

Comparative Analysis of Google Vision OCR with Tesseract on Newspaper Text Recognition

Bintang Timur Kusmanto¹, Nurcahya Pradana Taufik Prakisyah^{*2}, Puspanada Hatta³

^{1,2,3}Universitas Sebelas Maret, Indonesia

e-mail: ¹b.timur504@student.uns.ac.id, ^{*2}nurcahya.ptp@staff.uns.ac.id,

³hatta.puspanada@staff.uns.ac.id

Abstract

Optical Character Recognition (OCR) is a technique used to convert image files into machine-readable text. There are two Optical Character Recognition (OCR) algorithms that are currently well known and widely used, namely Google Vision's Optical Character Recognition (OCR) and Tesseract. The purpose of this study is to compare the Optical Character Recognition (OCR) algorithms of Google Vision and Tesseract so that people can more easily find out which algorithm is the right one to implement on the system they are going to build. The method used in this research is Research and Development (R&D) with the stages of literature study, needs analysis, dataset collection and expansion, architectural design development and application modeling, system implementation, testing and evaluation, drawing conclusions. To be able to determine the level of accuracy, precision and sensitivity of each algorithm, this research uses the Confusion Matrix formula. The results of this study conclude that Google Vision's Optical Character Recognition (OCR) algorithm is superior to Tesseract because the level of accuracy, sensitivity, and precision is superior to Google Vision.

Keywords : Comparative Analysis of OCR, Google Vision, Optical Character Recognition, Tesseract

Abstrak

Optical Character Recognition (OCR) merupakan sebuah teknik yang digunakan untuk mengubah file gambar menjadi sebuah teks yang dapat dibaca oleh mesin. Terdapat dua algoritma Optical Character Recognition (OCR) yang saat ini terkenal dan banyak digunakan yaitu Optical Character Recognition (OCR) Google Vision dan Tesseract. Tujuan dari penelitian ini adalah membandingkan algoritma Optical Character Recognition (OCR) Google Vision dan Tesseract sehingga orang dapat lebih mudah mengetahui mana algoritma yang tepat untuk diimplementasikan pada sistem yang akan dibangunnya. Metode yang digunakan pada penelitian ini adalah Research and Development (R&D) dengan tahapan studi literatur, analisis kebutuhan, pengumpulan dan perluasan dataset, pembangunan desain arsitektur dan pemodelan aplikasi, implementasi sistem, uji coba dan evaluasi, penarikan kesimpulan. Untuk dapat mengetahui tingkat akurasi, presisi dan sensitivitas dari masing-masing algoritma penelitian ini menggunakan rumus Confusion Matrix. Hasil dari penelitian ini menyimpulkan bahwa algoritma Optical Character Recognition (OCR) Google Vision lebih unggul jika dibandingkan dengan Tesseract karena tingkat akurasi, sensitivitas, dan presisi lebih unggul Google Vision.

Kata kunci : Analisis Perbandingan OCR, Google Vision, Optical Character Recognition, Tesseract

1. PENDAHULUAN

Digital image processing adalah sebuah bidang yang akan terus mengalami pertumbuhan yang pesat, dengan aplikasi di banyak bidang aspek kehidupan kita [1]. Salah satunya pengenalan teks secara otomatis yang berperan sangat penting dalam pengenalan identifikasi teks dalam sebuah foto yang berisi dokumen. Pengembangan aplikasi pengenalan teks dalam foto akan mempunyai dampak yang cukup penting terutama dalam membantu proses pendigitalisasian dokumen-dokumen lama yang penting [2].

Image processing (pengolahan citra) merupakan teknik mutakhir tentang pemrosesan gambar digital dengan penekanan pada pendekatan pemrosesan dengan algoritma terkait [3]. *Image processing* berfungsi untuk memproses atau memanipulasi sebuah gambar atau foto yang berbentuk 2 dimensi [3]. Metode *image processing* tersebut antara lain seperti *computer aided diagnosis*, *edge detection*, *image segmentation* [4]. Teknik-teknik tersebut dapat dipakai untuk meningkatkan hasil dari pengenalan teks pada koran. Selain digunakan untuk *Optical Character Recognition*, *Image Processing* dapat diimplementasikan kedalam aplikasi lain seperti *face recognition*

Dalam dunia digital *Optical Character Recognition (OCR)* adalah salah satu topik yang menarik dan menantang dalam pengenalan pola [5]. Pengenalan karakter dicapai melalui proses segmentasi, fitur ekstraksi dan klasifikasi [6]. *Optical Character Recognition (OCR)* melibatkan beberapa proses binerisasi gambar ketika sedang memproses pengenalan teks atau karakter yang terdapat dalam gambar. Dalam pemrosesan juga terdapat beberapa teknik pembantu yang bertujuan untuk mengenali karakter yaitu *normalize* dan *line and word detection*.

Google Vision merupakan salah satu algoritma yang dapat memudahkan orang untuk melakukan pengolahan gambar. Google Vision mempunyai banyak fitur dalam pemrosesan gambar (*Image Processing*) yang salah satunya adalah *Optical Character Recognition (OCR)*. Dengan adanya Google Vision kita dapat melakukan proses pengenalan teks dalam gambar dan mengekstrak karakter yang terdapat dalam gambar secara langsung.

Selain Google Vision terdapat juga algoritma yang serupa yang dapat melakukan pengenalan karakter yang terdapat di dalam sebuah gambar yaitu, Tesseract. Tesseract sendiri merupakan algoritma yang bersifat *open source*. Pada awalnya Tesseract dikembangkan di Hewlett-Packard pada tahun 1985-1995.

Kaur, R., & Sharma, D. V. pada tahun 2021 meneliti terkait perbandingan *OCR Cloud Vision API* dan Tesseract dengan membuat aplikasi *smartphone* dengan model *prototype* untuk studi kasus *Punjabi Text* [7]. Penelitian ini menerapkan model yang sama yaitu *prototype* untuk membandingkan *OCR Google Vision* dan Tesseract. Perbedaannya adalah penelitian ini membangun aplikasi berbasis web dan studi kasus pada penelitian ini adalah teks koran. Manuel, I. J. (2018) meneliti terkait dengan perbandingan *OCR Google Vision* dengan Tesseract pada plat nomor kendaraan pribadi di Indonesia dengan menggunakan aplikasi *smartphone* dengan model *prototype* dan akurasi dihitung dari per karakter yang terdeteksi [8]. Penelitian ini menerapkan model yang sama yaitu *prototype* untuk membandingkan *OCR Google Vision* dan Tesseract. Perbedaannya adalah penelitian ini membangun aplikasi berbasis web, studi kasus pada penelitian ini adalah teks koran, serta tingkat akurasi dihitung per kata. Dhiman, S., & Singh, A. (2013) meneliti terkait perbandingan *OCR GOCR* dan Tesseract dengan membuat *prototyping* berbentuk *Command User Interface (CUI)* untuk studi kasus teks pada komputer [9]. Penelitian ini menerapkan model yang sama yaitu *prototype* untuk membandingkan *OCR Google Vision* dan Tesseract. Perbedaannya adalah penelitian ini membangun aplikasi berbasis web, studi kasus pada penelitian ini adalah teks koran.

Penelitian terakhir yang dilakukan oleh Kaur, R., & Sharma, D. V. tahun 2021 *OCR Google Vision* mempunyai keunggulan akurasi yang lebih tinggi jika dibandingkan dengan Tesseract [7]. Maka dari itu peneliti berasumsi bahwa hasil tingkat akurasi dari Google Vision juga lebih tinggi dibandingkan dengan Tesseract. Namun peneliti belum dapat menyimpulkan hal tersebut secara langsung dikarenakan penelitian yang dilakukan oleh Kaur, R., & Sharma, D. V. [7] dilakukan pada studi kasus *Punjabi Text* sedangkan pada penelitian yang akan dilakukan

adalah teks koran dengan bahasa Indonesia. Peneliti memilih koran dikarenakan teks yang terdapat didalam koran cukup banyak dan kompleks sehingga bisa dijadikan sebagai acuan untuk studi kasus lain seperti majalah, buku, jurnal dan lain-lain.

Tujuan dari penelitian ini adalah menganalisa hasil perbandingan dari antara kedua algoritma diatas yaitu Google Vision dan Tesseract. Yang dimana Google Vision merupakan algoritma Optical Character Recognition yang dikembangkan oleh Google sedangkan Tesseract merupakan algoritma *Optical Character Recognition* yang bersifat *Open Source*. Selain itu, penelitian ini diharapkan dapat memberikan manfaat kepada orang yang bekerja sebagai *Software Developer* supaya mempunyai dasar yang kuat dalam memilih sebuah algoritma OCR yang akan diterapkan dalam sistem yang akan dikembangkannya. Algoritma OCR ini dapat diterapkan dalam beberapa hal misalnya pada aplikasi *smart parking system* [10], pada sistem digitalisasi dan pencarian dokumen [11], pada sistem akusisi dokumen jabatan fungsional dosen [12], pengenalan nomor ruangan [13], pada dokumen karya tulis ilmiah [14], serta dalam aplikasi-aplikasi lainnya.

2. METODE PENELITIAN

Studi kasus dalam penelitian ini adalah teks yang terdapat dalam koran. Metode dari penelitian ini adalah *Research and Development* atau *R&D* [15]. Maksud dari penggunaan metode *R&D* adalah peneliti membuat aplikasi berbentuk *prototype* kemudian *output* dari aplikasi ini yang nantinya akan diteliti. Tahapan penelitian dari metode *R&D* antara lain studi literatur, analisis kebutuhan, pengumpulan dan perluasan *dataset*, pembangunan desain arsitektur dan pemodelan aplikasi, implementasi sistem, uji coba dan evaluasi, penarikan kesimpulan.

2.1 Studi Literatur

Dilakukan terhadap teori yang mendukung penelitian yang bersumber dari berbagai referensi seperti sumber-sumber media cetak, elektronik, jurnal penelitian, serta sumber lain yang berkaitan dengan materi penulisan ini. Studi literatur juga berfungsi untuk mencari beberapa teori yang mendukung mengenai proses algoritma dari Google Vision dan Tesseract.

2.2 Analisis Kebutuhan

Tahapan ini merupakan tahapan yang penting dalam proses pembuatan sistem aplikasi yang digunakan untuk membandingkan algoritma OCR dari Google Vision dan Tesseract. Berikut adalah gambaran dari kebutuhan fungsional dan non fungsional dari aplikasi ini :

2.2.1 Kebutuhan fungsional

Pada tahapan ini akan dibahas mengenai kebutuhan apa saja yang diperlukan untuk membangun sistem berbasis web. Berikut adalah Tabel 1 daftar kebutuhan fungsional berdasarkan studi literatur yang telah dilakukan:

Tabel 1. Kebutuhan fungsional

Kebutuhan	Deskripsi
Sistem	<ul style="list-style-type: none">- Sistem dapat mendeteksi teks yang ada didalam gambar menggunakan dua algoritma yaitu Google Vision dan Tesseract- Sistem dapat melakukan pengambilan sampel gambar yang ada teksnya, kemudian mengubah gambar tersebut menjadi karakter dan tersimpan di database.

2.2.2 Kebutuhan non fungsional

Pada tahap ini akan membahas mengenai batasan dari sistem serta parameter dari sisi *user* dengan menggunakan prinsip *usability*. Berikut Tabel 2 daftar kebutuhan non fungsional dari aplikasi ini :

Tabel 1. Kebutuhan non fungsional

Kebutuhan	Deskripsi
Sistem Operasi	Windows 11 Home 64-bit
Machine Learning	Google Vision Tesseract
Interface	PHP Laravel 8 Bootstrap

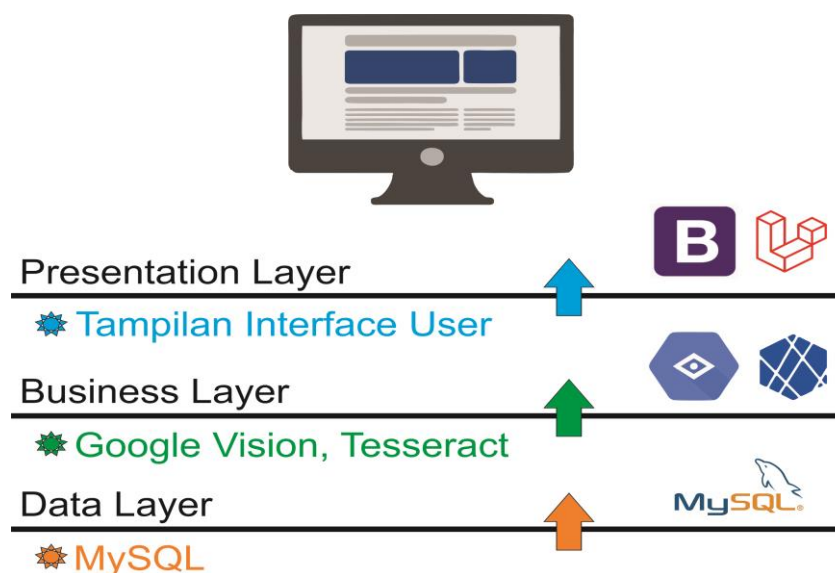
2.3 Pengumpulan dan Perluasan Dataset

Pengumpulan data gambar dilakukan dengan menggunakan kamera berjenis DSLR untuk mendapatkan kualitas gambar yang detail dan bagus. Setiap gambar yang dari DSLR disimpan dan dijadikan sebagai dataset. Jumlah dataset dalam penelitian ini adalah 35 halaman koran yang terdiri dari 30 foto terang dan 5 foto dengan kondisi kurang pencahayaan. Koran tersebut terdiri dari 3 merk yang berbeda yaitu Jawa Pos, Solopos, dan Republika.

2.4 Pembangunan Desain Arsitektur dan Pemodelan Aplikasi

2.4.1 Arsitektur aplikasi

Pada penelitian ini terdapat 3 lapisan arsitektur aplikasi yang terdiri dari *Data layer*, *Business layer*, dan *Presentation layer*. *Data layer* digunakan untuk menyimpan data hasil dari proses *scanning OCR* dengan Google Vision dan Tesseract, pada aplikasi ini *Data layer* menggunakan MySQL. *Business layer* digunakan untuk proses logika dari aplikasi serta proses *Optical Character Recognition (OCR)*, pada proses ini menggunakan dua algoritma yaitu Google Vision dan Tesseract. *Presentation layer* digunakan untuk menangani tampilan antarmuka *user*, pada aplikasi ini menggunakan Laravel 8 sebagai *backend* dan Bootstrap sebagai *frontend*. Berikut adalah Gambar 1 yang merupakan gambaran dari arsitektur aplikasi.



Gambar 1. Arsitektur aplikasi

2.4.2 Pemodelan aplikasi

Pemodelan aplikasi merupakan bentuk model dari aplikasi yang dibuat, aplikasi ini diimplementasikan dalam bentuk *web application*. Berikut adalah bentuk model dari aplikasi:

a. Halaman Google Vision

Gambar 2 di bawah menggambarkan pemodelan aplikasi saat berada di menu Google Vision ketika diakses oleh *user*, dalam *menu* tersebut *user* bisa memasukkan gambar dan kemudian *user* mensubmit gambar tersebut untuk dilakukan proses *scanning OCR* dengan algoritma Google Vision. Pada bagian bawah *form* terdapat riwayat dari proses *scanning* yang dilakukan oleh Google Vision.

Gambar 2. Halaman Google Vision

b. Halaman Tesseract

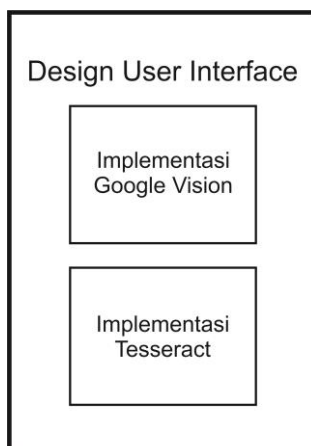
Gambar 3 di bawah menggambarkan pemodelan aplikasi saat berada di *menu* Tesseract ketika diakses oleh *user*, dalam *menu* tersebut *user* bisa memasukkan gambar dan kemudian *user* mensubmit gambar tersebut untuk dilakukan proses *scanning OCR* dengan algoritma Tesseract. Pada bagian bawah *form* terdapat riwayat dari proses *scanning* yang dilakukan oleh Tesseract.

Gambar 3. Halaman Tesseract

2.4 Implementasi Sistem

Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman PHP dengan menggunakan *framework* Laravel 8 kemudian didalam aplikasi tersebut terdapat dua algoritma *Optical Character Recognition (OCR)* yaitu Google Vision dan Tesseract untuk merubah sebuah gambar menjadi sebuah karakter atau teks yang kemudian nantinya akan dibandingkan tingkat akurasi, presisi, sensitivitas serta waktu proses algoritma tersebut berjalan dalam

mendeteksi teks yang ada didalam gambar. Gambar 4 di bawah adalah gambaran mengenai desain dari sistem aplikasi.



Gambar 4. Desain sistem aplikasi

2.5 Uji Coba dan Evaluasi

Pada tahap ini akan dilaksanakan uji coba dan evaluasi terhadap implementasi aplikasi yang sudah dirancang dan dikembangkan. Tahapan ini dilakukan untuk mengetahui apakah aplikasi yang sudah dibuat dapat berjalan dengan baik atau masih perlu beberapa perbaikan dan dijumpai *error*.

Peneliti menggunakan pengujian *Black Box Testing* untuk mengetahui apakah aplikasi sudah dapat digunakan dengan baik atau masih terdapat *error*. Metode *Black Box Testing* merupakan tahapan pengujian sistem berdasarkan tampilan luarnya saja dan tidak memperhatikan *code* yang ada didalam sistem. Metode ini dapat diterapkan pada semua tingkat pengujian perangkat lunak seperti unit, fungsional, integrasi, sistem dan penerimaan pengguna [16]. Adapun rancangan pengujian aplikasi yang akan diuji dengan menggunakan teknik *Black Box Testing* seperti tabel dibawah.

Tabel 2. Rancangan pengujian aplikasi

No	Komponen yang diuji	Butir uji
1	Google Vision	- Submit gambar - Hasil OCR
2	Tesseract	- Submit gambar - Hasil OCR
3	Menu	- Menu Google Vision - Menu Tesseract

2.6 Penarikan Kesimpulan

Pada tahapan ini peneliti mengambil kesimpulan terkait dengan analisis perbandingan dari kedua algoritma *Optical Character Recognition (OCR)* yaitu Google Vision dan Tesseract. Peneliti menghitung tingkat akurasi dan waktu pemrosesan dari kedua algoritma tersebut dan kemudian membandingkannya. Untuk menghitung tingkat akurasi pada masing-masing algoritma peneliti menggunakan *Confusion Matrix*.

Confusion Matrix adalah matriks yang berisi mengenai informasi terkait hasil prediksi klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi [17]. *Confusion Matrix* mempunyai 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Terdapat 4 istilah dalam representasi hasil proses klasifikasi pada *Confusion Matrix* yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Berikut adalah penjelasan dari 4 istilah dalam representasi hasil proses klarifikasi pada *Confusion Matrix* :

a. *True Positive (TP)*

Adalah data positif dan data tersebut diprediksi benar. Dalam kasus penelitian ini yang termasuk ke dalam *True Positive (TP)* adalah teks koran ada dan terdeteksi oleh *OCR* dengan tepat.

b. *True Negative (TN)*

Adalah data negatif dan data tersebut diprediksi benar. Dalam kasus penelitian ini yang termasuk ke dalam *True Negative (TN)* adalah gambar atau objek selain teks tidak terdeteksi ada teks apapun.

c. *False Positive (FP)*

Adalah data negatif dan data tersebut diprediksi sebagai data positif. Dalam kasus penelitian ini yang termasuk ke dalam *False Positive* adalah gambar atau objek selain teks terdeteksi sebagai teks.

d. *False Negative (FN)*

Adalah data positif dan data tersebut diprediksi sebagai data negatif. Dalam kasus penelitian ini yang termasuk ke dalam *False Negative (FN)* adalah teks koran tidak terdeteksi semua atau sebagian huruf saja yang terdeteksi.

Pada penelitian ini peneliti akan menghitung akurasi, presisi, dan sensitivitas dari algoritma *OCR* Google Vision dan Tesseract dengan menggunakan Confusion Matrix. Berikut adalah rumus dan penjelasan singkat terkait dengan akurasi, presisi dan sensitivitas :

- a. Akurasi merupakan gambaran tingkat ketepatan algoritma dalam mengolah sebuah data. Berikut adalah rumusnya [18] :

$$\text{akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- b. Presisi merupakan gambaran tingkat ketepatan prediksi antara data keadaan yang asli (data yang diminta) dengan hasil prediksi yang diberikan oleh sistem. Berikut adalah rumusnya [18] :

$$\text{presisi} = \frac{TP}{TP+FP} \quad (2)$$

- c. Sensitivitas merupakan gambaran tingkat kepekaan algoritma dalam mendeteksi data yang benar dengan data yang tidak sesuai. Berikut adalah rumusnya [18] :

$$\text{sensitivitas} = \frac{TP}{TP+FN} \quad (3)$$

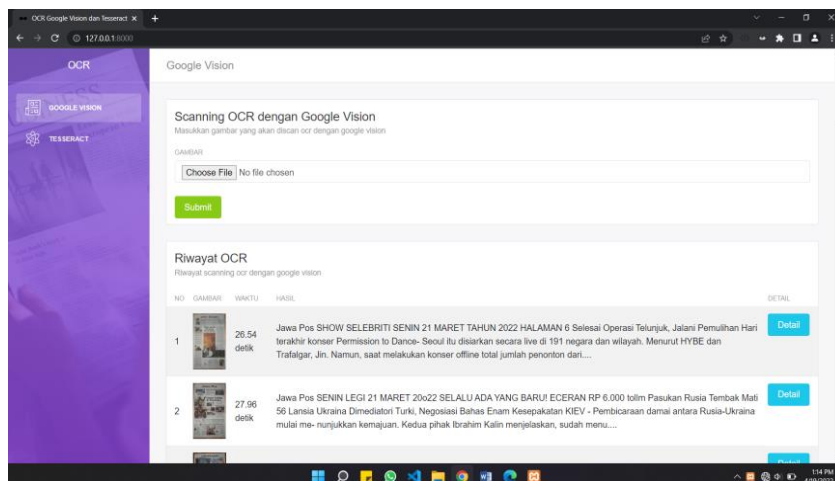
3. HASIL DAN PEMBAHASAN

3.1 Penjabaran Terkait Aplikasi

Pemodelan aplikasi merupakan bentuk model dari aplikasi yang telah dibuat dalam bentuk *wireframe* pada bab sebelumnya, aplikasi perbandingan *OCR* ini dibuat dalam bentuk *web application*. Gambar dibawah menunjukkan bahwa *user* (peneliti) dapat melakukan proses *upload* gambar, melihat detail, dan melihat riwayat hasil *OCR* yang dilakukan oleh Google Vision dan Tesseract.

3.1.1 Halaman antar muka *OCR* Google Vision

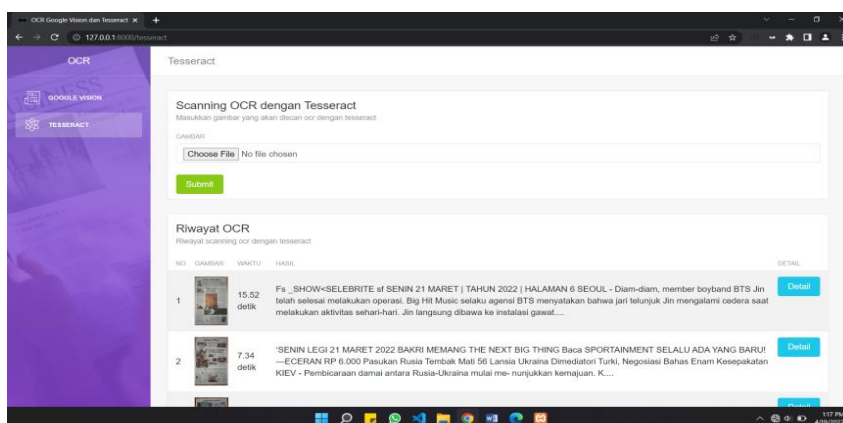
Gambar 5 di bawah menggambarkan aplikasi saat berada di *menu* Google Vision ketika diakses oleh *user*, dalam *menu* tersebut *user* bisa memasukkan gambar dan kemudian *user* mengsubmit gambar tersebut untuk dilakukan proses *scanning OCR* dengan algoritma Tesseract. Pada bagian bawah *form* terdapat riwayat dari proses *scanning* yang dilakukan oleh Google Vision.



Gambar 5. Halaman antar muka OCR Google Vision dalam halaman tersebut terdapat menu untuk upload file foto koran yang akan di OCR dan menu untuk melihat detail hasil ocr

3.1.2 Halaman antar muka OCR Tesseract

Pada Gambar 6 digambarkan aplikasi saat berada di *menu* Tesseract ketika diakses oleh *user*, dalam *menu* tersebut *user* bisa memasukkan gambar dan kemudian *user* mengsubmit gambar tersebut untuk dilakukan proses *scanning OCR* dengan algoritma Tesseract. Pada bagian bawah *form* terdapat riwayat dari proses *scanning* yang dilakukan oleh Tesseract.



Gambar 6. Halaman antar muka OCR Tesseract dalam halaman tersebut terdapat menu untuk upload file foto koran yang akan di OCR dan menu untuk melihat detail hasil ocr

3.2 Uji Coba dan Evaluasi

3.2.1 Black Box Testing

Berdasarkan butir rancangan pengujian aplikasi yang telah dibuat, maka peneliti melakukan pengujian aplikasi sesuai poin-poin yang terdapat dalam rancangan pengujian aplikasi. Pada tahapan ini peneliti meminta tolong dua orang pakar untuk menguji aplikasi ini. Yang pertama adalah Yanuarizal Kurniamanda yang merupakan *backend engineer* di evomo.id dan yang kedua adalah Ellianto Saroni Putro yang merupakan *frontend engineer* di evomo.id. Berikut adalah Tabel 4 hasil *Black Box Testing* yang telah dilakukan dua orang pakar tersebut.

Tabel 3. Hasil *Black Box Testing* yang dilakukan oleh pakar

No	Komponen yang diuji	Butir uji	Hasil
1	Google Vision	- Submit gambar	Sukses
		- Hasil OCR	

No	Komponen yang diuji	Butir uji	Hasil
2	Tesseract	- Submit gambar - Hasil OCR	Sukses
3	Menu	- Menu Google Vision - Menu Tesseract	Sukses

3.2.2 Uji coba hasil OCR Google Vision dan Tesseract

Pada tahapan ini dilakukan uji coba untuk melakukan proses *OCR* Google Vision dan Tesseract. Uji coba dilakukan dengan cara meng-*upload* gambar 1 file koran yang sama kedalam kedua OCR tersebut. Untuk mempermudah dalam proses menghitung tingkat akurasi, presisi, dan sensitivitas dari kedua algoritma tersebut peneliti menggunakan metode penandaan teks koran dengan menggunakan pensil warna. Warna pensil warna yang digunakan untuk menandai teks koran ini terdiri dari hijau untuk TP (True Positive), biru untuk TN (True Negative), kuning untuk FP (False Postive), dan merah untuk FN (False Negative). Peneliti membandingkan antara *ground truth* dengan hasil *output OCR* Google Vision dan Tesseract. *Ground truth* adalah kenyataan yang ada dilapangan dalam penelitian ini *ground truth* merupakan teks asli yang terdapat dalam koran. Metode ini telah divalidasi oleh dua orang teman saya yang merupakan mahasiswa PTIK UNS angkatan 2018 yaitu Arif Wiranata Patanduk dan Brilyan Ramadhan. Gambar 7 menunjukkan foto koran yang digunakan pada saat uji coba aplikasi.



Gambar 7. Foto koran yang dijadikan sebagai uji coba hasil *OCR*

3.3. Penarikan Kesimpulan dari Tahapan Uji Coba

Pada tahapan ini akan dilakukan analisa hasil OCR yang telah dilakukan pada tahapan uji coba aplikasi. Berikut Tabel 5 menampilkan nilai hasil perhitungan OCR Google Vision dan Tesseract yang dihitung dengan menggunakan Confusion Matrix yang dilakukan terhadap foto koran yang telah diupload pada tahapan uji coba aplikasi.

Tabel 4. Jumlah perhitungan kedua algoritma dengan menggunakan *Confusion Matrix*

Google Vision					Tesseract				
Waktu	TP	TN	FP	FN	Waktu	TP	TN	FP	FN
26,54 detik	1595	6	9	24	15,52 detik	1528	7	25	92

Berdasarkan hasil diatas maka dapat dihitung tingkat akurasi, presisi, dan sensitivitas dari masing-masing algoritma menggunakan rumus *Confusion Matrix* yang sudah dijelaskan pada bab sebelumnya. Perbandingan antara Google Vision dengan Tesseract ditunjukkan pada Tabel 6 berikut ini.

Tabel 5. Hasil perhitungan akurasi, presisi, dan sensitivitas menggunakan *Confusion Matrix*

	Google Vision	Persentase	Tesseract	Persentase
Akurasi	0.979804162	97,98%	0.929176755	92,92%
Presisi	0.994389027	99,44%	0.983902125	98,39%)
Sensitivitas	0.985176035	98,52%	0.943209877	94,32%

Berdasarkan tahapan-tahapan diatas yang telah dilakukan penulis, maka dapat disimpulkan bahwa aplikasi perbandingan OCR Google Vision dengan Tesseract sudah dapat digunakan dengan baik. Aplikasi tersebut sudah siap digunakan untuk mengolah data-data foto yang akan dilakukan proses OCR. Untuk membandingkan kedua algoritma tersebut penulis akan mengupload 30 halaman koran supaya lebih terlihat jelas hasil perbandingan antara kedua algoritma tersebut.

3.4 Pembahasan

3.4.1 Tingkat akurasi, presisi, dan sensitivitas dari Google Vision dan Tesseract dalam proses mengolah gambar koran menjadi sebuah kata

Pada tahapan ini peneliti telah melakukan *scanning OCR* terhadap 30 halaman koran yang mempunyai *layout* yang beragam sehingga diharapkan mampu mewakili koran-koran lain yang ada. Hasil tingkat akurasi, presisi dan sensitivitas dari Google Vision dan Tesseract dalam mengolah gambar koran menjadi sebuah kata di perlihatkan pada Tabel 7 dan Tabel 8.

Tabel 6. Hasil *OCR* Google Vision terhadap 35 halaman koran

No. Koran	TP	TN	FP	FN	Akurasi	Presisi	Sensitivitas
1	1595	6	9	24	0.979804162	0.994389027	0.985176035
2	805	8	14	11	0.970167064	0.982905983	0.986519608
3	1793	5	15	40	0.970318403	0.99170354	0.978177851
4	1901	2	12	21	0.982954545	0.99372713	0.989073881
5	1599	5	1	5	0.996273292	0.999375	0.996882793
6	632	13	27	6	0.951327434	0.959028832	0.990595611
7	1798	3	2	13	0.991740088	0.998888889	0.992821645
8	3183	0	3	16	0.994066209	0.99905838	0.994998437
9	1660	10	4	36	0.976608187	0.997596154	0.978773585
10	2557	3	48	25	0.972274972	0.981573896	0.990317583
11	1351	11	58	10	0.952447552	0.958836054	0.992652461
12	1973	2	19	15	0.983076157	0.990461847	0.992454728
13	2069	2	14	6	0.990435198	0.993278925	0.997108434
14	2132	1	30	5	0.983856089	0.986123959	0.997660271
15	2537	1	10	5	0.994124559	0.996073812	0.998033045
16	1643	1	11	6	0.989765202	0.993349456	0.996361431
17	2168	0	30	15	0.979665612	0.986351228	0.993128722

No. Koran	TP	TN	FP	FN	Akurasi	Presisi	Sensitivitas
18	1803	4	5	7	0.993402969	0.997234513	0.996132597
19	2347	2	8	3	0.995338983	0.996602972	0.998723404
20	1687	18	24	7	0.982142857	0.985973115	0.995867769
21	2586	4	18	19	0.985915493	0.993087558	0.992706334
22	2345	0	8	9	0.99280271	0.996600085	0.99617672
23	1586	1	44	3	0.97123623	0.973006135	0.99811202
24	1752	3	8	1	0.994897959	0.995454545	0.999429549
25	1267	2	15	1	0.987548638	0.988299532	0.999211356
26	1057	14	134	3	0.886589404	0.887489505	0.997169811
27	1856	2	2	3	0.997316157	0.998923574	0.998386229
28	823	23	22	12	0.961363636	0.973964497	0.985628743
29	1263	7	16	13	0.977675135	0.987490227	0.989811912
30	1097	1	11	11	0.980357143	0.990072202	0.990072202
31	1598	6	11	13	0.985257985	0.993163456	0.991930478
32	800	9	18	10	0.966547192	0.97799511	0.987654321
33	1791	8	24	45	0.963062099	0.98677686	0.975490196
34	1900	2	11	20	0.983962752	0.994243851	0.989583333
35	1604	6	2	2	0.997521685	0.99875467	0.99875467

Tabel 7. Hasil OCR Tesseract terhadap 35 halaman koran

No. Koran	TP	TN	FP	FN	Akurasi	Presisi	Sensitivitas
1	1528	7	25	92	0.92917676	0.9839021	0.9432099
2	757	11	18	66	0.90140845	0.9767742	0.9198056
3	1755	8	24	114	0.92740663	0.9865093	0.9390048
4	1746	4	26	180	0.89468303	0.9853273	0.9065421
5	1560	6	12	56	0.95838433	0.9923664	0.9653465
6	565	25	106	72	0.76822917	0.8420268	0.8869702
7	1753	3	38	43	0.95590637	0.9787828	0.9760579
8	3089	0	14	110	0.96140678	0.9954882	0.9656143
9	1503	12	38	182	0.87319885	0.9753407	0.8919881
10	2316	5	145	284	0.844	0.9410809	0.8907692
11	1315	11	58	51	0.92404181	0.9577567	0.9626647
12	1890	3	69	106	0.91537718	0.9647779	0.9468938
13	1989	5	22	86	0.94862036	0.9890602	0.9585542
14	1997	1	60	140	0.90900819	0.9708313	0.9344876
15	2394	2	21	153	0.93229572	0.9913043	0.9399293
16	1604	3	11	43	0.96748946	0.9931889	0.9738919
17	2115	3	17	63	0.96360328	0.9920263	0.9710744
18	1702	4	66	108	0.90744681	0.9626697	0.9403315
19	2156	2	62	204	0.89026403	0.9720469	0.9135593
20	1519	23	38	159	0.8867165	0.9755941	0.9052443
21	2349	6	49	208	0.90160796	0.9795663	0.9186547
22	2149	1	25	199	0.90564448	0.9885005	0.915247
23	1525	4	35	68	0.93688725	0.9775641	0.9573132
24	1639	3	31	121	0.91527313	0.9814371	0.93125
25	1247	5	27	21	0.96307692	0.9788069	0.9834385
26	988	16	137	75	0.82565789	0.8782222	0.929445
27	1748	2	46	110	0.9181532	0.974359	0.9407966
28	615	25	141	206	0.64842958	0.8134921	0.7490865
29	1209	7	63	75	0.89807976	0.9504717	0.9415888

No. Koran	TP	TN	FP	FN	Akurasi	Presisi	Sensitivitas
30	1025	3	36	95	0.88697153	0.9660697	0.9151786
31	1530	7	26	89	0.93038741	0.9832905	0.9450278
32	736	11	17	91	0.87368421	0.9774236	0.8899637
33	1644	8	58	207	0.86176317	0.9659224	0.8881686
34	1734	4	25	184	0.89265537	0.9857874	0.9040667
35	1524	6	16	86	0.9375	0.9896104	0.9465839

3.4.2 Perbandingan lama waktu dari Google Vision dan Tesseract dalam proses mengolah gambar koran menjadi sebuah kata

Pada tahapan ini peneliti telah melakukan perhitungan waktu dalam proses mengubah gambar menjadi teks. Pada tahapan ini peneliti menghitung menggunakan fungsi microtime atau timer yang telah disediakan dalam Laravel 8. Timer dimulai ketika user mulai mengirimkan gambar kedalam sistem. Berikut adalah Tabel 9 yang menunjukkan perbandingan lama waktu dari Google Vision dan Tesseract dalam proses mengolah gambar menjadi sebuah kata.

Tabel 8. Hasil perbandingan waktu proses OCR antara Google Vision dengan Tesseract

No. Koran	Google Vision	Tesseract
1.	26,54 detik	15,52 detik
2.	27,96 detik	7,34 detik
3.	37,92 detik	12,39 detik
4.	26,6 detik	12,29 detik
5.	29,85 detik	14,31 detik
6.	25,83 detik	6,65 detik
7.	28,14 detik	12,13 detik
8.	34,44 detik	21,23 detik
9.	28,88 detik	12,42 detik
10.	31,37 detik	17,51 detik
11.	169,88 detik	10,81 detik
12.	164,73 detik	14,46 detik
13.	137,98 detik	15,32 detik
14.	196,4 detik	14,28 detik
15.	161,02 detik	17,74 detik
16.	139,63 detik	11,78 detik
17.	115,52 detik	14,7 detik
18.	137,07 detik	12,21 detik
19.	126,69 detik	15,06 detik
20.	207,12 detik	10,9 detik
21.	34,82 detik	15,26 detik
22.	31,57 detik	17,28 detik
23.	30,69 detik	11,34 detik
24.	33,17 detik	13,79 detik
25.	30,13 detik	11,3 detik
26.	30,3 detik	10,08 detik
27.	28,18 detik	13,79 detik
28.	29,32 detik	7,41 detik
29.	30,07 detik	11,59 detik
30.	32,54 detik	9,33 detik

No. Koran	Google Vision	Tesseract
31.	28,19 detik	12,79 detik
32.	28,39 detik	7,14 detik
33.	27,72 detik	14,48 detik
34.	30,38 detik	15,63 detik
35.	29,01 detik	13,42 detik

Berdasarkan pengamatan penulis serta hasil perbandingan seperti diatas maka dapat disimpulkan bahwa algoritma Tesseract lebih cepat dalam melakukan proses *OCR*. Google Vision membutuhkan waktu yang lebih lama dan kadang hasil waktunya berbeda karena dipengaruhi oleh koneksi internet seperti pada contoh nomor koran 1-10 dan 21-35 yang menggunakan koneksi internet yang stabil dan lancar sehingga hanya memerlukan waktu rata-rata 30,08 detik, sedangkan nomor koran 11-20 menggunakan koneksi internet yang kurang stabil sehingga memerlukan rata-rata waktu 155,60 detik.

3.4.3 Hasil perbandingan dari Google Vision dan Tesseract dalam proses mengolah gambar koran menjadi sebuah kata

Berdasarkan pengamatan dan penelitian yang dilakukan oleh peneliti maka dapat dibandingkan antara kedua algoritma tersebut. Tabel 10 menunjukkan hasil perbandingan dari Google Vision dan Tesseract dalam mengolah gambar koran menjadi sebuah kata.

Tabel 9. Hasil perbandingan dari Google Vision dan Tesseract

Aspek	Google Vision	Tesseract
Waktu	Memerlukan waktu rata-rata 30.08 detik (dalam kondisi internet stabil)	Memerlukan waktu rata-rata 12.96 detik
Akurasi	Berdasarkan 35 halaman koran diatas Google Vision memiliki akurasi rata-rata 97,89%. Yang berarti algoritma ini mampu menggambarkan tingkat ketepatan algoritma dalam mengolah sebuah data sebesar rata-rata 97,89 %.	Berdasarkan 35 halaman koran diatas Tesseract memiliki akurasi rata-rata 90.16%. Yang berarti algoritma ini mampu menggambarkan tingkat ketepatan algoritma dalam mengolah sebuah data sebesar rata-rata 90.16%.
Presisi	Berdasarkan 35 halaman koran diatas Google Vision memiliki presisi rata-rata 98,62%. Yang berarti algortima ini mampu menggambarkan tingkat ketepatan prediksi antara data keadaan yang asli (data yang diminta) dengan hasil prediksi yang diberikan oleh sistem sebesar rata-rata 98,62%.	Berdasarkan 35 halaman koran diatas Tesseract memiliki presisi rata-rata 96,62%. Yang berarti algortima ini mampu menggambarkan tingkat ketepatan prediksi antara data keadaan yang asli (data yang diminta) dengan hasil prediksi yang diberikan oleh sistem sebesar rata-rata 96,62%.
Sensitivitas	Berdasarkan 35 halaman koran diatas Google Vision memiliki sensitivitas rata-rata 99,23%. Yang berarti algoritma ini mampu menggambarkan tingkat kepekaan algoritma dalam mendeteksi data yang benar dengan data yang tidak sesuai sebesar rata-rata 99,23%.	Berdasarkan 35 halaman koran diatas Tesseract memiliki rata-rata 92,82%. Yang berarti algoritma ini mampu menggambarkan tingkat kepekaan algoritma dalam mendeteksi data yang benar dengan data yang tidak sesuai sebesar rata-rata 92,82%.

Aspek	Google Vision	Tesseract
Jenis perangkat lunak	Bersifat <i>Closed Source</i> .	Bersifat <i>Open Source</i> .
Tarif perangkat lunak	- 1-1.000 unit/bulan (gratis) - 1.001-5.000.000 unit/bulan (\$1.50/1.000 unit) - 5.000.001-∞ unit/bulan (\$0.60/1.000 unit)	Gratis

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan diatas, maka dapat disimpulkan bahwa *Google Vision* memiliki tingkat akurasi, presisi, dan sensitivitas lebih baik dalam membaca teks koran dibandingkan dengan *Tesseract*. Dalam hal waktu *Tesseract* lebih unggul karena *Tesseract* bersifat *offline* sedangkan *Google Vision* bersifat *online* dan sangat tergantung pada kecepatan internet yang terhubung dalam jaringan tersebut. Selain itu, *Google Vision* berifat *Closed Source* dan dikenakan biaya bulanan apabila melakukan proses *OCR* lebih dari 1000 gambar per bulan sedangkan *Tesseract* bersifat *Open Source* dan sama sekali tidak dikenakan biaya (gratis). Maka dapat disimpulkan bahwa secara umum *Google Vision* lebih unggul jika dibandingkan dengan *Tesseract* karena tingkatan dari akurasi, sensitivitas, serta presisi lebih unggul, serta lama waktu untuk memproses *OCR* tidak terjadi selisih yang cukup lama jika koneksi stabil. Tarif dari *OCR Google Vision* juga tergolong murah bahkan gratis apabila melakukan kurang dari 1000 kali perbulan.

5. SARAN

Berdasarkan penelitian yang telah dilaksanakan, beberapa saran yang dapat diberikan yaitu penelitian dapat dikembangkan dengan memperluas variasi jenis dokumen yang akan discan/deteksi seperti koran lama, surat, majalah, dan lain-lain. Serta teknologi *OCR* dapat di implementasikan dalam pendidikan misalnya diterapkan dalam sistem informasi pencarian dipergustakaan dengan cara melakukan *scanning cover-cover* buku yang ada dipergustakaan sehingga memudahkan petugas perpustakaan untuk menginputkan buku dalam sistem informasi dan memudahkan pengunjung perpustakaan untuk mencari buku

DAFTAR PUSTAKA

- [1] S. E. Umbaugh, *Digital Image Processing and Analysis: Applications with MATLAB® and CVIPtools*. CRC press, 2017.
- [2] M. Saifudin and H. Widrani, "Rancang Bangun Sistem Digitalisasi Dokumen Menggunakan Metode Visible Watermark Di Kantor Urusan Agama (Kua) Kecamatan Sayung," *J. Tek. Inform. dan Teknol. Inf.*, vol. 1, no. 3, pp. 1–7, 2021.
- [3] R. C. Gonzalez and R. E. Woods, "Digital Image Processing, Hoboken." NJ: Pearson, 2018.
- [4] V. Kajla, A. Gupta, and A. Khatak, "Analysis of x-ray images with image processing techniques: A review," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, 2018, pp. 1–4.
- [5] N. Prameela, P. Anjusha, and R. Karthik, "Off-line Telugu handwritten characters recognition using optical character recognition," in *2017 International conference of*

- electronics, communication and aerospace technology (ICECA)*, IEEE, 2017, pp. 223–226.
- [6] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, “Optical character recognition systems,” in *Optical Character Recognition Systems for Different Languages with Soft Computing*, Springer, 2017, pp. 9–41.
- [7] R. Kaur and D. V. Sharma, “Punjabi text recognition system for portable devices: A comparative performance analysis of cloud vision API with Tesseract,” *J. Comput. Sci. Eng.*, vol. 2, no. 2, pp. 104–111, 2021.
- [8] I. J. Manuel, “Analisis Perbandingan Optical Character Recognition Google Vision Dengan Tesseract Pada Pengenalan Plat Nomor Kendaraan Pribadi Di Indonesia.” Universitas Multimedia Nusantara, 2018.
- [9] S. Dhiman and A. Singh, “Tesseract vs gocr a comparative study,” *Int. J. Recent Technol. Eng.*, vol. 2, no. 4, p. 80, 2013.
- [10] A. Athira, S. Lekshmi, P. Vijayan, and B. Kurian, “Smart parking system based on optical character recognition,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2019, pp. 1184–1188.
- [11] R. Rismanto, A. Prasetyo, and D. A. Irawati, “Optimalisasi Image Thresholding Pada Optical Character Recognition Pada Sistem Digitalisasi dan Pencarian Dokumen,” *PETIR*, vol. 13, no. 1, pp. 1–11, 2020.
- [12] M. R. Firdaus, “Penerapan OCR (Optical Character Recognition) Pada Sistem Akuisisi Dokumen Jabatan Fungsional Dosen.” Universitas Muhammadiyah Malang, 2020.
- [13] S. Muharom, “Pengenalan Nomor Ruangan Menggunakan Kamera Berbasis OCR Dan Template Matching,” *J. Inf.*, vol. 4, no. 1, 2019.
- [14] R. D. Madhani, “Optical Character Recognition (Ocr) Pada Dokumen Karya Tulis Ilmiah Menggunakan Metode Long Short-Term Memory (Lstm).” Universitas Komputer Indonesia, 2021.
- [15] I. Andi, M. Muchtar, and J. Y. Sari, “Mask Detection Using the YOLO (You Only Look Once) Method,” *Media Inf. Teknol.*, vol. 1, no. 1, pp. 1–12, 2024.
- [16] D. Febiharsa, I. M. Sudana, and N. Hudallah, “Uji fungsionalitas (blackbox testing) sistem informasi lembaga sertifikasi profesi (silsp) batik dengan appperfect web test dan uji pengguna,” *Joined J. (Journal Informatics Educ.*, vol. 1, no. 2, pp. 117–126, 2018.
- [17] P. N. Andono and T. Sutojo, *Pengolahan citra digital*. Penerbit Andi, 2017.
- [18] P. Subarkah, E. P. Pambudi, and S. O. N. Hidayah, “Perbandingan Metode Klasifikasi Data Mining untuk Nasabah Bank Telemarketing,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 1, pp. 139–148, 2020.