



Gambar 7. Gambar Plat Nomor B1201 AND

Ekstraksi citra pada gambar 7 menghasilkan luaran seperti pada gambar 8 berikut. Pada hasil ini terlihat bahwa banyak teks lain yang terbaca yang bukan teks plat nomor. Sedangkan hasil teks yang diinginkan hanya “B 1201 AND”. Sehingga hasil ini belum akurat dan perlu diproses lebih lanjut lagi.

```

=== HASIL OCR ===
FREEDOM
TK. TE
MITSUBIS
nsit Telaga
Pasar Minggu
Videotron erlimaan
Jl. Andalas
B 1201 AND
07.27
17 Sep 2023 08.54.27
323° NW
Hulawa
Kecamatan Telaga
Kabupaten Gorontalo
Gorontalo
Google
RIZKY GROSIR

```

Gambar 8. Hasil Ekstraksi dengan Google Vision Pada Plat Nomor B1201 AND

3.3 Penerapan Regex Pattern

Pada hasil sebelumnya, teks hasil OCR akan membaca keseluruhan teks pada citra. Padahal sebenarnya yang dibutuhkan hanya teks plat nomornya saja. Untuk mendapatkan teks plat nomor saja, maka bisa digunakan filter regex. Fungsi regex adalah fungsi yang digunakan untuk memfilter karakter teks sesuai dengan pattern/format tertentu. Fungsi ini bisa didapatkan dari library re. Pattern regex untuk membaca plat nomor, khususnya plat nomor kendaraan di Indonesia adalah sebagai berikut :

```
r'\b([A-Z]{1,2} ?\d{1,4} ?[A-Z]{1,3})\b'
```

Format ini akan mengambil karakter dengan pattern sebagai berikut :

- [A-Z]{1,2}**
Karakter adalah huruf string A sampai Z dengan jumlah karakter 1 atau 2
- ?**
Menandakan boleh spasi atau tidak (opsional)
- \d{1,4}**
Karakter angka antara 1-4 karakter.
- ?**
Menandakan boleh spasi atau tidak (opsional)
- [A-Z]{1,3}**
Karakter diakhiri dengan huruf antara A sampai Z dengan jumlah karakter 1 hingga 3.

Pada gambar 6 sebelumnya, hasil ekstraksinya adalah sebagai berikut :

B 4183 SEW
HONDA
04-29

Dengan filter ini, maka tulisan HONDA dan 04-29 akan diacuhkan, karena tidak memenuhi format yang ada. Sehingga hasil ekstraksi hanya akan menyisakan teks “**B 4183 SEW**” seperti yang ditunjukkan pada gambar 9 berikut.

```

=== NOMOR KENDARAAN TERDETEKSI ===
B 4183 SEW

```

Gambar 9. Hasil Filtering Regex Pattern Pada Plat Nomor B 4183 SEW

Sedangkan hasil ekstraksi untuk gambar 8, setelah difilter dengan regex akan menghasilkan keluaran seperti pada gambar 10 berikut.

=== NOMOR KENDARAAN TERDETEKSI ===
B 1201 AND

Gambar 10. Hasil Filtering Regex Pattern Pada Plat Nomor B 1201 AND

Selanjutnya, dikarenakan jumlah data sample mencapai 150 buah citra, maka citra sample dikumpulkan pada satu folder dan dilakukan proses perulangan untuk mengekstraksi karakter dari setiap gambar yang ada. Tabel 1 berikut adalah hasil ekstraksi dari sebagian data sample.

Data pada tabel 1 menunjukkan hasil dari ekstraksi plat nomor menggunakan Google Vision dengan beberapa sudut kamera. Gambar nomor 1-4 merupakan gambar standar yang sudah disesuaikan (di-crop) sehingga lebih mudah terbaca. Dari keempat gambar tersebut Google Vision mampu mengekstraksi dengan tepat semua plat nomor kendaraan.

Tabel 1 Hasil Ekstraksi Terhadap Data Sample

No	Gambar Plat Nomor	Hasil	No	Gambar Plat Nomor	Hasil
1		B 4183 SEW	5		B 1046 HUB
2		B 5758 FUH	6		D 5353 VIP
3		B 4266 FWZ	7		DD 8080 YU
4		B 2858 YH	8		B 1201 AND

Selanjutnya pada gambar 5-8 gambar plat nomor sengaja diambil sesuai dengan tangkapan kamera langsung di lapangan secara apa adanya, dimana gambar plat nomor belum disesuaikan. Pada gambar jenis inipun Google Vision mampu mengekstraksi katakter plat nomor dengan tepat. Bahkan pada gambar ke-8, posisi plat nomor sangat kecil dan dikaburkan dengan berbagai macam teks lain. Akan tetapi program tetap dapat mengidentifikasi plat nomor dengan tepat.

3.4 Menghitung Match Rate

Hasil akhir OCR akan dihitung *match rate*-nya menggunakan rumus yang sudah dipaparkan sebelumnya. Pada gambar 1 misalnya, bisa ditentukan :

Karakter Asli (dibaca manual) : B 4183 SEW

Karakter Hasil OCR : B 4183 SEW

Jumlah Karakter Asli (m) : 8

Jumlah Karakter OCR sesuai (n): 8

Maka, nilai *match rate* adalah :

$$r = \left(\frac{n}{m}\right) * 100\%$$

$$r = \left(\frac{8}{8}\right) * 100\%$$

$$r = 100\%$$

Rangkuman hasil ekstraksi teks beserta nilai *match rate*-nya bisa dilihat pada tabel 2.

3.5 Analisa Hasil

Analisa hasil ekstraksi didasarkan pada rangkuman hasil pengujian yang bisa dilihat pada tabel 2 berikut.

Tabel 2. Rangkuman Hasil Ekstraksi Terhadap Data Sample

No Gambar	Plat Nomor (Manual)	Plat Nomor (OCR)	Match rate (%)
Gambar 1	B 4183 SEW	B 4183 SEW	100
Gambar 2	B 5758 FUH	B 5758 FUH	100
Gambar 3	B 4266 FWZ	B 4266 FWZ	100
Gambar 4	B 2858 YH	B 2858 YH	100
Gambar 5	B 1046 HUB	B 1046 HUB	100
Gambar 6	D 5353 VIP	D 5353 VIP	100
Gambar 7	DD 8080 YU	DD 8080 YU	100
Gambar 8	B 1201 AND	B 1201 AND	100
Gambar 9	B 3732 KFU	B 3732 KFU	100
Gambar 10	B 6685 TOQ	B 6685 TOQ	100
Gambar 11	B 4399 KIX	B 4399 KIX	100
Gambar 12	B 5226 TNK	B 5226 TNK	100
Gambar 13	F 3363 MP	F 3363 MP	100
Gambar 14	B 5560 KJR	B 5560 KJR	100
Gambar 15	B 4512 KBZ	B 4512 KBZ	100
Gambar 16	B 4090 FJN	B 4090 FJN	100
Gambar 17	B 4271 KTA	B 4271 KTA	100
Gambar 18	B 4372 KID	B 4372 KID	100
Gambar 19	F 1658 IS	F 1658 IS	100
Gambar 20	F 5217 FIQ	F 5217 FIQ	100
Gambar 21	AA 4913 GL	AA 1229 TC AA4913 GL	50
dan seterusnya	dan seterusnya	dan seterusnya	dst
Gambar 149	B 1384 KZR	B 1384 KZR	100
Gambar 150	B 6408 KWV	B 6408 KWV	100
Rata-rata			99,50

Rangkuman hasil pengujian terhadap 150 data sample menunjukkan hasil OCR menggunakan Google Vision dan Regex Pattern sangat bagus dengan nilai rata-rata *Match rate* sebesar 99,50 %. Artinya hampir semua jenis sample dapat terbaca dengan tepat plat nomornya. Hasil ini memperkuat penelitian pada jurnal sebelumnya [13] yang hanya menghasilkan tingkat ketepatan sebesar 97,89 %. Penerapan Regex Pattern juga berperan sangat besar pada ketepatan ini karena berhasil memisahkan teks yang tidak dibutuhkan dan hanya mengambil teks plat nomor saja. Perpaduan kedua teknik ini relatif lebih baik dibandingkan menggunakan fungsi deteksi plat nomor saja, karena memberikan hasil yang tepat dengan waktu pemrosesan yang lebih cepat.

Sebagai catatan, sample data yang kurang sesuai juga tidak disebabkan karena kegagalan program, akan tetapi karena di dalam satu citra terdapat lebih dari satu plat nomor seperti yang ditunjukkan pada gambar 11 dan 12 berikut.



Gambar 11. Gambar Plat Nomor AA 4913 GL

```
=== NOMOR KENDARAAN TERDETEKSI ===  
AA 1229 TC  
AA 4913 GL
```

Gambar 12. Hasil Ekstaksi dengan Google Vision Pada Plat Nomor AA 4913 GL

Pada kasus ini program tetap mampu membaca kedua plat nomor dengan tepat, akan tetapi tidak bisa menentukan plat nomor mana yang dimaksud. Solusi yang bisa ditawarkan dari kasus seperti ini adalah dengan mengatur metode pengambilan gambar, dimana gambar yang dihasilkan kamera harus hanya memiliki satu plat nomor saja. Hal ini bisa dilakukan dengan beberapa cara. Salah satunya adalah memposisikan kamera tepat di depan plat nomor kendaraan, sehingga gambar plat nomor yang didapat benar-benar hanya kendaraan yang dimaksud. Atau bisa juga dengan cara membuat baris antrian yang lebih sempit, sehingga tidak memungkinkan plat nomor kendaraan di belakangnya dapat terlihat dalam kamera.

Penelitian ini juga menyoroti penggunaan proses *preprocessing* yaitu Edge Detection dan Contour. Hasil penelitian menunjukkan bahwa hasil ekstraksi teks pada OCR baik yang menggunakan citra hasil *preprocessing* maupun langsung pada citra asli tidak terdapat perbedaan terhadap hasil ekstraksi. Hal ini berarti proses *preprocessing* yang telah dilakukan, yaitu edge detection dan contour, tidak memberikan dampak terhadap hasil ekstraksi. Besar kemungkinan, Google Vision sudah memasukkan proses *preprocessing* di dalam rangkaian aktivitasnya. Atau bisa jadi Google Vision memiliki cara lain yang lebih bagus, yang tidak dipublikasikan. Sehingga, dalam penelitian ini, untuk efektifitas proses, langkah *preprocessing* citra tidak diperlukan lagi.

Melihat hasil yang sangat bagus dan mencapai 99,50 % penggunaan metode ini bisa direkomendasikan untuk diterapkan di industri, khususnya pada *Manless Parking System*. Penerapan metode ini pada *Manless Parking System* diharapkan dapat meningkatkan efisiensi dan keamanan sistem parkir.

4. KESIMPULAN

Penggunaan teknologi OCR menggunakan Google Vision dan Regex Pattern untuk ekstraksi karakter plat nomor sangat efektif. Hal ini dibuktikan dengan nilai rata-rata *match rate* yang mencapai 99,50 %. Nilai ini lebih tinggi dibandingkan penggunaan metode-metode lain pada penelitian-penelitian sebelumnya [11], [12], [13]. Penggunaan Regex Pattern juga berpengaruh signifikan pada hasil ekstraksi teks pada plat nomor.

Pada penelitian ini, proses *preprocessing* citra yang terdiri dari edge detection, dan contour, ternyata tidak berdampak pada hasil ekstraksi teks. Hal ini dibuktikan dengan hasil ekstraksi yang sama persis antara citra asli maupun citra setelah dilakukan *preprocessing*.

Sehingga dapat disimpulkan bahwa penerapan *preprocessing* pada teknologi OCR Google Vision tidak efektif digunakan. Selain itu tidak diperlukan pula fungsi deteksi plat nomor, dikarenakan metode ini dapat membaca plat nomor dengan tepat meskipun dengan sudut gambar yang luas dan obyek gambar yang beragam.

DAFTAR PUSTAKA

- [1] K. Ibnutama dan G. Suryanata, "Ekstraksi Karakter Citra Menggunakan Optical Character Recognition Untuk Pencetakan Nomor Kendaraan Pada Struk Parkir," *Jurnal Media Informatika Budidarma*, vol. 4, no. 4, 2020.
- [2] K. A. Hamad dan M. Kaya, "A Detailed Analysis of Optical Character Recognition Technology," *International Journal of Applied Mathematics, Electronics and Computers*, vol. 1, no. 4, 2016.
- [3] G. Kumar dan K. Bhatia, "A Detailed Review of Feature Extraction in Image Processing Systems," dalam *Fourth International Conference on Advanced Computing & Communication Technologies*, Rohtak, India, 2020.
- [4] M. D. N. Azis dan S. A. Syakri, "Rancang Bangun Aplikasi Perbaikan Citra Hasil Scan Dokumen Lama Dengan Metode Filtering," *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 2, no. 1, 2018.
- [5] K. Fatimah, Segmentasi Teks Naskah Kuno yang Lapuk Menggunakan Adaptive Local Threshholding, Surabaya, Indonesia: Departemen Teknik Komputer, Institut Teknologi Sepuluh November, 2018.
- [6] M. Brisinello, R. Grbić, M. Pul dan T. Anđelić, "Improving Optical Character Recognition Performance for Low Quality Images," dalam *International Symposium ELMAR*, Zadar, Croatia, 2017.
- [7] D. T. Anggraeni, "Perbaikan Citra Dokumen Hasil Pindai Menggunakan Metode Simple, Adaptive-Gaussian, dan Otsu Binarization Thresholding," *Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 2, no. 11, pp. 71-77, 2021.
- [8] C. Wibawa dan D. T. Anggraeni, "Comparison of Image Segmentation Method in Image Character Extraction *Preprocessing* Using Optical Character Recognition," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 4, pp. 583-589, 2023.
- [9] N. S. Salsabila, S. I. Anwar, I. A. Mahendar, A. N. Annaufal, M. A. Al Ataya, E. P. Giri dan G. P. Mindara, "Penerapan Teknologi OCR Plat Nomor Untuk Meningkatkan Efisiensi dan Keamanan Akses Kendaraan," *Journal of Sciencetech Research and Development*, vol. 2, no. 6, 2024.
- [10] Y. Galahartlambang, T. Khotiah, Z. Fanani dan A. A. Y. Solekhah, "Deteksi Plat Nomor Kendaraan Otomatis dengan Convolutional Neural Network dan OCR Pada Tempat Parkir ITB Ahmad Dahlan Lamongan," *Jurnal Manajemen Informatika & Sistem Informasi (MISI)*, vol. 2, no. 6, 2023.
- [11] A. Aprilino dan I. H. Al Amin, "Implementasi Algoritma YOLO dan Tesseract OCR Pada Sistem Deteksi Plat Nomor Otomatis," *Jurnal TEKNOINFO*, vol. 1, no. 16, pp. 54-59, 2022.
- [12] R. M. Ridwan, S. Sumaryo dan A. Zamhuri, "Implementasi Teknologi OCR Untuk Mendeteksi Nomor Plat Mobil," dalam *e-Proceeding of Engineering*, Bandung, Indonesia, 2024.
- [13] B. T. Kusmanto, N. P. T. Prakisya dan P. Hatta, "Comparative Analysis of Google Vision OCR with Tesseract on Newspaper Text Recognition Media of Computer science (MCs)," *Media of Computer Science*, vol. 1, no. 1, 2024.

- [14] S. dan S. , “Comparative Analysis of Sobel and Canny Method in Batik Kawung Image,” *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. X, no. 3, 2024.
- [15] A. Yudiyanto dan M. , “Implementasi Metode Canny Untuk Deteksi Tepi Mutu Daun Tembakau,” *Jurnal Sarjana Teknik Informatika*, vol. 3, no. 2, 2014.
- [16] M. Yokoyama dan T. Poggio, “A Contour-Based Moving Object Detection and Tracking. Massachusetts Institute of Technology,” dalam *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 2005.
- [17] Google Cloud, “OCR (Pengenalan Karakter Optik) dengan AI Google Cloud kelas dunia,” Google, 2024. [Online]. Available: <https://cloud.google.com/use-cases/ocr?hl=id#features>. [Diakses 10 04 2025].
- [18] Google Cloud, “Ekspresi Regular Python,” Google, 2025. [Online]. Available: <https://developers.google.com/edu/python/regular-expressions?hl=id>. [Diakses 10 04 2025].
- [19] M. Erwig dan R. Gopinath, “Explanations for Regular Expressions,” dalam *Proceedings of the 15th international conference on Fundamental Approaches to Software Engineering*, Oregon, USA, 2012.
- [20] W3Schools, “Python RegEx,” W3Schools, 2025. [Online]. Available: https://www.w3schools.com/python/python_regex.asp. [Diakses 14 04 2025].