

# PERBANDINGAN AKURASI TESSERACT DAN EASYOCR SEBELUM DAN SESUDAH PRAPEMROSESAN PADA CITRA NOTA

**Khinanti Angelita Puteri<sup>1</sup>, Faliana Alifia<sup>2</sup>, Puti Aisyah Lailatulrahmi<sup>3</sup>, Gema Parasti Mindara<sup>4</sup>, Endang Purnama Giri<sup>5</sup>**

<sup>1,2,3</sup>Teknologi Rekayasa Perangkat Lunak, IPB University; Jl. Kumbang No.14; (0251) 8329101

<sup>4</sup>Teknologi Rekayasa Komputer, IPB University; Jl. Kumbang No.14; (0251) 8329101

<sup>5</sup>Ilmu Komputer, IPB University; Wing 20, Jl. Meranti Kampus IPB Level 5; (0251) 8625584

## Keywords:

Character Error Rate;  
EasyOCR;  
Nota Transaksi;  
Optical Character  
Recognition;  
Prapemrosesan Citra

## Correspondent Email:

gemaparasti@apps.ipb.ac.id

**Abstrak.** Pengenalan teks pada citra nota menggunakan *Optical Character Recognition* (OCR) masih relevan diteliti karena tingginya variasi kualitas citra. Penelitian ini mengevaluasi kinerja Tesseract dan EasyOCR dalam mengenali teks pada citra nota dengan beberapa metode prapemrosesan. Dataset berasal dari Kaggle dengan 50 sampel citra yang dipilih menggunakan *stratified sampling*. Pengujian dilakukan dengan menghitung *Character Error Rate* (CER) antara hasil OCR dan *ground truth*. Hasil menunjukkan nilai CER berada pada kisaran 18%–25%, dengan performa terbaik Tesseract pada mode *denoise* dan EasyOCR pada mode *grayscale*. Metode *threshold* memberikan penurunan akurasi paling signifikan. Kualitas citra dan jenis prapemrosesan terbukti memengaruhi kinerja OCR, sehingga pemilihan prapemrosesan yang tepat sangat penting dalam meningkatkan akurasi pengenalan teks pada citra nota.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

**Abstract.** *Text recognition on receipt images using Optical Character Recognition (OCR) remains a relevant research topic due to the high variation in image quality. This study evaluates the performance of Tesseract and EasyOCR in recognizing text from receipt images using several preprocessing methods. The dataset was obtained from Kaggle, consisting of 50 samples selected through stratified sampling. The evaluation was conducted by calculating the Character Error Rate (CER) between the OCR results and the ground truth. The results show that CER values range from 18% to 25%, with the best performance achieved by Tesseract in the denoise mode and by EasyOCR in the grayscale mode. The threshold method causes the most significant decrease in accuracy. Image quality and preprocessing methods are proven to influence OCR performance, indicating that proper preprocessing selection is essential for improving text recognition accuracy on receipt images.*

## 1. PENDAHULUAN

Dokumen berbasis kertas seperti nota dan faktur masih sangat umum digunakan dalam aktivitas bisnis maupun personal sebagai media pencatatan transaksi keuangan. Pengolahan manual terhadap dokumen tersebut membutuhkan waktu dan rentan terhadap kesalahan manusia, sehingga mendorong kebutuhan akan otomasi digital melalui teknologi pengenalan karakter optis atau *Optical Character Recognition* (OCR).

Teknologi OCR memungkinkan ekstraksi teks dari citra dokumen secara otomatis dan cepat, serta telah menjadi bagian integral dalam proses digitalisasi dokumen pada berbagai aplikasi bisnis modern [1]. Seiring kemajuan pembelajaran mesin dan komputasi awan, performa OCR terus mengalami peningkatan dari sisi akurasi maupun cakupan bahasa yang dapat dikenali.

Tesseract dan EasyOCR merupakan dua solusi OCR *open-source* yang paling sering

digunakan. Tesseract mengadopsi arsitektur Long Short-Term Memory (LSTM) yang unggul pada teks cetak standar [2], sedangkan EasyOCR memanfaatkan kombinasi CRAFT (*Character Region Awareness For Text*) dan CRNN (*Convolutional Recurrent Neural Network*) dengan 7 convolutional layers dan 2 LSTM layers yang lebih adaptif terhadap variasi font dan kondisi citra menantang [3], [4].

Prapemrosesan citra merupakan komponen penting dalam pipeline OCR, terutama pada dokumen nota yang memiliki pencahayaan tidak merata, noise, atau distorsi. Akoushideh et al. (2025) menunjukkan bahwa prapemrosesan dapat meningkatkan akurasi hingga 98,6% dengan error rate turun dari 13,90% menjadi 1,40% [3]. Maliński dan Okarma (2023) mengidentifikasi berbagai metode efektif seperti adaptive thresholding dan noise reduction pada kondisi pencahayaan tidak terkontrol [5].

Penelitian terkait OCR pada nota dan faktur menunjukkan variasi hasil antar-engine. Ha dan Horák (2022) mencapai akurasi 90% untuk faktur berbahasa Inggris dengan sistem OCRMiner [6]. Nayeem dan Mondal (2022) menemukan EasyOCR mencapai akurasi 70% pada gambar statis, lebih tinggi dari Tesseract (68%) [7], dengan EasyOCR lebih baik pada handwritten text dan noisy images, sementara Tesseract lebih cepat dalam processing. Rexhepi et al. (2025) mengonfirmasi Tesseract sebagai OCR paling populer, sementara solusi deep learning seperti EasyOCR semakin diminati [1].

Secara umum, performa OCR dipengaruhi oleh engine, kualitas citra, dan pipeline prapemrosesan. Namun, masih sedikit penelitian yang secara khusus membandingkan Tesseract dan EasyOCR pada citra nota dengan mengevaluasi pengaruh prapemrosesan secara terukur terhadap akurasi keduanya. Oleh karena itu, penelitian ini bertujuan menganalisis perbandingan akurasi Tesseract dan EasyOCR sebelum dan sesudah prapemrosesan, mengevaluasi pengaruh teknik prapemrosesan terhadap performa kedua engine, serta memberikan rekomendasi pipeline OCR yang optimal untuk ekstraksi teks nota.

## 2. TINJAUAN PUSTAKA

### 2.1. *Optical Character Recognition*

*Optical Character Recognition* (OCR) merupakan teknik yang berfungsi mengubah teks pada citra baik berupa dokumen hasil pemindaian, foto, maupun gambar lainnya menjadi karakter digital yang dapat diproses oleh komputer. Teknologi ini memungkinkan sistem mengenali dan mengekstraksi tulisan dari gambar sehingga informasi tersebut dapat diolah sebagai data teks. Dalam konteks penelitian ini, OCR diaplikasikan untuk membaca dan mengekstraksi informasi teks dari foto nota [8], [9].

### 2.2. *Prapemrosesan*

Prapemrosesan merupakan tahap awal dalam pengolahan citra yang bertujuan meningkatkan kualitas gambar agar lebih mudah diinterpretasikan pada proses selanjutnya. Pada tahap ini biasanya diterapkan berbagai teknik seperti binarisasi, deteksi dan reduksi noise, serta penyesuaian visual lainnya untuk menghasilkan citra yang lebih bersih dan terstruktur [10], [11].

### 2.3. *Tesseract*

Tesseract OCR merupakan mesin pengenalan karakter yang dikembangkan oleh Google dan banyak digunakan sebagai pustaka pendukung dalam berbagai aplikasi untuk mendeteksi serta mengenali teks pada citra [12]. Mesin OCR ini dibangun menggunakan bahasa C++, sehingga dapat berjalan pada berbagai sistem operasi seperti Windows, Linux, dan MacOS. Selain itu, tersedia pula beragam library antarmuka untuk banyak bahasa pemrograman, yang memungkinkan Tesseract digunakan secara luas pada berbagai platform, termasuk perangkat mobile seperti Android [11].

### 2.4. *EasyOCR*

EasyOCR merupakan perangkat OCR berbasis kecerdasan buatan yang bersifat open-source dan dirancang untuk mengekstraksi teks dari gambar maupun dokumen digital menjadi teks yang dapat dibaca dan diolah oleh komputer. Model ini memanfaatkan arsitektur CRNN (*Convolutional Recurrent Neural Network*) untuk melakukan proses identifikasi karakter, khususnya huruf latin, pada citra.

Selain itu, EasyOCR mendukung pengenalan teks dalam berbagai bahasa sehingga dapat diterapkan secara luas dalam kebutuhan ekstraksi informasi dari gambar [13], [14].

### 3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan rancangan studi komparatif untuk membandingkan kinerja dua *engine Optical Character Recognition* (OCR), yaitu Tesseract dan EasyOCR, pada citra nota dengan berbagai kondisi prapemrosesan (Al-Shboul et al., 2025). Seluruh proses pengujian dirancang dalam bentuk *pipeline* pemrosesan terstruktur yang mencakup pemilihan *dataset*, penerapan teknik prapemrosesan, eksekusi OCR, evaluasi akurasi menggunakan *Character Error Rate* (CER) [15], serta pengujian signifikansi secara statistik menggunakan uji *Wilcoxon Signed-Rank* [14]. Perancangan metode ini bertujuan untuk menghasilkan pengukuran yang objektif, terkontrol, dan dapat direplikasi secara ilmiah.

#### 3.1. Rancangan Penelitian

Penelitian ini menerapkan pendekatan eksperimental dengan desain *comparative study*, yaitu membandingkan performa dua sistem OCR pada kondisi perlakuan yang sama. Objek yang dibandingkan adalah dua *engine* OCR yang memiliki pendekatan berbeda, yakni Tesseract yang berbasis segmentasi dan LSTM, serta EasyOCR yang berbasis *deep learning*.

Tujuan utama dari rancangan penelitian ini adalah untuk mengukur dan membandingkan tingkat akurasi pengenalan karakter Tesseract dan EasyOCR pada empat kondisi prapemrosesan, yaitu *original*, *grayscale*, *threshold*, dan *denoise*. Dengan pendekatan ini, pengaruh masing-masing teknik prapemrosesan terhadap kedua *engine* OCR dapat dianalisis secara kuantitatif [16].

#### 3.2. Sumber Data dan Teknik Pengumpulan

*Dataset* yang digunakan dalam penelitian ini berasal dari platform Kaggle dengan nama Find It Again Dataset. *Dataset* ini terdiri dari 218 pasang data, di mana setiap pasangan berisi satu citra nota hasil pemindaian dan satu berkas *ground truth* yang merepresentasikan isi teks asli dari nota tersebut.

*Dataset* ini dipilih karena merepresentasikan kondisi dunia nyata yang menantang bagi sistem OCR, seperti variasi tingkat pencahayaan, keberadaan *noise* hasil pemindaian, ketidakteraturan tata letak, perbedaan ketebalan tinta, serta degradasi fisik kertas. Karakteristik tersebut menjadikan *dataset* ini relevan untuk menguji ketahanan (*robustness*) sistem OCR pada skenario praktis.

Teknik pengambilan sampel dilakukan menggunakan *stratified sampling* berdasarkan tingkat kecerahan citra. Stratifikasi ini bertujuan untuk memastikan bahwa sampel uji mencakup variasi kondisi terang, sedang, dan gelap secara proporsional. Dari total 218 pasangan data, dipilih 50 sampel citra sebagai data uji utama. Pendekatan ini digunakan untuk menjaga keseimbangan karakteristik visual data sekaligus memastikan keterwakilan kondisi ekstrem pada nota transaksi.

#### 3.3. Teknik dan Prosedur Prapemrosesan Citra

Empat metode prapemrosesan diterapkan dalam penelitian ini untuk mengkaji pengaruh perubahan kualitas visual terhadap kinerja OCR, yaitu *original*, *grayscale*, *threshold*, dan *denoise*. Setiap metode memiliki tujuan, mekanisme penerapan, serta dasar pemilihan yang berbeda [16].

##### 3.3.1. Original

Mode *original* mempertahankan citra tanpa perubahan apapun. Citra langsung diproses oleh engine OCR dalam kondisi visual apa adanya. Mode ini digunakan sebagai baseline untuk membandingkan dampak teknik prapemrosesan lainnya terhadap kinerja pengenalan karakter.

##### 3.3.2. Grayscale

Konversi *grayscale* bertujuan untuk mereduksi kompleksitas data dari tiga kanal warna (RGB) menjadi satu kanal intensitas. Proses ini dilakukan menggunakan transformasi linear berdasarkan kontribusi masing-masing kanal warna terhadap intensitas cahaya. Metode ini dipilih karena dapat menyederhanakan

informasi visual tanpa menghilangkan struktur dasar karakter.

### 3.3.3. *Threshold (Otsu)*

Metode *thresholding* digunakan untuk melakukan binarisasi citra dengan memisahkan objek teks dan latar belakang berdasarkan nilai ambang. Ambang ditentukan secara otomatis menggunakan metode Otsu yang memaksimalkan separasi antara dua kelas histogram intensitas. Teknik ini dipilih karena umum digunakan pada sistem OCR untuk meningkatkan kontras teks terhadap latar.

### 3.3.4. *Denoise (Gaussian Blur)*

Denoising dilakukan menggunakan Gaussian Blur dengan kernel berukuran 5x5 untuk mengurangi *noise* frekuensi tinggi yang sering muncul akibat proses pemindaian. Teknik ini bertujuan untuk menghaluskan latar belakang tanpa menghilangkan struktur utama karakter [17]. Gaussian Blur dipilih karena memiliki efisiensi komputasi yang tinggi dan efektif dalam mengurangi *noise* acak.

## 3.4. *Engine OCR yang Digunakan*

Penelitian ini menggunakan dua *engine* OCR dengan karakteristik pendekatan yang berbeda untuk memastikan bahwa proses perbandingan akurasi dilakukan secara objektif dan representatif terhadap dua paradigma utama dalam teknologi OCR, yaitu pendekatan berbasis model konvensional dan pendekatan berbasis pembelajaran mendalam [18]. Penggunaan dua *engine* yang berbeda dimaksudkan untuk mengevaluasi sejauh mana metode prapemrosesan citra memengaruhi performa masing-masing pendekatan dalam konteks pengenalan teks pada citra nota.

### 3.4.1. *Tesseract OCR*

Tesseract digunakan sebagai representasi *engine* OCR konvensional yang telah lama digunakan sebagai *baseline* dalam berbagai penelitian ekstraksi teks. Dalam penelitian ini, Tesseract dimanfaatkan untuk menguji bagaimana *engine* berbasis LSTM merespons variasi kualitas citra nota hasil penerapan metode prapemrosesan yang

berbeda. Tesseract dijalankan dengan konfigurasi `--oem 3` untuk mengaktifkan mode LSTM sepenuhnya dan `--psm 6` untuk mengasumsikan bahwa setiap citra memuat satu blok teks utama. Konfigurasi ini dipilih agar proses ekstraksi teks lebih terfokus pada isi utama nota dan meminimalkan kesalahan segmentasi akibat asumsi tata letak yang tidak sesuai.

Penggunaan Tesseract dalam penelitian ini ditujukan untuk mengukur tingkat sensitivitas *engine* OCR konvensional terhadap perubahan kualitas citra akibat prapemrosesan [19], khususnya pada kondisi citra dengan *noise*, kontras rendah, dan variasi pencahayaan. Hasil dari Tesseract digunakan sebagai *baseline* pembandingan untuk menilai efektivitas pendekatan OCR berbasis *deep learning*.

### 3.4.2. *EasyOCR*

EasyOCR digunakan sebagai representasi *engine* OCR modern berbasis pembelajaran mendalam untuk menguji performa pengenalan teks pada citra nota dengan karakteristik kompleks. Dalam penelitian ini, EasyOCR dimanfaatkan untuk mengevaluasi bagaimana model berbasis jaringan saraf merespons variasi prapemrosesan citra yang sama dengan yang diberikan kepada Tesseract. Pendekatan deteksi wilayah teks berbasis CRAFT dan pengenalan karakter menggunakan CRNN memungkinkan EasyOCR untuk memproses teks dengan variasi orientasi, ketebalan huruf, dan tata letak yang tidak seragam [18].

Pemilihan EasyOCR dalam penelitian ini bertujuan untuk melihat apakah pendekatan berbasis *deep learning* memberikan tingkat ketahanan (*robustness*) yang lebih baik terhadap degradasi citra dibandingkan *engine* konvensional. Selain itu, EasyOCR digunakan untuk mengidentifikasi sejauh mana metode prapemrosesan tertentu mampu meningkatkan atau justru menurunkan akurasi pengenalan pada model modern berbasis data. Evaluasi kinerja kedua *engine* ini akan diukur menggunakan metrik *Character Error Rate* (CER) [20].

### 3.5. Evaluasi Akurasi Menggunakan Character Error Rate (CER)

Evaluasi kinerja OCR dilakukan menggunakan metrik *Character Error Rate* (CER), yang mengukur tingkat kesalahan pengenalan karakter berdasarkan jarak edit Levenshtein antara teks hasil OCR dan *ground truth* [15]. CER dirumuskan sebagai berikut:

$$CER = \frac{\text{Levenshtein Distance}}{\text{Jumlah Karakter Ground Truth}} \quad (1)$$

**Gambar 1.** Rumus CER

CER dipilih karena lebih sensitif dalam mendeteksi kesalahan kecil pada tingkat karakter dibandingkan metrik berbasis kata seperti *Word Error Rate* (WER). Nilai CER yang semakin kecil menunjukkan tingkat akurasi pengenalan karakter yang semakin baik.

### 3.6. Prosedur Pengujian dan Analisis Statistik

Setiap citra sampel diproses menggunakan empat kondisi prapemrosesan. Seluruh hasil citra kemudian dieksekusi menggunakan Tesseract dan EasyOCR secara terpisah. Hasil teks OCR dibandingkan dengan *ground truth* untuk memperoleh nilai CER pada setiap kombinasi citra, metode prapemrosesan, dan engine OCR.

Untuk memperoleh estimasi statistik yang stabil, dilakukan bootstrapping sebanyak 2000 iterasi untuk menghasilkan nilai rata-rata CER, standar deviasi, serta interval kepercayaan 95%. Selanjutnya, uji *Wilcoxon Signed-Rank Test* digunakan untuk menentukan signifikansi perbedaan performa antara Tesseract dan EasyOCR pada setiap metode prapemrosesan. Uji ini dipilih karena data CER bersifat berpasangan dan tidak diasumsikan berdistribusi normal.

### 3.7. Implementasi Sistem OCR

Implementasi sistem dilakukan secara terkomputerisasi menggunakan bahasa pemrograman Python. Lingkungan pengembangan menggunakan sistem operasi Windows dengan dukungan pustaka pemrosesan citra, OCR, dan analisis statistik. Pustaka utama yang digunakan meliputi OpenCV untuk prapemrosesan, pytesseract sebagai antarmuka Tesseract OCR, EasyOCR untuk pengenalan berbasis *deep learning*, NumPy untuk manipulasi numerik, pandas untuk pengelolaan data eksperimen, serta SciPy untuk pengujian statistik.

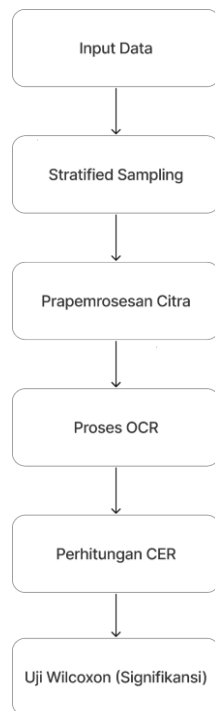
*Dataset* dikelola dalam struktur direktori terpisah antara citra dan *ground truth* teks. Setiap citra dipetakan secara otomatis dengan berkas referensi yang sesuai. Proses iterasi dilakukan secara *batch* untuk memastikan konsistensi pengujian pada seluruh sampel.

*Pipeline* pemrosesan pada penelitian ini mengikuti tahapan yang ditunjukkan pada Gambar 2, dimulai dari pemuatan citra beserta berkas *ground truth*, kemudian dilanjutkan dengan proses *stratified sampling* untuk memperoleh 50 sampel uji yang mewakili variasi tingkat kecerahan. Setiap citra sampel diproses menggunakan empat mode prapemrosesan, yaitu *original*, *grayscale*, *threshold*, dan *denoise*. Konsep penerapan prapemrosesan ini mengacu pada temuan sebelumnya bahwa kualitas citra memiliki pengaruh langsung terhadap akurasi OCR dan bahwa teknik prapemrosesan mampu meningkatkan performa pengenalan karakter [16].

Citra hasil prapemrosesan selanjutnya dieksekusi menggunakan kedua engine OCR, yaitu Tesseract dan EasyOCR. Teks hasil ekstraksi kemudian dinormalisasi sebelum dihitung nilai CER. Seluruh nilai CER disimpan dalam struktur data yang terorganisir untuk keperluan perhitungan rata-rata, standar deviasi, dan interval kepercayaan melalui proses bootstrapping.

Tahap akhir meliputi pengujian signifikansi menggunakan uji *Wilcoxon Signed-Rank*, yang digunakan untuk menilai perbedaan performa Tesseract dan EasyOCR pada setiap mode prapemrosesan. Seluruh implementasi dijalankan secara terotomasi menggunakan

Python agar eksperimen dapat direplikasi dengan konfigurasi yang sama.



**Gambar 2.** Kerangka Penelitian

## 4. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil pengujian dan pembahasan terhadap sistem *Optical Character Recognition* (OCR) yang dikembangkan. Pengujian difokuskan pada analisis kinerja dua engine OCR, yaitu Tesseract dan EasyOCR, dengan menerapkan empat metode praproses citra, yakni *original*, *grayscale*, *threshold*, dan *denoise*. Evaluasi dilakukan menggunakan metrik *Character Error Rate* (CER) dan uji statistik *Wilcoxon Signed-Rank Test* untuk mengetahui signifikansi perbedaan performa. Selain itu, disajikan pula analisis visual untuk mengkaji pengaruh masing-masing metode praproses terhadap kualitas citra nota.

### 4.1. Rancangan dan Pengaturan Eksperimen

Rancangan eksperimen disusun untuk memastikan bahwa proses evaluasi kinerja OCR dilakukan secara sistematis, terukur, dan selaras dengan tahapan metodologis yang telah dijelaskan pada Bab 3. Seluruh komponen pengujian dirancang untuk menghasilkan penilaian yang objektif terhadap pengaruh variasi prapemrosesan serta perbedaan

karakteristik antara dua engine OCR yang diuji. Ruang lingkup pengaturan eksperimen mencakup deskripsi dataset yang digunakan, strategi pemilihan sampel uji, konfigurasi prapemrosesan, mekanisme eksekusi OCR, serta prosedur evaluasi kuantitatif yang menjadi landasan analisis pada bagian-bagian berikutnya.

#### 4.1.1. Dataset dan Pengaturan Pengujian

Penelitian ini menggunakan dataset citra nota transaksi beserta teks *ground truth* yang diperoleh dari repositori publik Kaggle dengan judul *Find It Again Dataset* [21]. Dataset tersebut terdiri atas 218 pasangan data, di mana setiap pasangan mencakup satu citra nota dan satu berkas teks yang merepresentasikan isi sebenarnya dari nota. Dataset ini digunakan sebagai acuan utama dalam mengevaluasi kinerja sistem *Optical Character Recognition* (OCR).

Untuk menjamin pemerataan karakteristik data uji, pemilihan sampel dilakukan menggunakan metode *stratified sampling* berdasarkan tingkat kecerahan citra. Pendekatan ini bertujuan untuk menghindari bias pengujian yang disebabkan oleh dominasi kondisi pencahayaan tertentu. Dari keseluruhan dataset, dipilih sebanyak 50 sampel data yang mewakili setiap strata secara proporsional.

Setiap citra sampel selanjutnya diproses menggunakan empat variasi metode praproses, yaitu *original* (tanpa praproses), *grayscale*, *threshold*, dan *denoise*. Metode *grayscale* digunakan untuk mengonversi citra menjadi skala abu-abu, metode *threshold* digunakan untuk proses binarisasi citra, sedangkan metode *denoise* digunakan untuk mengurangi derau (*noise*) pada citra. Keempat variasi praproses tersebut diterapkan untuk mengevaluasi pengaruh karakteristik citra terhadap kinerja sistem OCR.

Proses pengenalan karakter dilakukan menggunakan dua engine OCR, yaitu Tesseract dan EasyOCR. Tesseract merupakan engine OCR berbasis metode konvensional, sedangkan EasyOCR merupakan engine OCR berbasis *deep learning*. Penggunaan dua engine dengan pendekatan berbeda ini bertujuan untuk membandingkan performa pengenalan karakter pada setiap variasi metode praproses citra.

Evaluasi kinerja sistem dilakukan menggunakan metrik *Character Error Rate* (CER). CER digunakan untuk mengukur tingkat kesalahan pengenalan karakter dengan membandingkan jumlah operasi penyuntingan (*edit distance*) yang diperlukan untuk mengubah hasil OCR menjadi teks *ground truth* terhadap jumlah karakter pada *ground truth*. Nilai CER yang semakin kecil menunjukkan tingkat akurasi pengenalan karakter yang semakin baik.

Selain itu, untuk mengetahui apakah terdapat perbedaan performa yang signifikan antara Tesseract dan EasyOCR pada setiap metode praproses, dilakukan uji statistik *Wilcoxon Signed-Rank Test*. Uji ini digunakan karena data hasil pengujian bersifat berpasangan dan tidak diasumsikan berdistribusi normal. Hasil *Wilcoxon Test* digunakan sebagai dasar dalam penarikan kesimpulan mengenai signifikansi perbedaan performa kedua engine OCR.

#### 4.2. Hasil Pengujian Akurasi OCR Berdasarkan Prapemrosesan

Evaluasi performa OCR dilakukan pada dua engine, yaitu Tesseract dan EasyOCR, menggunakan empat jenis prapemrosesan citra *original*, *grayscale*, *threshold*, dan *denoise*. Analisis ini bertujuan untuk melihat bagaimana setiap metode prapemrosesan memengaruhi tingkat kesalahan pengenalan karakter yang diukur menggunakan *Character Error Rate* (CER). Pengujian dilakukan terhadap 50 sampel citra nota hasil *stratified sampling* dari total 218 dataset, sehingga setiap kategori kualitas nota tetap terwakili. Untuk memperoleh estimasi statistik yang lebih stabil, perhitungan CER dilengkapi dengan proses *bootstrapping* sebanyak 2000 iterasi untuk menghasilkan nilai *mean*, *standard deviation*, dan *confidence interval* (CI) 95%.

Untuk menentukan apakah perbedaan performa antara kedua engine signifikan secara statistik, digunakan uji *Wilcoxon Signed-Rank Test* pada seluruh mode prapemrosesan. Melalui pendekatan ini, analisis memberikan pemahaman komprehensif mengenai pengaruh prapemrosesan terhadap akurasi dan stabilitas OCR pada citra nota yang kompleks.

##### 4.2.1. Tabel Perbandingan CER Tesseract dan EasyOCR

**Tabel 1.** Perbandingan CER Tesseract dan EasyOCR

Mode Prapemrosesan	Tesseract CER	EasyOCR CER	p-value	Makna Statistik
original	0.1895	0.1952	0.5299	Tidak signifikan
grayscale	0.1947	0.1936	0.1117	Tidak signifikan
threshold	0.2211	0.2521	0.0057	signifikan
denoise	0.1897	0.2087	0.0150	signifikan

##### A. Analisis Performa Tesseract

Berdasarkan nilai CER yang ditampilkan pada Tabel 1, Tesseract menunjukkan sensitivitas yang cukup tinggi terhadap perubahan prapemrosesan, sehingga kualitas input visual sangat memengaruhi performanya. Pada mode *original*, Tesseract menghasilkan nilai CER sebesar 0.1895 yang menjadi salah satu performa terbaiknya. Nilai ini sekaligus menunjukkan bahwa Tesseract bekerja optimal ketika citra nota dipertahankan dalam kondisi alami tanpa modifikasi drastis. Stabilitas performa pada mode ini juga terlihat dari interval kepercayaan yang cukup sempit, sehingga variasi kesalahan relatif tidak besar antar-sampel.

Namun, ketika citra dikonversi ke *grayscale*, CER justru sedikit meningkat menjadi 0.1947. Hal ini menunjukkan bahwa penghilangan informasi warna tidak memberikan keuntungan signifikan. Bahkan, beberapa komponen visual seperti perbedaan kontras natural antara tinta dan latar justru hilang, sehingga segmentasi karakter menjadi kurang stabil. Kondisi ini juga tercermin pada peningkatan standar deviasi yang mengindikasikan bertambahnya variabilitas kesalahan.

Dampak terburuk terlihat pada prapemrosesan *threshold*, di mana CER meningkat tajam menjadi 0.2211 nilai tertinggi

untuk Tesseract. Proses binarisasi Otsu mengubah citra menjadi hitam-putih secara ekstrem dan cenderung merusak bentuk huruf tipis, mempertebal noise dari latar belakang, serta menghilangkan detail pixel-level yang penting bagi proses segmentasi. Hal inilah yang menjadikan thresholding tidak hanya menurunkan akurasi, tetapi juga menyebabkan kesalahan yang cenderung konsisten buruk di seluruh sampel.

Sebaliknya, prapemrosesan denoise memberikan peningkatan yang signifikan, mengembalikan CER ke nilai 0.1897, hampir setara dengan mode original. Gaussian blur mampu menekan noise frekuensi tinggi seperti bintik hasil pemindaian atau tekstur kertas tanpa menghilangkan struktur huruf, sehingga model LSTM pada Tesseract dapat mengenali karakter secara lebih stabil. Dengan demikian, denoise dapat dipandang sebagai metode prapemrosesan yang paling optimal untuk Tesseract. Temuan tersebut diperkuat oleh distribusi nilai CER yang lebih homogen dan interval kepercayaan yang kembali berada pada rentang performa terbaik sebagaimana ditunjukkan pada Tabel 1.

### B. Analisis Performa EasyOCR

Jika dibandingkan dengan Tesseract, EasyOCR menunjukkan performa yang jauh lebih stabil pada berbagai prapemrosesan, sebagaimana terlihat pada nilai-nilai CER dalam Tabel 1. Pada mode original, EasyOCR memperoleh CER sebesar 0.1952, yang hanya berbeda sedikit dari Tesseract. Ketika citra diubah ke grayscale, nilai CER bahkan menurun sedikit menjadi 0.1936. Perbedaan kecil ini menunjukkan bahwa EasyOCR, yang berbasis arsitektur gabungan CRAFT untuk deteksi teks dan CRNN untuk pengenalan karakter, tidak terlalu bergantung pada informasi warna dan lebih mengutamakan pola struktural huruf.

Robustness EasyOCR terlihat dari *standard deviation* yang relatif kecil, menandakan bahwa engine ini mampu

mempertahankan performa stabil pada nota dengan variasi kondisi pencahayaan, ketebalan tinta, dan noise pemindaian. Hal ini jauh lebih baik dibandingkan Tesseract yang memiliki variabilitas kesalahan lebih besar akibat ketergantungan kuat pada segmentasi piksel.

Namun, pada prapemrosesan threshold, performa EasyOCR mengalami penurunan paling drastis dengan CER meningkat menjadi 0.2521. Walaupun EasyOCR lebih baik daripada Tesseract dalam menangani variasi citra kompleks, proses binarisasi ekstrem membuat algoritma CRAFT kesulitan mendeteksi area teks secara presisi. Hilangnya detail kontur huruf menyebabkan CRNN tidak memperoleh fitur yang cukup untuk melakukan pengenalan karakter secara akurat.

Pada prapemrosesan denoise, EasyOCR kembali menunjukkan stabilitas dengan nilai CER sebesar 0.2087. Gaussian blur mengurangi noise, tetapi smoothing berlebihan membuat beberapa tepi huruf menjadi kurang tajam, sehingga berdampak pada penurunan kualitas fitur yang diekstraksi oleh backbone CNN. Meskipun begitu, performanya tetap lebih baik daripada threshold dan distribusi kesalahan relatif stabil.

Secara keseluruhan, EasyOCR terbukti lebih robust terhadap variasi prapemrosesan dibandingkan Tesseract, namun tidak mendapatkan manfaat signifikan dari prapemrosesan agresif seperti threshold atau smoothing besar, sebagaimana terlihat dari pola CER pada Tabel 1.

### C. Analisis Hasil Uji *Wilcoxon Signed-Rank Test*

Uji statistik *Wilcoxon Signed-Rank Test* digunakan untuk menentukan apakah perbedaan performa antara Tesseract dan EasyOCR signifikan pada masing-masing mode prapemrosesan. Berdasarkan p-value pada Tabel 1, dua mode pertama yaitu original ( $p = 0.5299$ ) dan grayscale ( $p = 0.1117$ ) menunjukkan bahwa tidak terdapat perbedaan signifikan antara kedua engine. Hal ini



mengindikasikan bahwa ketika citra nota diproses secara minimal atau hanya melalui konversi intensitas, kedua OCR memiliki kemampuan pengenalan teks yang relatif setara dan stabil.

Sebaliknya, perbedaan performa menjadi signifikan pada prapemrosesan threshold ( $p = 0.0057$ ) dan denoise ( $p = 0.0150$ ). Pada threshold, penurunan performa Tesseract sangat tajam sehingga jarak CER antara kedua engine meningkat drastis, menghasilkan nilai  $p$ -value yang jauh di bawah 0.05. Hal ini menunjukkan bahwa metode binarisasi memberikan dampak negatif yang tidak proporsional terhadap Tesseract jika dibandingkan EasyOCR. Pada mode denoise, performa Tesseract kembali meningkat, sementara EasyOCR justru mengalami penurunan ringan. Kondisi inilah yang menciptakan gap performa yang cukup besar sehingga perbedaannya menjadi signifikan secara statistik.

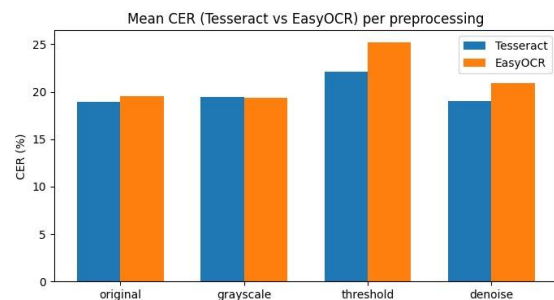
Dengan demikian, uji Wilcoxon mengonfirmasi bahwa prapemrosesan tidak hanya memengaruhi akurasi absolut, tetapi juga mengubah pola kompetitif kedua engine OCR, terutama pada skenario threshold dan denoise yang memiliki karakteristik visual ekstrem atau sangat halus.

#### 4.3. Visualisasi Perbandingan Akurasi OCR

Visualisasi hasil pengujian akurasi OCR digunakan untuk memberikan gambaran yang lebih intuitif terhadap perbedaan performa antara Tesseract dan EasyOCR pada setiap metode prapemrosesan yang diterapkan. Penyajian dalam bentuk grafik batang memungkinkan pembaca untuk dengan mudah mengamati pola perubahan nilai *Character Error Rate* (CER) serta melihat secara langsung pengaruh masing-masing teknik prapemrosesan terhadap kedua engine OCR. Dengan demikian, visualisasi ini berfungsi sebagai pendukung analisis kuantitatif yang telah disajikan dalam bentuk tabel pada subbab sebelumnya.

##### 4.3.1. Grafik Batang Perbandingan CER

Gambar 3 menampilkan grafik batang yang menunjukkan perbandingan nilai rata-rata CER antara Tesseract dan EasyOCR pada empat kondisi prapemrosesan, yaitu original, grayscale, threshold, dan denoise. Sumbu horizontal merepresentasikan jenis prapemrosesan, sedangkan sumbu vertikal menunjukkan nilai CER dalam satuan persentase, yang mencerminkan tingkat kesalahan pengenalan karakter. Setiap pasangan batang menggambarkan performa kedua engine OCR pada kondisi prapemrosesan yang sama, sehingga memudahkan analisis perbandingan secara visual.



**Gambar 3.** Grafik Batang Perbandingan Mean CER Tesseract dan EasyOCR pada Setiap Metode Prapemrosesan.

Berdasarkan Gambar 3, terlihat bahwa pada kondisi original, performa Tesseract dan EasyOCR relatif seimbang dengan nilai CER masing-masing sebesar 18,95% dan 19,52%. Hal ini menunjukkan bahwa pada citra nota tanpa prapemrosesan tambahan, kedua engine OCR mampu mengenali karakter dengan tingkat akurasi yang hampir setara. Kondisi serupa juga terjadi pada prapemrosesan grayscale, di mana nilai CER Tesseract sebesar 19,47% dan EasyOCR sebesar 19,36%, yang kembali memperlihatkan bahwa konversi ke skala keabuan tidak memberikan perbedaan performa yang signifikan bagi kedua engine.

Perbedaan yang paling mencolok terlihat pada metode threshold, di mana terjadi peningkatan nilai CER yang cukup tajam untuk kedua OCR, khususnya pada EasyOCR dengan nilai CER mencapai 25,21%, sedangkan Tesseract berada pada 22,11%. Lonjakan kesalahan ini secara visual tampak jelas melalui

batang grafik yang jauh lebih tinggi dibandingkan kondisi lainnya. Fenomena ini mengindikasikan bahwa proses binarisasi dengan metode Otsu cenderung menghilangkan detail halus pada karakter, merusak kontur huruf, serta memperparah efek noise pada citra nota, sehingga berdampak negatif terhadap proses deteksi dan pengenalan teks.

Pada kondisi denoise, nilai CER kembali mengalami penurunan dibandingkan threshold. Tesseract menunjukkan nilai CER sebesar 18,97%, yang hampir sama dengan kondisi original, sedangkan EasyOCR berada pada 20,87%. Secara visual, perbedaan ini mengindikasikan bahwa prapemrosesan Gaussian blur lebih menguntungkan bagi Tesseract dibandingkan EasyOCR. Pengurangan noise yang dihasilkan oleh Gaussian blur membantu Tesseract dalam melakukan segmentasi karakter dengan lebih stabil, sedangkan pada EasyOCR efek smoothing yang berlebihan justru dapat mengurangi ketajaman tepi huruf yang dibutuhkan oleh model CRNN untuk ekstraksi fitur.

Secara keseluruhan, Gambar 4 memperkuat hasil analisis numerik pada Tabel 1, bahwa threshold merupakan metode prapemrosesan yang paling menurunkan akurasi OCR, sedangkan denoise menjadi prapemrosesan yang paling optimal bagi Tesseract. Selain itu, EasyOCR menunjukkan karakteristik yang lebih stabil pada variasi prapemrosesan ringan seperti original dan grayscale, namun tetap sensitif terhadap prapemrosesan ekstrem seperti thresholding.

#### 4.4. Analisis Pengaruh Prapemrosesan terhadap Kualitas Citra Nota

Analisis pengaruh prapemrosesan dilakukan untuk mengidentifikasi perubahan kualitas visual citra nota setelah penerapan tiga teknik praproses citra, yaitu grayscale, threshold, dan denoise dengan citra original digunakan sebagai kondisi pembanding (baseline). Kajian ini berfokus pada aspek-aspek visual yang relevan terhadap kinerja OCR, termasuk kejelasan

bentuk karakter, tingkat kontras antara teks dan latar belakang, keberadaan noise, serta ketajaman tepi huruf yang memengaruhi proses deteksi dan pengenalan teks. Melalui analisis ini, dapat dinilai sejauh mana setiap metode prapemrosesan berkontribusi dalam meningkatkan atau menurunkan keterbacaan teks pada citra nota.

Gambar 4 menyajikan perbandingan visual hasil prapemrosesan untuk seluruh metode yang diuji, sehingga memungkinkan observasi langsung terhadap perubahan karakteristik citra pada masing-masing pendekatan.



**Gambar 4.** Perbandingan Visual Hasil Prapemrosesan

##### 4.4.1. Original

Berdasarkan Gambar 4(a), citra *original* mempertahankan seluruh informasi visual asli, termasuk warna nota, tekstur kertas, garis tabel, serta variasi ketebalan tulisan. Bentuk karakter, jarak antar huruf, dan batas antara teks dan latar belakang masih terjaga secara alami tanpa distorsi akibat pemrosesan citra. Kondisi ini menguntungkan bagi sistem OCR karena struktur karakter dapat dikenali secara utuh. Namun, pada citra *original* masih terlihat keberadaan noise hasil pemindaian, bayangan lipatan, serta variasi pencahayaan yang berpotensi mengganggu proses segmentasi karakter.

##### 4.4.2. Grayscale

Hasil prapemrosesan *grayscale* yang ditunjukkan pada Gambar 4(b) memperlihatkan bahwa seluruh informasi warna dihilangkan dan direpresentasikan dalam intensitas keabuan. Perubahan ini meningkatkan kontras antara teks dan latar belakang pada sebagian besar area citra. Akan tetapi, pada beberapa bagian, garis tabel dan teks memiliki intensitas yang saling mendekati sehingga batas antar objek menjadi kurang jelas. Meskipun demikian, representasi

struktur karakter pada mode *grayscale* masih relatif stabil dan tetap mendukung proses pengenalan karakter oleh sistem OCR.

#### 4.4.3. *Threshold*

Gambar 4(c) memperlihatkan hasil prapemrosesan *threshold* yang mengubah citra menjadi hitam-putih secara ekstrem melalui proses binarisasi. Metode ini menyebabkan hilangnya banyak detail halus, seperti tulisan tipis, bayangan, serta variasi intensitas tinta. Selain itu, latar belakang yang seharusnya homogen sering kali berubah menjadi noise acak. Akibatnya, beberapa karakter tampak terputus, menebal secara tidak wajar, atau menyatu dengan objek lain. Distorsi bentuk karakter ini sangat menyulitkan proses deteksi dan pengenalan karakter oleh sistem OCR.

#### 4.4.4. *Denoise*

Hasil prapemrosesan *denoise* pada Gambar 4(d) menunjukkan bahwa noise berfrekuensi tinggi berhasil dikurangi secara signifikan, sehingga latar belakang citra menjadi lebih halus dan bersih dibandingkan citra *original*. Kondisi ini membantu meningkatkan keterbacaan teks. Namun, efek *smoothing* akibat penerapan *Gaussian blur* juga menyebabkan beberapa tepi huruf menjadi kurang tajam. Pada karakter dengan ukuran kecil dan ketebalan tipis, efek blur yang berlebihan berpotensi mengurangi kejelasan bentuk karakter dan dapat menurunkan akurasi pengenalan teks.

### 5. KESIMPULAN

- a. Evaluasi kinerja Tesseract dan EasyOCR pada pengenalan teks citra nota menunjukkan hasil yang bervariasi pada setiap metode prapemrosesan, dengan sebagian besar teks dapat dikenali dengan baik, namun masih ditemukan kesalahan pengenalan terutama pada citra dengan kualitas rendah dan hasil *threshold*.
- b. Nilai *Character Error Rate* (CER) yang diperoleh berada pada rentang 0,18 hingga 0,25, dengan performa terbaik dicapai Tesseract pada mode *original* dengan CER sebesar 0,1895 dan EasyOCR pada mode *grayscale* dengan CER sebesar 0,1936. Hasil ini menunjukkan bahwa tingkat akurasi pengenalan kedua engine berada pada kisaran 75%–82%.

- c. Evaluasi kinerja ini dipengaruhi oleh kualitas citra nota, jenis prapemrosesan, serta karakteristik masing-masing engine OCR. Metode *threshold* terbukti menurunkan akurasi secara signifikan, sedangkan metode *denoise* memberikan peningkatan performa khususnya pada Tesseract. Meskipun EasyOCR lebih stabil, namun tetap sensitif terhadap prapemrosesan ekstrem.
- d. Saran untuk penelitian selanjutnya adalah menggunakan variasi teknik prapemrosesan yang lebih beragam seperti *adaptive thresholding* dan *contrast enhancement*, memperluas dataset dengan jenis dokumen dan bahasa yang berbeda, serta menambahkan metrik evaluasi lain seperti *Word Error Rate* (WER) dan waktu komputasi untuk memperoleh analisis kinerja OCR yang lebih komprehensif.

### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

### DAFTAR PUSTAKA

- [1] A. Rexhepi, E. Hasi, A. Haxholli, and E. Bytyçi, "Invoice and receipt optical character recognition: review on current methods and future trends".
- [2] M. Kumar, Shalu, A. Dureja, R. Narula, Shyla, and R. Arora, "OCR-CRNN (WBS): an optical character recognition system based on convolutional recurrent neural network embedded with word beam search decoder for extraction of text," *Int. J. Inf. Technol.*, vol. 17, no. 7, pp. 4013–4020, Sept. 2025, doi: 10.1007/s41870-025-02540-x.
- [3] A. Akoushideh, A. Ranjkesh Rashtehroudi, and A. Shahbahrani, "Persian/Arabic Scene Text Recognition With Convolutional Recurrent Neural Network," *IET Smart Cities*, vol. 7, no. 1, p. e70001, Jan. 2025, doi: 10.1049/smc2.70001.
- [4] A. Yadav, S. Singh, M. Siddique, N. Mehta, and A. Kotangale, "OCR using CRNN: A Deep Learning Approach for Text Recognition," in *2023 4th International Conference for Emerging Technology (INCET)*, Belgaum, India: IEEE, May 2023, pp. 1–6. doi: 10.1109/INCET57972.2023.10170436.
- [5] K. Maliński and K. Okarma, "Analysis of Image Preprocessing and Binarization

- Methods for OCR-Based Detection and Classification of Electronic Integrated Circuit Labeling,” *Electronics*, vol. 12, no. 11, p. 2449, May 2023, doi: 10.3390/electronics12112449.
- [6] H. T. Ha and A. Horák, “Information extraction from scanned invoice images using text analysis and layout features,” *Signal Process. Image Commun.*, vol. 102, p. 116601, Mar. 2022, doi: 10.1016/j.image.2021.116601.
- [7] D. R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, “Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm,” in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, Coimbatore, India: IEEE, Dec. 2022, pp. 966–971. doi: 10.1109/ICECA55336.2022.10009215.
- [8] A. Aprilino and I. H. Al Amin, “Implementasi Algoritma YOLO dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis,” *J. Teknoinfo*, vol. 16, no. 1, p. 54, Jan. 2022, doi: 10.33365/jti.v16i1.1522.
- [9] R. I. Indrakusuma, A. S. Ahmadiyah, and N. F. Ariyani, “Pengenalan dan Klasifikasi Tulisan pada Nota Pembelian Material (Studi Kasus Proyek Konstruksi),” *J. Tek. ITS*, vol. 10, no. 2, pp. A478–A483, Dec. 2021, doi: 10.12962/j23373539.v10i2.77109.
- [10] U. P. Sanjaya, Z. Alawi, A. R. Zayn, and G. P. Dirgantoro, “Optimasi Convolutional Neural Network dengan Standard Deviasi untuk Klasifikasi Pneumonia pada Citra X-rays Paru,” *Gener. J.*, vol. 7, no. 3, pp. 40–47, Oct. 2023, doi: 10.29407/gj.v7i3.20183.
- [11] K. A. Nugraha, “Penerapan Optical Character Recognition untuk Pengenalan Variasi Teks pada Media Presentasi Pembelajaran”.
- [12] O. Rahmdani, “Evaluasi Kinerja Tesseract-OCR dalam Pengenalan Teks Tulisan Tangan Menggunakan Dataset Kustom,” *J. Inform. Dan Tek. Elektro Terap.*, vol. 13, no. 3, July 2025, doi: 10.23960/jitet.v13i3.7162.
- [13] Muhammad Hafizh Husein and Imelda Imelda, “Deteksi Pelanggar Garis Marka Pada Traffic Light Dengan Metode Haar Cascade Dan Easyocr,” *J. Ticom Technol. Inf. Commun.*, vol. 13, no. 1, pp. 45–49, Sept. 2024, doi: 10.70309/ticom.v13i1.141. [7] S. A. Nugroho, N. Kholis, Endryansyah, and F. Baskoro, Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM
- [15] F. Hidayat, “Perbandingan Kualitas Hasil Cetakan Nota Struk Belanja Menggunakan Metode Optical Character Recognition (OCR) pada Capstone Project Kantongin.”
- [16] S. Kumar, Computer Science And Engineering Department, ABES Institute Of Technology, Ghaziabad, India., R. Jaiswal, Prof. S. Kumar, and Computer Science And Engineering Department, ABES Institute Of Technology, Ghaziabad, India., “Improve OCR Accuracy with Advanced Image Preprocessing using Machine Learning with Python,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 7, pp. 1026–1030, May 2020, doi: 10.35940/ijitee.G5745.059720.
- [17] S. Sutriawan, A. Kurniawan, and R. Rosyid, “Improving the Quality of Optical Character Recognition (OCR) Based on Neural Network with the Image Enhancement Process,” *Sci. J. Comput. Sci. Inform.*, vol. 1, no. 2, pp. 58–67, July 2024, doi: 10.34304/scientific.v1i2.334.
- [18] I. Singh, M. Colom, and K. Bontcheva, “A Comparative Analysis of OCR Models on Diverse Datasets: Insights from Memes and Hiertext Dataset”.
- [19] R. H. Sukarna, F. Safira, and H. D. Marzuliyanti, “Studi Komparatif Model OCR Berbasis AI untuk Dokumen Cetak dan Tulisan Tangan,” 2025.
- [20] N. Shabbir, R. Ahmed, M. W. Raza, A. Zeb, H. Elahi, and H. Waheed, “Comparative Performance and Resource Utilization Analysis of OCR Models for Number Plate Recognition on Raspberry Pi 4,” in *2025 International Conference on Communication Technologies (ComTech)*, Rawalpindi, Pakistan: IEEE, Apr. 2025, pp. 1–6. doi: 10.1109/ComTech65062.2025.11034455.
- [21] Beatriz Martínez Tornés *et al.*, “Find it again! Dataset.” Kaggle. doi: 10.34740/KAGGLE/DSV/11192869.