JKTi Jurnal Keilmuan Teknologi informasi

Volume 02, No. 1, Juni 2026, Hal. 13 – 19
e-ISSN : 3109-4163

Perbandingan Kinerja EasyOCR dan Tesseract dalam Ekstraksi Teks pada citra Digital

Abyan Hanif[1*], Ardan Sahid[2], Alifia Anita F[3]

[*1, 2, 3] Program Studi Teknologi Informasi, Fakultas Kesehatan dan Teknologi, Universitas Muhammadiyah Klaten, Klaten

Email: [1]abyanhanif@gmail.com, [2]ardansahid9@gmail.com, [3]alifiaanita28@gmail.com
[*]abyanhanif79@gmail.com

ABSTRACT — Rapid advancements in digital image processing technology have increased the demand for automatic text extraction systems from images, commonly known as OCR (Optical Character Recognition). In this field, there are two widely used tools: *Tesseract* and *EasyOCR*, which are two very popular open-source software packages frequently utilized by developers to meet their needs. However, selecting the most appropriate tool often poses a unique challenge for researchers due to significant differences in the underlying architecture and performance offered by each library. This study aims to conduct an in-depth comparative analysis between Tesseract and EasyOCR, specifically regarding character recognition accuracy and data processing speed under various image conditions. The methodology employed in this study involves collecting a diverse dataset of images, ranging from very clean printed text to images with significant visual noise or distortion. Both software programs will then be tested using the Python programming language to systematically and measurably extract text from the same dataset. Performance evaluation is measured objectively using the Weighted Average Character Error Rate (CER) and Word Error Rate (WER) metrics.

KEYWORDS — OCR; EasyOCR; Tesseract; Text Extraction; Deep Learning; Image Processing; CER; WER;

INTISARI — Perkembangan terhadap pengolahan teknologi citra digital yang perkembang sangat pesat telah meningkatkan kebutuhan kepada sistem ekstraksi teks otomatis dari sebuah gambar atau biasa dikenal sebagai (OCR). Di teknologi ini terdapat dua tools yaitu *Tesseract* dan *EasyOCR*, yang merupakan dua perangkat lunak bersifat sumber terbuka yang sangat populer dan seringkali digunakan oleh para pengembang untuk memenuhi kebutuhan yang diperlukan. Namun, pemilihan alat yang paling tepat sering kali menjadi tantangan tersendiri yang harus dihadapi oleh peneliti karena adanya perbedaan signifikan dalam arsitektur dasar serta performa yang ditawarkan oleh masing-masing pustaka tersebut. Penelitian ini dilakukan bertujuan untuk melakukan analisis perbandingan yang mendalam antara *Tesseract* dan *EasyOCR*, khususnya dalam aspek akurasi pengenalan karakter serta kecepatan pemrosesan data pada berbagai kondisi citra. Metodologi yang digunakan dalam penelitian ini meliputi pengumpulan dataset citra yang bervariasi, mulai dari teks cetak yang sangat bersih hingga gambar yang memiliki gangguan visual atau noise yang cukup tinggi. Kedua perangkat lunak tersebut kemudian akan diuji menggunakan bahasa pemrograman Python untuk mengekstraksi teks dari dataset yang sama secara sistematis dan terukur. Evaluasi kinerja diukur secara objektif menggunakan metrik *Weighted Average Character Error Rate* (CER) dan *Word Error Rate* (WER).

KATA KUNCI — OCR; EasyOCR; Tesseract; Ekstraksi Teks; Deep Learning; Pengolahan Citra; CER; WER;

I. PENDAHULUAN

Perkembangan teknologi pengolahan citra digital yang pesat telah meningkatkan urgensi sistem *Optical Character Recognition* (OCR) dalam mendukung efisiensi operasional di berbagai sector [1]. Sistem Tesseract yang dikelola oleh Google telah lama menjadi standar industri dengan memanfaatkan teknologi *Long Short-Term Memory* (LSTM) [1]. Meskipun demikian, beberapa studi menunjukkan bahwa performa pengenalan karakter pada Tesseract cenderung mengalami penurunan drastis ketika dihadapkan pada objek dengan kompleksitas visual yang tinggi atau label kemasan yang tidak standar [1]. Oleh karena itu, diperlukan evaluasi lebih lanjut terhadap efektivitas penggunaannya dalam kondisi non-standar.

Beberapa penelitian terdahulu telah berupaya membandingkan kinerja antar-pustaka OCR menggunakan metrik *Character Error Rate* (CER) dan *Word Error Rate* (WER) sebagai standar evaluasi tingkat akurasi[2], [3], [4], [5], [6]. Pustaka Tesseract diketahui unggul pada dokumen formal terstruktur dengan latar belakang bersih. Sementara itu, EasyOCR yang berbasis kerangka kerja *deep learning* PyTorch menawarkan ketangguhan pada *scene text recognition* melalui integrasi model *Character Region Awareness for Text Detection* (CRAFT) [7]. Model CRAFT ini terbukti handal dalam menangani pembentukan *bounding box* lokasi teks pada aksara non-Latin maupun teks dengan variasi sudut yang ekstrem [7], [8].

Meskipun perbandingan umum telah banyak dibahas, masih terdapat celah penelitian (*research gap*) terkait analisis performa mendalam saat kedua mesin ini dihadapkan pada dataset dengan degradasi kualitas citra yang ekstrem secara berdampingan. Kebaruan (*novelty*) dari penelitian ini terletak pada analisis komprehensif yang mengevaluasi tingkat akurasi sekaligus efisiensi pemrosesan pada lingkungan pengujian yang dirancang secara khusus [9].

Penelitian ini bertujuan untuk menganalisis perbandingan objektif guna memberikan landasan data bagi praktisi dalam menentukan alat OCR yang paling sesuai dengan kebutuhan aplikasi [3] [10] [11] [12]. Fokus pengujian mencakup aspek akurasi pada tingkat karakter dan kata serta kecepatan pemrosesan data. Kontribusi utama dari hasil penelitian ini adalah pemberian rekomendasi teknis mengenai penggunaan mesin OCR yang optimal pada lingkungan statis maupun dinamis. Dengan demikian, penelitian ini diharapkan dapat membantu menjaga integritas data digital yang dihasilkan dari proses ekstraksi teks otomatis di berbagai kondisi lingkungan.

II. TINJAUAN PUSTAKA

EasyOCR memanfaatkan algoritma deep learning untuk pengenalan gambar jaringan saraf konvolusional seperti *Residual Network* yang dipadukan dengan *recurrent neural networks*, terutama jaringan LSTM *(Long Short Term Memory)* dan CTC *(Connectionist Temporal Classification*), sebagai blok bangunan arsitekturnya. EasyOCR juga menggunakan *Algoritma Greedy Decoder* sebelum melakukan post processing dan CRAFT (*Character Region Awareness For Text Recognition*) setelah melakukan pre processing[3].algoritma CRAFT cukup akurat dalam pengenalan text dengan pembentukan bounding box yang membantu memberikan lokasi letak text [7], easyOCR memiliki struktur yang cukup kompleks dan karena mengandalkan algoritma *neural network* menyebabkan tingginya permintaan daya komputasi walaupun EasyOCR cukup baik dengan gambar dengan kualitas sedang dan rendah.

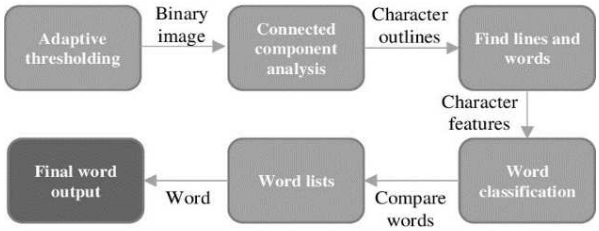
Tessaract menggunakan struktur yang lebih sederhana dibandingkan dengan EasyOCR,Tessaract hanya menggunakan LSTM (*Long Short Term Memory*) dan *language Model correction*[11],dan karena struktur yang linear dan algoritma yang lebih sedikit menyebabkan kecepatan tinggi walaupun daya komputasi yang lebih sedikit ,namun memiliki kelemahan terhadap gambar dengan kerumitan dan kualitas yang rendah.

A. *OPTICAL CHARACTER RECOGNITION (OCR)*

Optical Character Recognition (OCR) adalah teknologi yang berfungsi untuk mengonversi dokumen fisik dalam bentuk citra digital seperti hasil pemindaian atau foto menjadi teks yang dapat diedit dan diproses oleh mesin komputer[12]. Proses kerja OCR secara umum melibatkan beberapa tahap utama, yaitu prapemrosesan (*preprocessing*), segmentasi karakter, ekstraksi fitur, dan pengenalan karakter . Teknologi ini telah menjadi komponen inti dalam proses digitalisasi informasi untuk kebutuhan arsip, ekstraksi data otomatis, hingga pengembangan sistem kecerdasan buatan yang mampu memahami konten visual. Efektivitas sebuah sistem OCR sangat bergantung pada kemampuan algoritma dalam menangani variasi jenis huruf, resolusi citra, dan gangguan visual (*noise*) [13].

B. *ARSITEKTUR TESSERACT*

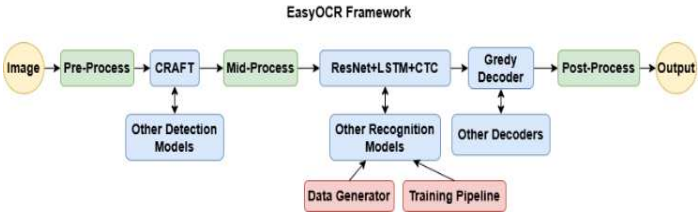
Tesseract merupakan salah satu mesin OCR sumber terbuka yang paling luas penggunaannya dan saat ini dikembangkan oleh Google [14]. Sejak peluncurannya, Tesseract telah mengadopsi arsitektur Neural Network berbasis *Long Short-Term Memory* (LSTM) untuk meningkatkan akurasi pengenalan teks yang bersifat sekuensial. Keunggulan utama Tesseract terletak pada dukungannya terhadap lebih dari 100 bahasa dan kemampuannya yang efisien dalam mengenali teks pada dokumen dengan format standar atau terstruktur. Namun, performa Tesseract diketahui sangat sensitif terhadap kualitas gambar; tanpa prapemrosesan yang memadai seperti thresholding atau perbaikan kontras, tingkat akurasinya cenderung menurun pada citra dengan latar belakang yang kompleks.



Gambar 1. Arsitektur TESSERACT

C. *ARSITEKTUR EASYOCR*

EasyOCR adalah pustaka OCR modern berbasis bahasa pemrograman Python yang dibangun di atas kerangka kerja *deep learning* PyTorch. Pustaka ini dikembangkan oleh Jaided AI dan dirancang untuk mempermudah implementasi ekstraksi teks pada berbagai platform. Arsitektur EasyOCR menggunakan model *Character Region Awareness for Text Detection* (CRAFT) untuk mendeteksi lokasi teks pada citra dan *Convolutional Recurrent Neural Network* (CRNN) untuk mengenali karakter tersebut. Berbeda dengan pendekatan tradisional, EasyOCR dikenal lebih tangguh (*robust*) dalam menangani *scene text recognition*, yaitu teks yang berada di lingkungan alami dengan sudut pengambilan gambar yang beragam, pencahayaan yang tidak merata, serta gaya tulisan yang bervariasi.



Gambar 2. Arsitektur EasyOCR

D. *PARAMETER CER & WER*

Untuk mengukur kinerja dan akurasi dari sistem OCR, digunakan metrik standar berupa *Character Error Rate* (CER) dan *Word Error Rate* (WER). CER dihitung berdasarkan jumlah operasi minimum yang diperlukan untuk mengubah teks hasil ekstraksi menjadi teks referensi (termasuk substitusi, penghapusan, dan penyisipan karakter), dibagi dengan jumlah total karakter . Sementara itu, WER menggunakan logika yang sama namun diterapkan pada tingkat kata. Nilai CER dan WER yang mendekati angka 0 menunjukkan tingkat akurasi yang semakin tinggi. Penggunaan rata-rata tertimbang (*Weighted Average*) pada kedua metrik ini sering kali dilakukan untuk memberikan gambaran yang lebih representatif terhadap performa sistem pada dataset dengan distribusi panjang teks yang berbeda [2].

Berikut persamaan (1)(2) untuk menghitung CER dan WER:

$$CER = \frac{(S+D+I)}{NH} \quad (1)$$

$$WER = \frac{(S+D+I)}{NK} \quad (2)$$

Keterangan variabel:

- **S**: Jumlah substitusi (penggantian kata atau karakter yang salah)
- **D**: Jumlah penghapusan (kata atau karakter yang hilang dari teks asli)
- **I**: Jumlah penyisipan (kata atau karakter tambahan yang tidak seharusnya ada)
- **NH**: Jumlah total **karakter** pada teks referensi (*ground truth*) untuk perhitungan CER.
- **NK**: Jumlah total **kata** pada teks referensi (*ground truth*) untuk perhitungan WER.

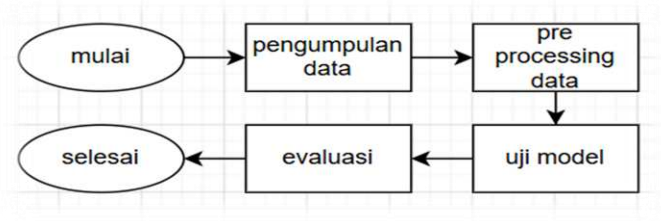
Semakin rendah nilai CER, semakin baik kualitas ekstraksi yang dihasilkan. Semakin rendah nilai WER, semakin baik kualitas ekstraksi per kata yang dihasilkan. Selain itu, Akurasi dihitung untuk memberikan nilai persentase seberapa tepat *output* OCR dibandingkan dengan teks referensi. Berikut persamaan (3) untuk menghitung akurasi:

$$Akurasi = (1 - CER)x\ 100 \quad (3)$$

III. METODOLOGI

A. *ALUR PENELITIAN*

Penelitian ini dilakukan secara sistematis melalui beberapa tahapan utama untuk menjamin validitas data yang dihasilkan. Tahapan pertama dimulai dengan pengumpulan dataset citra yang mengandung teks dengan variasi tingkat kesulitan. Tahap kedua adalah perancangan dan implementasi dua program AI menggunakan bahasa pemrograman Python, di mana program pertama mengimplementasikan mesin Tesseract OCR dan program kedua mengimplementasikan pustaka EasyOCR. Tahap ketiga adalah proses ekstraksi teks secara otomatis dari seluruh dataset menggunakan kedua program tersebut. Tahapan terakhir adalah evaluasi hasil ekstraksi dengan membandingkan teks *output* terhadap teks referensi (*ground truth*) untuk mendapatkan nilai akurasi berdasarkan metrik yang telah ditentukan. Berikut Gambar 3 merupakan contoh alur penelitian dalam bentuk diagram alir atau flowchart.



Gambar 3. Alur penelitian

B. *DATA UJI*

Data uji (*test set*) yang digunakan dalam penelitian ini terdiri dari sampel citra digital yang disiapkan secara mandiri (*synthetic dataset*) oleh peneliti. Pembatasan jumlah sampel ini sengaja dilakukan secara terukur guna memfasilitasi analisis kualitatif dan kuantitatif secara mendalam, khususnya pada studi kasus ekstraksi teks berspesifikasi khusus yang membutuhkan ketelitian tinggi.

Proses pengumpulan, kriteria pemilihan, dan karakteristik dari sampel citra tersebut dijabarkan secara terperinci sebagai berikut:

1) Sumber data

Citra teks diproduksi secara digital oleh peneliti menggunakan perangkat lunak penyunting teks dengan variasi *font* standar. Langkah ini diambil untuk memastikan bahwa teks awal memiliki struktur geometri yang ideal sebelum diberikan perlakuan gangguan visual tertentu.

2) Karakteristik Kluster Gangguan:

Untuk menguji batas ekstrem performa deteksi arsitektur *Character Region Awareness for Text Detection* (CRAFT) pada EasyOCR serta mesin berbasis LSTM pada Tesseract, sampel citra dibagi menjadi tiga kluster tingkat kesulitan berdasarkan standarisasi variasi *noise* pada citra digital [11], [15]:

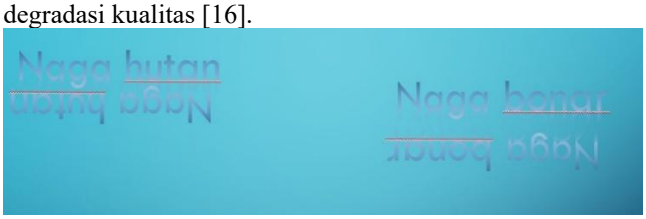
- **Kluster I (Kontrol):** Citra teks cetak bersih tanpa gangguan visual sebagai standar performa dasar (*baseline*).

Gambar 4. Contoh citra kluster I (Kontrol)

- **Kluster II (Sedang):** Citra dengan efek pencahayaan tidak merata (*uneven illumination*) dan latar belakang kontras rendah (*low contrast*) guna mensimulasikan gangguan optik pada pemindaian fisik.

Gambar 5. Contoh citra kluster II (Sedang)

- **Kluster III (Ekstrem):** Citra dengan tingkat derau tinggi (*high noise*) dan resolusi rendah (*low-resolution text* di bawah 72 DPI). Variasi ini mengacu pada tantangan nyata pengenalan karakter pada dokumen digital yang mengalami degradasi kualitas [16].



Gambar 6. Contoh citra kluster III (Ekstrem)

3) Teks Referensi (Ground Truth):

Setiap sampel citra merepresentasikan representasi kata-kata spesifik yang kompleks seperti biodata kartu pramuka. Seluruh teks asli pada citra telah dicatat secara manual sebagai *ground truth*.

Data *ground truth* ini berfungsi sebagai standar pembanding mutlak dalam evaluasi dan perhitungan nilai *Character Error Rate* (CER) serta *Word Error Rate* (WER) pada tahap pembahasan.

C. LINGKUNGAN PENGEMBANGAN

Prosedur pengujian dilakukan dengan menjalankan kedua program AI pada dataset yang sama secara berdampingan tanpa memberikan perlakuan prapemrosesan (*preprocessing*) manual yang berlebihan, guna melihat performa asli (*baseline performance*) dari masing-masing mesin. Karakteristik visual dan visualisasi dari sampel citra teks yang diuji dalam penelitian ini diilustrasikan secara mendalam pada Gambar A. Hasil ekstraksi teks dari *Tesseract* dan *EasyOCR* kemudian disimpan dalam format digital untuk dihitung tingkat kesalahannya.

Evaluasi akurasi dilakukan menggunakan metode perhitungan *Character Error Rate* (CER) dan *Word Error Rate* (WER) [2], [4]. Perhitungan ini dilakukan secara otomatis dengan membandingkan jumlah karakter atau kata yang salah—mencakup aspek substitusi (*substitution*), penghapusan (*deletion*), atau penyisipan (*insertion*)—terhadap total karakter atau kata pada teks referensi (*ground truth*). Hasil akhir disajikan dalam bentuk rata-rata tertimbang (*weighted average*) untuk mendapatkan kesimpulan performa yang menyeluruh [2].

IV. HASIL DAN PEMBAHASAN

A. HASIL

Penelitian ini menggunakan data uji yang telah disiapkan dan mendapatkan hasil seperti pada Gambar 7 serta Tabel I dibawah.

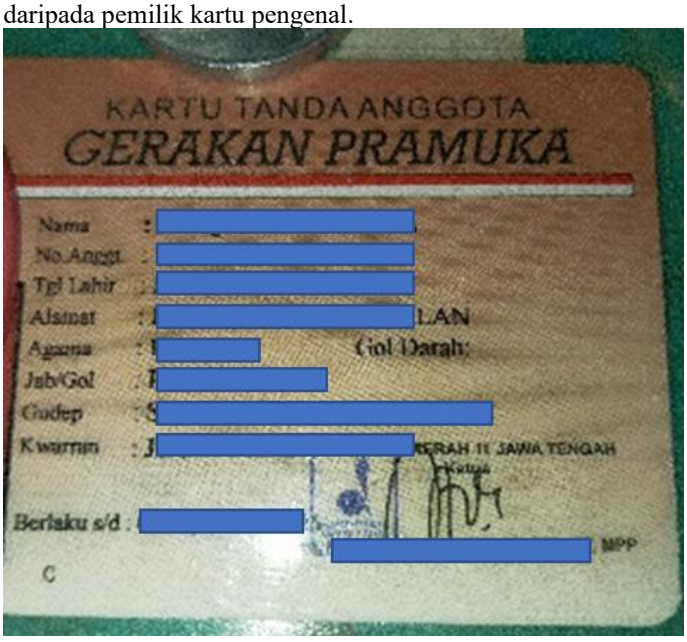
Gambar 7. Data ujicoba menggunakan struk pembelanjaan

TABEL I
HASIL EKSTRAK TEKS EASYOCR PADA DATA UJI COBA STRUK PEMBELANJAAN

Teks Ekstrak	Hasil deteksi
Lnre 4a	0.00
[	0.56
egenda Burger Klaten	0.67
cempaka	0.98
No	0.52
11 ,	0.97
Ngepos,	0.93
Klaten,	0.66
Jawa	1.00
Tengah	1.00
No, 192487-1	0.66
2026-05-05	1.00
Kasir	1.00
{	0.51
KASIR	1.00
B	1.00
15,07:15	0.62
In	0.00
Pe	0.02
Baee	0.01
1.	0.58
Cilik	1.00
1 X Rp 12. 500	0.61
Rp	1.00

Pada sampel citra ini, Tesseract mengalami kegagalan total dalam mendeteksi dan mengekstraksi teks, sehingga tidak ada *output* teks yang dihasilkan.

Selanjutnya dilakukan juga ujicoba pada objek lain yaitu kartu pengenal seperti yang terlihat pada Gambar 8 dan Tabel II dan III untuk hasil dari pada uji coba yang dilakukan. Namun pada Gambar 8 diberikan *noise* untuk menjaga privasi daripada pemilik kartu pengenal.



Gambar 8. Data ujicoba menggunakan kartu pengenal

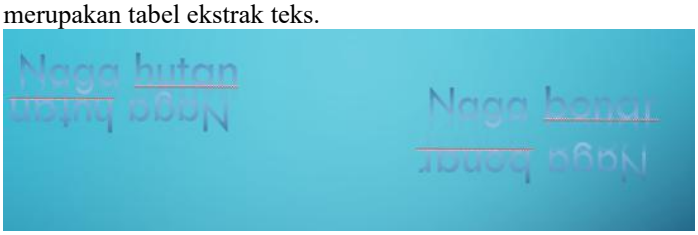
TABEL II
HASIL EKSTRAK TEKS EASYOCR PADA DATA UJI COBA KARTU PENGENAL

Teks Ekstrak	Hasil deteksi
KARTU	1.00
TANDA ANGGOTA	0.79
GERAKAN PRAMUKA	0.85
PENGGALANG	0.86
Gudep	1.00
SMP NEGERI 1 JOGONALAN	0.95
Kwarran	0.95
JOGONALANIWARTIRPAFRAH	0.48
11 JAWA TENGAH	
tety	0.09

TABEL III
HASIL EKSTRAK TEKS TESSERACT PADA DATA UJI COBA KARTU PENGENAL

Teks Ekstrak	Hasil deteksi
Pk	0.07
See	0.25
Swen	0.04
eee	0.03
Se	0.43
lect	0.11
ee	0.19
GR	0.29
EC	0.25
=	0.66
ae	0.06
a	0.44
ae	0.12
Seen:	0.16

Selain itu data yang diuji cobakan juga menggunakan jenis teks yang berbeda dari ujicoba sebelumnya seperti yang terlihat pada Gambar 9 dibawah serta Tabel IV dan V merupakan tabel ekstrak teks.



Gambar 9. Data ujicoba menggunakan gambar berisi tulisan

TABEL IV
HASIL EKSTRAK TEKS EASYOCR PADA DATA UJI GAMBAR BERISI TULISAN

Teks Ekstrak	Hasil deteksi
Naga hutan	0.44
Naga Lamaf	0.17

TABEL V
HASIL EKSTRAK TEKS TESSERACT PADA DATA UJI GAMBAR BERISI TULISAN

Teks Ekstrak	Hasil deteksi
Rla	0.13
awa	0.13
aoe	0.29
KlAAA	0.55
Law	0.57
wep	0.07
nny	0.19
i	0.39

Selain itu data yang diuji cobakan juga menggunakan variasi objek jenis teks yang berbeda dari ujicoba sebelumnya seperti yang terlihat pada gambar 10, serta Tabel VI merupakan tabel ekstrak teks.

Gambar 10. Data ujicoba menggunakan foto helm

TABEL VI
HASIL EKSTRAK TEKS EASYOCR PADA DATA UJI COBA FOTO HELM

Teks Ekstrak	Hasil deteksi
7iu	0.80
H E L M E T	0.65
OSAKHI	1.00
HELMET	0.80
OC y	0.23
100+	0.43
OK	0.18

Pada sampel citra juga ini, Tesseract mengalami kegagalan total dalam mendeteksi dan mengekstraksi teks, sehingga tidak ada *output* teks yang dihasilkan.

Selain itu data yang diuji cobakan juga menggunakan variasi jenis teks yang berbeda dari ujicoba sebelumnya seperti yang terlihat pada gambar 11 dibawah serta Tabel VII DAN VIII merupakan tabel ekstrak teks.

Gambar 11. Data ujicoba menggunakan gambar berisi tulisan resolusi rendah

TABEL VII
HASIL EKSTRAK TEKS EASYOCR PADA DATA UJI GAMBAR BERISI TULISAN

Teks Ekstrak	Hasil deteksi
Tuchs	0.69
@ohr	0.28

TABEL VIII
HASIL EKSTRAK TEKS TESSERACT PADA DATA UJI GAMBAR BERISI TULISAN

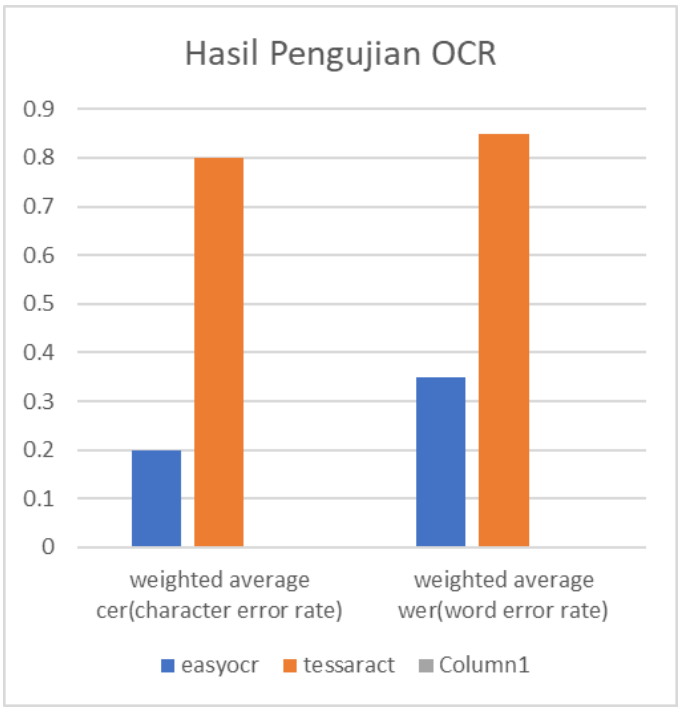
Teks Ekstrak	Hasil deteksi
om	0.95

B. *PEMBAHASAN*

Hasil penelitian ini memberikan implikasi penting bagi pengembang sistem berbasis AI. Untuk kebutuhan ekstraksi teks pada media statis seperti dokumen pindaian yang bersih, Tesseract masih merupakan pilihan ekonomis. Namun, untuk aplikasi yang beroperasi di lingkungan dinamis seperti pembacaan nota transaksi melalui kamera atau pelat nomor kendaraan, penggunaan EasyOCR sangat direkomendasikan demi menjaga integritas data digital yang dihasilkan.

Penelitian yang dilakukan mendapatkan hasil ekstraksi Tesseract menunjukkan kegagalan pada kata 'hallo', 'tugas' dan angka '8' yang sama sekali tidak terdeteksi dalam output teks. Selain itu, pada kata 'nomor', Tesseract menghasilkan nilai kepercayaan yang sangat tinggi, yaitu 95, jauh di diatas tiga kata lainnya. Fenomena ini memperkuat temuan kuantitatif bahwa Tesseract memiliki ketergantungan yang tinggi pada kejelasan piksel dan segmentasi karakter, sehingga pada citra *low-res*, mesin ini cenderung melewatkan teks atau menghasilkan pembacaan yang tidak akurat.

Prosedur pengujian dilakukan dengan mengeksekusi setiap sampel citra pada kedua mesin OCR secara sistematis. Untuk memastikan stabilitas sistem dan mengonfirmasi sifat deterministik dari arsitektur EasyOCR dan Tesseract, dilakukan pengujian replikasi sebanyak 10 kali iterasi untuk sampel citra yang sama. Hasil eksperimen menunjukkan tingkat konsistensi sebesar 100% tanpa adanya variasi pada output teks maupun nilai kepercayaan (*confidence score*) di setiap iterasinya. Setelah konsistensi performa terbukti stabil, nilai akhir performa dihitung menggunakan metode *Character Error Rate* (CER) dan *Word Error Rate* (WER). Perbandingan akumulatif kinerja dari kedua model tersebut disajikan pada Gambar 12 dibawah.



Gambar 12. Data perbandingan kinerja easyocr dan tessaract

Gambar 12 merepresentasikan perbandingan hasil perhitungan CER dan WER dari EasyOCR dan Tesseract menggunakan rumus (1) untuk CER dan rumus (2) untuk WER

Nilai Weighted Average CER dan WER yang disajikan pada Tabel II diperoleh melalui proses perhitungan matematis menggunakan persamaan metrik evaluasi yang telah diuraikan pada bagian metodologi. Secara spesifik, setiap output teks yang dihasilkan oleh EasyOCR dan Tesseract dari ke-19 sampel data uji dibandingkan secara langsung dengan teks referensi aslinya (ground truth). Tingkat kesalahan kemudian dikalkulasi berdasarkan akumulasi jumlah substitusi (S), penghapusan (D), dan penyisipan (I), yang dibagi dengan total jumlah karakter atau kata aktual (N). Hasil perhitungan dari seluruh sampel tersebut kemudian dirata-rata secara tertimbang (weighted average) untuk merepresentasikan skor akhir performa deteksi masing-masing mesin secara objektif.

TABEL VIII
HASIL EKSTRAK TEKS PADA DATA UJI GAMBAR BERISI TULISAN

Kluster Pengujian	EasyOCR CER	EasyOCR WER	Tesseract CER	Tesseract WER
Kluster I (Kontrol)	4,8192%	11,11%	16,26%	29,62%
Kluster II (Sedang)	15,602%	28%	40,42%	56%
Kluster III (Ekstrem)	18,28%	39,13%	99,20%	99,37%

1) *ANALISIS CHARACTER ERROR RATE (CER) BERDASARKAN KLUSTER*

Berdasarkan data pada Tabel VIII, ditemukan tren pergeseran nilai CER yang kontras antara kedua mesin seiring meningkatnya kompleksitas degradasi citra input.

- **Kluster 1**: Kedua pustaka menunjukkan performa yang fungsional. EasyOCR mencatatkan kesalahan karakter yang sangat minim, yaitu sebesar 4,82%, disusul oleh Tesseract dengan nilai CER sebesar 16,26%. Hal ini menunjukkan bahwa pada citra teks cetak yang bersih dan memiliki kontras seimbang, arsitektur berbasis *Long Short-Term Memory* (LSTM) milik Tesseract mampu melakukan pemetaan karakter secara memadai.
- **Kluster 2**: Ketika citra mulai diinterfensi oleh gangguan pencahayaan tidak merata, nilai CER Tesseract meningkat signifikan menjadi 40,42%, sedangkan EasyOCR mampu menahan laju kesalahan pada angka 15,60%.
- **Kluster 3**: Pada kluster gangguan ekstrem (*high noise* dan resolusi rendah), terjadi divergensi performa yang sangat masif. Tesseract mengalami kelumpuhan total dengan nilai CER menyentuh 99,20%. Skor ini mengindikasikan bahwa hampir seluruh piksel karakter gagal diidentifikasi dengan benar. Kegagalan fatal Tesseract disebabkan oleh sensitivitas tahap binerisasi awalnya yang memaksakan piksel noise menjadi bagian dari teks, memicu insertion error yang tinggi. Sebaliknya, EasyOCR menunjukkan ketangguhan luar biasa

dengan nilai CER yang hanya berada di angka 18,28%. Model *Character Region Awareness for Text Detection* (CRAFT) pada EasyOCR berhasil mempertahankan lokalisasi teks lewat prediksi heatmaps yang fleksibel, sehingga tidak mudah terkecoh oleh degradasi piksel yang ekstrem.

2) *ANALISIS WORD ERROR RATE (WER) BERDASARKAN KLUSTER*

Analisis tingkat kesalahan kata (WER) mempertegas superioritas pemrosesan sekuensial EasyOCR dibandingkan metode pembatasan spasi tradisional milik Tesseract.

- **Kluster 1** dan **Kluster 2**: EasyOCR secara konsisten menjaga integritas kata dengan nilai WER sebesar 11,11% pada kondisi kontrol dan 28,00% pada kondisi sedang. Di sisi lain, kesalahan kata pada Tesseract melonjak dari 29,62% menjadi 56,00% pada Kluster 2. Lonjakan ini mendeteksi adanya kelemahan analisis tata letak (*layout analysis*) Tesseract ketika menghadapi spasi spasial yang tidak seragam akibat distorsi citra.
- **Kluster 3:** Pada kondisi paling ekstrem, Tesseract mencatatkan WER sebesar 99,37%, yang berarti akurasi ekstraksi kata bernilai hampir nol. Tesseract melakukan over-segmentation secara masif, memecah satu kata utuh menjadi beberapa fragmen karakter acak yang tidak bermakna. Sebaliknya, arsitektur *Convolutional Recurrent Neural Network* (CRNN) pada EasyOCR mampu menjaga konteks baris teks sebagai satu kesatuan rangkaian (*sequence*), sehingga nilai WER pada Kluster 3 dapat ditekan secara optimal pada angka 39,13%.

3) *PERBANDINGAN ARSITEKTUR DAN EFISIENSI*

Perbedaan hasil yang kontras ini berakar pada perbedaan teknologi inti. EasyOCR memanfaatkan *deep learning* berbasis PyTorch yang lebih fleksibel dalam mengekstraksi fitur visual yang kompleks. Sementara itu, Tesseract versi 4/5, meskipun sudah menggunakan LSTM, masih sangat bergantung pada kualitas prapemrosesan gambar tradisional (binerisasi).

Namun, perlu diperhatikan adanya trade-off antara akurasi dan sumber daya. EasyOCR membutuhkan memori GPU yang lebih besar dan waktu pemrosesan yang relatif lebih lama per citra dibandingkan Tesseract yang sangat ringan dan cepat jika dijalankan pada CPU standar.

V. KESIMPULAN

Penelitian ini telah berhasil melakukan analisis perbandingan performa antara EasyOCR dan Tesseract dalam ekstraksi teks pada citra digital. Berdasarkan hasil pengujian menggunakan metrik rata-rata tertimbang, dapat disimpulkan bahwa EasyOCR memiliki keunggulan performa yang sangat signifikan dibandingkan Tesseract pada dataset yang diuji. EasyOCR mencatatkan nilai *Character Error Rate* (CER) sebesar 0,16 dan *Word Error Rate* (WER) sebesar 0,35. Sementara itu, Tesseract menunjukkan tingkat kesalahan yang jauh lebih tinggi dengan nilai CER 0,80 dan WER 0,88.

REFERENSI

[1] C. Wick, C. Reul, and F. Puppe, "Calamari − A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition," *Digital Humanities Quarterly*, vol. 14, no. 2, Jun. 2020, doi: 10.63744/wnzk3f3ymuez.

[2] S. Nagaonkar, A. Sharma, A. Choithani, and A. Trivedi, "Benchmarking Vision-Language Models on Optical Character Recognition in Dynamic Video Environments," *ArXiv*, vol. abs/2502.06445, 2025, [Online]. Available: https://api.semanticscholar.org/CorpusID:276250355

[3] A. N. Chowdhury, A. A. Sami, S. M. P. Mamun, S. Absar, F. Rahman, and Md. S. R. Kohinoor, "Performance Analysis of Tesseract and EasyOCR for Bangla Optical Character Recognition on the Novel Bangla CrossHair Dataset," in *2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, IEEE, Feb. 2025, pp. 716–721. doi: 10.1109/ISACC65211.2025.10969286.

[4] M. Nagayi, A. S. Khan, T. Frank, R. Swart, and C. Nyirenda, "Evaluating OCR performance on food packaging labels in South Africa," *ArXiv*, vol. abs/2510.03570, 2025, [Online]. Available: https://api.semanticscholar.org/CorpusID:281843012

[5] E. Purushotham, R. Kasarapu, and B. C. Shoba, "Benchmarking open-source OCR engines on semantic slide regions in educational videos using a subset of FITVID dataset," *i-manager's Journal on Future Engineering and Technology*, vol. 20, no. 4, p. 9, 2025, doi: 10.26634/jfet.20.4.22287.

[6] K. C. Kirana, I. Kumalasari, G. Q. Oktagalu, A. Maqbullah, A. F. Shobari, and B. Hidayat, "Comparison of Tesseract OCR, Easy OCR, and Transformer OCR on Handwritten Image," in *2025 9th International Conference On Electrical, Electronics And Information Engineering (ICEEIE)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/ICEEIE66203.2025.11252079.

[7] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character Region Awareness for Text Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 9357–9366. doi: 10.1109/CVPR.2019.00959.

[8] P. Kiatphaisansophon, D. Wanvarie, and N. Cooharojananone, "Efficient Text Bounding Box Identification Using Mask R-CNN: Case of Thai Documents," *IEEE Access*, vol. 12, pp. 49306–49328, 2024, doi: 10.1109/ACCESS.2024.3383911.

[9] N. M. Xi and J. Li, "Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3646565.

[10] M. A. Naqi Hadi, M. Gul, M. Khan, G. N. Alwakid, and N. Zaman Jhanjhi, "Benchmarking Performance Analysis of Optical Character Recognition Techniques," in *2024 26th International Multi-Topic Conference (INMIC)*, IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/INMIC64792.2024.11004392.

[11] Qiaomu Zhang, "Adaptive OCR Engine Selection and Evaluation for Multi-Format Government Document Digitization," *Artificial Intelligence and Machine Learning Review*, vol. 7, no. 1, pp. 29–39, Jan. 2026, doi: 10.69987/AIMLR.2026.70103.

[12] T. Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment," *J. Comput. Soc. Sci.*, vol. 5, no. 1, pp. 861–882, May 2022, doi: 10.1007/s42001-021-00149-1.

[13] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[14] K. Inoue, "Context-Independent OCR with Multimodal LLMs: Effects of Image Resolution and Visual Complexity," *ArXiv*, vol. abs/2503.23667, 2025, [Online]. Available: https://api.semanticscholar.org/CorpusID:277452503

[15] Muhammad Nashiruddin, F. D. Noor Praditya, Agiel Faiz Mufazzal, and A. Ardiansyah, "Implementation of Tesseract-based OCR for UMKLA student card data extraction," *JKTI Jurnal Keilmuan Teknologi Informasi*, vol. 1, no. 1, pp. 11–14, Jun. 2025, doi: 10.61902/jkti.v1i1.1688.

[16] P. Schönfelder, F. Stebel, N. Andreou, and M. König, "Deep learning-based text detection and recognition on architectural floor plans," *Autom. Constr.*, vol. 157, p. 105156, Jan. 2024, doi: 10.1016/j.autcon.2023.105156.

[17] S. Gonwirat and O. Surinta, "DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters," *IEEE Access*, vol. 10, pp. 90133–90148, 2022, doi: 10.1109/ACCESS.2022.3201560.