

EVALUASI KINERJA TESSERACT-OCR DALAM PENGENALAN TEKS TULISAN TANGAN MENGGUNAKAN DATASET KUSTOM

Oktavia Rahmadani^{1*}, Rosa salsa saida², Chaerur Rozikin³

^{1,2,3}Universitas Singaperbangsa Karawang; Jl. HS. Ronggo Waluyo; (0267) 641177

Keywords:

Handwritten (OHTR);
Tesseract-OCR;
CER.

Correspondent Email:

2110631170152@student.unsika.ac.id



JITET is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Abstrak. Pengenalan teks tulisan tangan (OHTR) masih menjadi tren penelitian hingga saat ini karena masih banyaknya hal yang dapat dieksplorasi. Penelitian ini mengevaluasi kinerja *Tesseract-OCR* dalam mengenali teks tulisan tangan dengan mengumpulkan dan memproses 500 kata dalam bahasa Indonesia secara acak. Pengujian dilakukan dengan menghitung *Character Error Rate* (CER) antara teks asli dan teks hasil *Optical Character Recognition* (OCR) menggunakan *Tesseract*. Hasil menunjukkan bahwa rata-rata CER sebesar 35,08%, yang mengindikasikan tingkat akurasi pengenalan teks tulisan tangan sebesar 64,92%. Faktor-faktor seperti kualitas *input* data dan kompleksitas tulisan tangan mempengaruhi kinerja *Tesseract-OCR*. Penelitian ini memberikan kontribusi terhadap pemahaman kemampuan *Tesseract-OCR* dalam mengenali teks tulisan tangan serta menyoroti pentingnya pengembangan teknologi *Optical Character Recognition* (OCR) untuk aplikasi masa depan.

Abstract. Handwritten text recognition (OHTR) remains a research trend to date as there are still many aspects to explore. This research evaluates the performance of *Tesseract-OCR* in recognizing handwritten text by collecting and processing 500 randomly selected words in Indonesian language. Testing is conducted by calculating the *Character Error Rate* (CER) between the actual text and the obtained through *Optical Character Recognition* (OCR) using *Tesseract*. The results show an average CER of 35,08%, indicating a handwritten text recognition accuracy of 64,92%. Factors such as the quality of input data and complexity of handwritten text influence the performance of *Tesseract-OCR*. This research contributes to understanding the capabilities of *Tesseract-OCR* in recognizing handwritten text and highlights the importance of developing *Optical Character Recognition* (OCR) technology for future applications.

1. PENDAHULUAN

Teknologi saat ini dapat dikatakan sebagai bagian dari kehidupan sehari-hari. Teknologi terus dikembangkan seiring tingginya penggunaan operasi digital. Operasi digital dianggap memberikan kemudahan bagi penggunanya seperti pemesanan *online*, pertukaran informasi, pengenalan objek, dan sebagainya. Pengenalan objek yang dibahas

dalam penelitian ini adalah pengenalan karakter tulisan tangan.

Menurut Priya, dkk (2016), pengenalan karakter tulisan tangan dibedakan menjadi dua jenis, yaitu pengenalan karakter tulisan tangan *offline* dan *online*. Penelitian dalam bidang yang sama [10] juga membahas berbagai pendekatan dalam bidang pengenalan tulisan tangan, termasuk tulisan tangan *offline* dan *online*. Pengenalan karakter tulisan tangan

offline berupa pengenalan melalui gambar, sedangkan pengenalan karakter tulisan tangan *online* berupa pengenalan langsung menggunakan perangkat elektronik [1]. Penelitian lainnya dalam [12], menjelaskan bahwa dalam dekade terakhir telah banyak peneliti yang memperkenalkan berbagai metode untuk pengenalan teks tulisan tangan *offline* (OHTR) menggunakan teknik *deep learning* dan penelitian tersebut juga mengatakan bahwa OHTR ini masih menjadi tren penelitian di masa depan karena masih banyak yang dapat dieksplorasi dan ditingkatkan dalam bidang ini. Penelitian ini merupakan pengenalan karakter tulisan tangan *offline* yaitu menggunakan *dataset* kustom berupa gambar yang akan dikenali.

Pada penelitian-penelitian sebelumnya, teknologi pemindai (*scanner*) sering digunakan untuk menangkap gambar dari dokumen cetak. Namun, untuk mengonversi gambar tersebut menjadi teks digital, diperlukan perangkat lunak tambahan. *Optical Character Recognition* (OCR) adalah teknologi yang digunakan untuk mengubah gambar teks yang diketik, ditulis tangan, atau dicetak menjadi teks digital yang bisa diproses oleh komputer [2].

Tesseract-OCR adalah salah satu perangkat lunak OCR *open-source* dengan tingkat akurasi tinggi, sekitar 96,38% untuk dokumen cetak. Akurasi ini sangat dipengaruhi oleh kualitas citra dan ukuran *font*. Namun, akurasi dapat turun jika kualitas gambar kurang dari 300 dpi [2]. *Tesseract-OCR* ini telah banyak digunakan dalam berbagai aplikasi, baik untuk teks cetak maupun tulisan tangan.

Penelitian ini berfokus pada evaluasi kinerja *Tesseract-OCR* dalam mengenali teks tulisan tangan. mengingat tantangan yang lebih kompleks dalam pengenalan teks tulisan tangan dibandingkan dengan teks cetak, tujuan penelitian ini adalah mengevaluasi kemampuan *Tesseract-OCR* menggunakan *dataset* kustom yang diperoleh sendiri yang mencakup beberapa gaya tulisan tangan.

2. TINJAUAN PUSTAKA

2.1. *Tesseract-OCR*

Tesseract-OCR adalah sebuah *engine Optical Character Recognition* yang dikembangkan oleh Google. Dalam

pengembangan aplikasi, *Tesseract OCR* sering digunakan sebagai *library* tambahan dalam sebuah proyek, yang berfungsi untuk mendeteksi dan mengenali teks dari gambar. Contohnya, ketika mengembangkan aplikasi Android dengan menggunakan Android Studio, platform tersebut tidak memiliki fitur OCR bawaan. Oleh karena itu, pengembang yang ingin membuat aplikasi dengan OCR sebagai fitur utama harus menambahkan *library* eksternal, seperti *Tesseract-OCR*, ke dalam proyek mereka. *Tesseract-OCR* memungkinkan aplikasi tersebut untuk mengenali dan memproses teks yang ada pada gambar atau dokumen yang di-*scan*.

2.2. *Pengenalan Teks Tulisan Tangan*

Pengenalan tulisan tangan bertujuan untuk mengidentifikasi karakter seperti huruf dan angka. Berdasarkan metode *input*-nya, terdapat dua jenis pengenalan tulisan tangan: *offline* dan *online*. Pengenalan tulisan tangan *offline* mengidentifikasi teks dari citra teks tertulis, mengonversi teks dalam citra menjadi kode yang dapat digunakan oleh komputer dan aplikasi pengolahan teks. Data yang diperoleh dianggap sebagai representasi statis dari tulisan tangan. Tahapan dalam pengenalan tulisan tangan *offline* meliputi:

2.2.1 *Image Scanning*

Dokumen tertulis dipindai dengan *scanner*, menghasilkan citra dalam format seperti JPEG atau BMP.

2.2.2 *Image Processing*

Citra hasil *scanning* diolah melalui tahapan *pre-processing*, meliputi tahap *thresholding*, penghalusan, penyaringan, pengubahan skala, dan normalisasi, untuk mengekstraksi bagian teks yang relevan dan mempersiapkannya untuk segmentasi dan ekstraksi ciri.

2.2.3 *Segmentasi*

Citra diuraikan menjadi bagian-bagian kecil yang mewakili karakter individual.

2.2.4 *Ekstraksi Ciri*

Proses ini menemukan bentuk karakter pada citra, menyediakan informasi yang paling relevan dari data asli dan merepresentasikan dalam ruang dimensi lebih rendah.

2.2.5 *Neural Network*

Jaringan syaraf dilatih menggunakan citra awal untuk mengenali karakter. Algoritma *backpropagation* digunakan untuk melatih jaringan ini, sehingga jaringan dapat

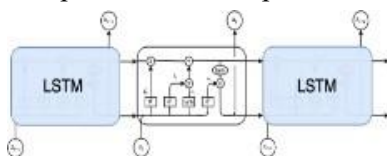
mengenali dan mengidentifikasi karakter pada citra baru.

2.3. LSTM (Long Short-Term Memory)

Long-Short Term Memory (LSTM) adalah salah satu jenis *Recurrent Neural Network* (RNN) yang sering digunakan untuk mengatasi masalah penumpukan gradien atau ketergantungan jangka panjang dalam pemrosesan dan prediksi data deret waktu. RNN cenderung mengalami penumpukan gradien, di mana nilai gradien bertabrakan, mengaburkan nilai kumulatifnya. LSTM dirancang untuk mengatasi masalah ini (Cholissodin dan Soebroto, 2019).

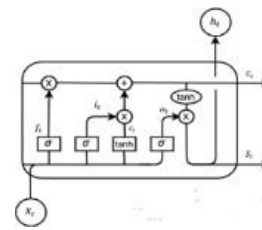
Menurut Sepp Hochreiter dan Jurgen Schmidhuber, model LSTM terdiri dari rangkaian sel memori yang menggantikan sel neuron pada lapisan tersembunyi RNN. LSTM menggunakan struktur gerbang untuk menyaring dan mempertahankan informasi yang relevan serta mengubah keadaan sel memori. Struktur gerbang ini meliputi *input gate*, *forget gate*, dan *output gate* (Dhea Larasati & Primandari, 2021). LSTM memiliki beberapa lapisan tersembunyi yang menyimpan informasi relevan dan membuang informasi yang tidak relevan di setiap sel (Hasiholan, Cholissodin, dan Yudistira, 2022).

Arsitektur LSTM sendiri terdiri dari tiga lapisan utama: lapisan input yang menerima data mentah, lapisan tersembunyi yang memproses dan menyimpan informasi temporal melalui unit-unit LSTM, serta lapisan output yang menghasilkan prediksi atau hasil akhir dari pemrosesan data pada Gambar 1.



Gambar 1. Arsitektur LSM

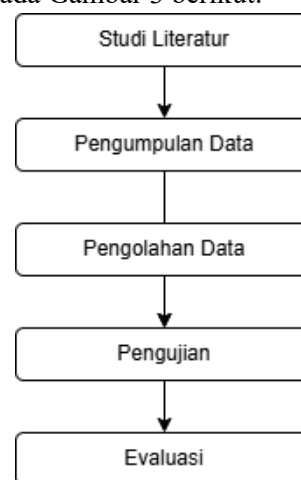
Pada lapisan tersembunyi terdapat *memory cell* yang terdiri dari tiga gerbang: *input gate*, *forget gate*, dan *output gate*. Penjelasan lebih rinci mengenai hal ini dapat dilihat pada Gambar 2.



Gambar 2. Memory Cell LSTM

3. METODE PENELITIAN

Metode penelitian dalam mengevaluasi kinerja *Tesseract-OCR* dilakukan berdasarkan tahapan pada Gambar 3 berikut.



Gambar 3. Kerangka Penelitian

3.1. Studi Literatur

Dalam penelitian ini, tahap awal yang dilakukan adalah studi literatur. Setiap peneliti melakukan studi literatur dengan tujuan menjadi dasar pemikiran untuk memperoleh dan membangun hipotesis penelitian. Tahap ini melibatkan identifikasi sumber-sumber pustaka yang relevan, seperti jurnal ilmiah terkini, buku digital, dan sumber internet lainnya.

3.2. Pengumpulan Data

Metode pengumpulan data dalam penelitian ini melibatkan proses menulis sebanyak 500 kata bahasa Indonesia secara acak pada kertas putih. Beberapa yang ditulis menggunakan huruf kecil semua dan beberapa diawali huruf besar. Kemudian, setiap tulisan akan difoto dengan kondisi pencahayaan yang cukup terang di bawah lampu, tanpa adanya bayangan yang mengganggu, dan dengan jarak sekitar 7-10 cm antara kamera dan kertas. Dengan demikian, data yang dimiliki sebanyak 500 data.

3.3. Pengolahan data

Berdasarkan data yang telah dikumpulkan, data pada tahap pengolahan data, hasil foto-foto tersebut akan diberi efek dokumen sehingga berwarna hitam putih. Proses ini bertujuan untuk meningkatkan kontras dan memudahkan pengenalan teks oleh *Tesseract-OCR*, yang selanjutnya akan diuji dan dievaluasi kinerjanya dalam mengenali teks tulisan tangan.

3.4. Pengujian

Pada tahap pengujian, data yang telah dikumpulkan diproses dengan dimasukkan ke dalam kode program untuk dibaca oleh *Tesseract-OCR*.

Tesseract-OCR mengadopsi metode dalam melakukan pengenalan kata, yaitu dengan menerapkan algoritma *Long Short-Term Memory* (LSTM) untuk mengenali urutan karakter dari teks pada gambar [2].

Pada tahap ini, terjadilah proses pengenalan gambar teks yang selanjutnya akan diubah menjadi teks yang dapat dibaca komputer. Dalam hal ini, terdapat beberapa tahapan di dalamnya yang bertujuan meningkatkan kualitas citra dan menghilangkan *noise*, yang dimulai dari tahap *input* citra, mengubah citra dari RGB ke *grayscale*, *threshold*, hingga konversi citra menjadi bentuk teks [12].

3.5. Evaluasi

Setelah tahap pengujian, dilakukan evaluasi terhadap kemampuan atau kinerja *Tesseract-OCR* dalam mengenali teks tulisan tangan berdasarkan hasil pengolahan data. Evaluasi ini mencakup analisis akurasi pengenalan teks serta membandingkan antara data asli dan hasil pengenalan teks dari *Tesseract-OCR*.

4. HASIL DAN PEMBAHASAN

Berikut adalah hasil dan pembahasan evaluasi kinerja *Tesseract-OCR* yang sudah dilakukan.

Sebelumnya telah disebutkan bahwa evaluasi pengenalan teks tulisan tangan menggunakan dataset kustom yang diperoleh sebanyak 500 data seperti Gambar 4.

DATASET



Gambar 4. Contoh Data yang Digunakan

Data yang dimasukkan diubah menjadi *image grayscale* yaitu gambar warna abu-abu hasil dari rata-rata *image* RGB [4]. Selanjutnya, dilakukan pengenalan teks tulisan tangan atau OCR dengan *Tesseract*.

```
# Baca file gambar
try:
    image = Image.open(image_path)
except Exception as e:
    raise Exception(f"Gagal membuka
file gambar: {e}")

# Konversi gambar menjadi grayscale
image = image.convert('L')

# Lakukan OCR dengan Tesseract
text =
pytesseract.image_to_string(image,
config='--psm 10')

# Kembalikan teks yang diekstrak
return text
```

Untuk melakukan evaluasi, diperlukan perhitungan *Character Error Rate* (CER) antara *string* asli dan *string* hasil OCR.

```
import pandas as pd
df_eval = pd.DataFrame({
    'Predicted Word Result' :
all_predicted_string,
    'Actual Word' : all_actual_string,
    'Character Error Rate (CER)' :
all_cer
})
df_eval
import numpy as np
np.mean(all_cer)
```

	Predicted Word Result	Actual Word	Character Error Rate(CER)
0	Baan	tujuan	1.000000
1	Yiwu	pintu	0.800000
2	Ungormas	informasi	0.333333
3	Clegan	elegan	0.166667
4	tayam	tajam	0.200000
...	...	...	...
495	dia	dia	0.000000
496	Dowrase	dewasa	0.714286
497	dekat	dekat	0.000000
498	Unog tuhan	ulang tahun	0.909091
499	ear	lebar	1.000000

500 rows x 3 columns

Gambar 5. Hasil perbandingan teks yang asli dengan teks yang dikenali dari hasil OCR.

Gambar 5 menunjukkan nilai *Character Error Rate* untuk masing-masing kata yang dihasilkan dari proses pengenalan teks menggunakan *library Tesseract* dengan label *predicted word*, yang kemudian dibandingkan dengan teks sebenarnya.

Tabel 1. Hasil akurasi pengenalan teks tulisan tangan

Nilai CER = 0	162
Nilai CER = 1	39
Nilai $0 < CER < 1$	299
Jumlah	500

Nilai CER berada di antara nilai 0 dan 1 sebagaimana disajikan pada Tabel 1. Berdasarkan 500 data yang dievaluasi, sebanyak 162 data memiliki nilai CER = 0, yang menunjukkan bahwa kata-kata tersebut dikenali tanpa kesalahan oleh *Tesseract* karena tidak terdapat perbedaan antara teks hasil (*predicted word*) dan teks asli (*actual word*). Namun, sebanyak 39 data memiliki nilai CER = 1, yang menunjukkan bahwa kata-kata tersebut secara keseluruhan berbeda antara *predicted word* dan *actual word*. Sementara itu, sebanyak 299 data memiliki nilai $0 < CER < 1$, menunjukkan adanya beberapa perbedaan antara *predicted word* dan *actual word*.

```
np.mean(all_cer)
```

0.35086237096237094

Gambar 6. Hasil Rata-Rata CER

Rata-rata CER yang diperoleh sebesar 0,3508 atau 35,08% yang dapat dilihat pada Gambar 6. Hasil tersebut menunjukkan bahwa kinerja pengenalan teks tulisan tangan oleh *Tesseract-OCR* mencapai 64,92%.

Untuk mengevaluasi kinerja, ini membutuhkan beberapa pertimbangan di antaranya:

- 1) Melihat kualitas *input* data;
- 2) Kompleksitas tulisan tangan;
- 3) Perbandingan dengan perangkat OCR lain; dan
- 4) Tujuan aplikasi yaitu apakah untuk keperluan sederhana atau tidak.

Pada penelitian terkait implementasi hirarki *dataset* dalam membangun model *language* Bali [9], mereka menjelaskan bahwa penggunaan kombinasi *dataset* yang berbeda dan hirarki *dataset* mempengaruhi peningkatan performa *coincidence*, dalam hal ini kebetulan pengenalan kata. Penelitian tersebut mengelompokkan *dataset* ke dalam beberapa kategori, yaitu kelompok *dataset* per karakter, per kata, per kalimat, dan per paragraf.

Dalam konteks kompleksitas, hasil evaluasi mungkin dianggap kurang memadai karena tulisan tangan yang diuji belum banyak variasi bentuk huruf sehingga dapat lebih mengetahui kemampuan dari *Tesseract-OCR*. Di sisi lain, dalam konteks seberapa besar kebutuhan akurasi, aplikasi yang hanya bertujuan untuk mengatasi atau memperbaiki kesalahan dapat dikatakan cukup memadai. Namun, untuk aplikasi yang memerlukan akurasi tinggi, seperti pengenalan formulir resmi atau dokumen penting, hasil ini mungkin kurang memadai.

5. KESIMPULAN

Berikut adalah kesimpulan dari penelitian yang telah dilakukan.

- a. Evaluasi kinerja *Tesseract-OCR* dalam pengenalan teks tulisan tangan menggunakan *dataset* kustom menunjukkan hasil yang bervariasi, dengan sebagian besar kata-kata dikenali dengan benar, tetapi juga terdapat sejumlah kata yang tidak dikenali dengan tepat.
- b. Nilai *Character Error Rate* (CER) yang diperoleh berada di antara 0 dan 1, dengan rata-rata CER sebesar 0,3508 atau 35,08%.

Hal ini menunjukkan bahwa kinerja pengenalan teks tulisan tangan oleh *Tesseract-OCR* mencapai sekitar 64,92%.

- c. Evaluasi kinerja ini mempertimbangkan kualitas *input* data, kompleksitas tulisan tangan, dan perbandingan dengan perangkat OCR lainnya. Namun, hasil evaluasi masih bisa dianggap kurang memadai dalam beberapa konteks, terutama untuk aplikasi yang memerlukan akurasi tinggi.
- d. Saran untuk penelitian selanjutnya adalah dapat melakukan evaluasi lebih lanjut dengan menggunakan variasi bentuk huruf yang lebih luas untuk menguji kemampuan *Tesseract-OCR* dalam mengenali teks tulisan tangan yang lebih kompleks. Selain itu, juga dapat mengeksplorasi penggunaan kombinasi *dataset* yang berbeda dan hirarki *dataset* untuk meningkatkan performa pengenalan kata, sebagaimana disarankan dalam penelitian terkait. Serta, dapat memperluas penelitian ke bidang aplikasi yang lebih luas, seperti implementasi pengenalan teks tulisan tangan dalam konteks bahasa atau budaya tertentu untuk mendapatkan pemahaman yang lebih mendalam tentang kinerja *Tesseract-OCR* dalam berbagai konteks penggunaan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada tim penelitian ini, dosen mata kuliah *Machine Learning*, dan semua pihak yang telah mendukung penelitian ini. Tanpa kerja sama dan dukungannya, penelitian ini tidak akan mencapai hasil yang memuaskan. Terima kasih atas waktu, dukungan, dan saran yang berharga.

DAFTAR PUSTAKA

- [1] M. A Liman, A. Josef, and G. P. Kusuma, "Handwritten Character Recognition Using Deep Learning Algorithm with Machine Learning Classifier", *Int. J. Inform. Visualization*, vol. 8, no. 1, pp. 150-157, 2024, doi: <https://dx.doi.org/10.62527/joiv.8.1.1707>.
- [2] M. Rizkiyanda, W. Priharti, and I. Hidayat, "Implementasi Pengenalan Tulisan Cetak Menggunakan *Tesseract* dan Komunikasi Jaringan Menggunakan *Python*", *eProceedings of Engineering*, vol. 11, no. 1, pp. 16-20, 2024.
- [3] A. F. I. Setyadi, and Y. A. Susetyo, "Implementasi Algoritma LSTM pada Aplikasi *Optical Character Recognition* Berbasis Website Menggunakan *Tesseract-OCR*", *J. Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 2, pp. 63-71, 2023, [Online]. Available: <https://repository.uksw.edu/handle/123456789/29937>
- [4] F. Abdiansyah, K. A. Santoso, and A. Kamsyakawuni, "Perbandingan *Image* RGB dan *Grayscale* pada Pengkodean *Image* dengan Algoritma *3D Playfair*", P. Seminar Nasional Matematika, vol. 4, pp. 524-533, 2021, [Online]. Available: <https://journal.unnes.ac.id/sju/prisma/article/view/45012>
- [5] M. I. N. Sidiq, "Aplikasi untuk Mengubah Gambar ke Teks dan Teks ke Suara dengan Metode *Optical Character Recognition* Berbasis Android", 2021, [Online]. Available: <http://elibrary.unikom.ac.id/id/eprint/5284>
- [6] F. P. Putri and A. Kusnadi, "Pengenalan Tulisan Tangan *Offline* dengan Algoritma *Generalized Hough Transform* dan *Backpropagation*", *J. Sistem Komputer*, vol. 10, no. 1, pp. 5-12, 2018, doi: <https://doi.org/10.31937/sk.v10i1.890>.
- [7] T. B. Sianturi, I. Cholissodin, and N. Yudistira., "Penerapan Algoritma *Long Short-Term Memory* (LSTM) Berbasis Multi-Fungsi Aktivasi Terbobot dalam Prediksi Harga *Ethereum*", *J. Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 3, pp. 1101-1107, 2023, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12384>
- [8] M. F. Rizkilloh and S. Widiyanesti, "Prediksi Harga *Cryptocurrency* Menggunakan Algoritma Long Short Term Memory (LSTM)", *J. Rekayasa Sistem dan Teknologi Informasi*, vol. 6, no. 1, pp. 25-31, 2022, doi: <https://doi.org/10.29207/resti.v6i1.3630>.
- [9] A. Asroni, G. Indrawan, and L. J. E. Dewi, "Implementasi Hirarki *Dataset* dalam Membangun Model *Language* Bali Menggunakan *Framework Tesseract OCR*", *J. Resistor*, vol. 6, no.1, 2023, doi: <https://doi.org/10.31598>.
- [10] P. C. Vashist, A. Pandey, and A. Tripathi, "A Comparative Study of Handwriting Recognition Techniques", *Int. Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 2020, doi: <https://doi.org/10.1109/ICCAKM46823.2020.9051464>.
- [11] Y. Wang, W. Xiao, and S. Li, "Offline Handwritten Text Recognition Using Deep Learning A Review", *J. of Physics: Conference Series*, 2021, doi: 10.1088/1742-6596/1848/1/012015.

- [12]K. Kusnantoro, T. Rohana, and D. S. Kusumaningrum, "Implementasi Metod sad Tesseract OCR (Optical Character Recognition) untuk Deteksi Plat Nomor Kendaraan pada Sistem Parkir", Scientific Student J. for Information, Tech. and Science, vol. 3, no. 1, pp. 59-67, 2022.
- [13]J. A. Patel, "Handwritten and Printed Text Recognition Using Tesseract-OCR", Int. J. of Creative Research Thoughts, vol. 9, no. 9, pp. 69-77, 2021, [Online]. Available: <http://www.ijert.org/papers/IJCRT2109326.pdf>
- [14]M. R. Toha and A. Triayudi, "Penerapan Membaca Tulisan di dalam Gambar Menggunakan Metode OCR Berbasis Website pada e-KTP", J. Sains dan Teknologi, vol. 11, no. 1, pp. 175-183, 2022, doi: <https://dx.doi.org/10.23887/jst-undiksha.v11i1>.
- [15]T. I. Cahyani, dkk., "Perbandingan Akurasi Pengenalan Karakter Plat Nomor Menggunakan Tesseract dan Data Latih Emnist", J. I. Technology and Computer Science, vol. 5, no. 2, 2022.

