

Pencocokan Kata dalam *Optical Character Recognition* Menggunakan Metode *Hamming Distance*

Rakhmadhan Rizky Brilliant¹⁾, Soffiana Agustin²⁾

^{1,2)}Teknik Informatika, Universitas Muhammadiyah Gresik, Jl. Sumatra 101 Gresik Kota Baru (GKB),
Randuagung, 661121, Indonesia

e-mail: rakhmadhanrizky112@gmail.com¹⁾, soffiana@umq.ac.id²⁾

ABSTRAK

Citra adalah representasi objek dua dimensi dari dunia visual, menyangkut berbagai macam disiplin ilmu yang mencakup seni, human vision, astronomi, teknik, dan sebagainya. Pada Penelitian ini, dilakukan penerapan *Hamming Distance* pada *Optical Character Recognition* menggunakan *Matlab* dan *Python*. Kata yang digunakan dalam penelitian ini merupakan hasil tulis tangan dalam bahasa Indonesia, dimana Penulis dibagi dalam tiga kelompok dengan usia 10-15 tahun (kelompok A), 20-25 tahun (kelompok B) dan usia lebih dari 50 tahun (kelompok C). Penelitian dimulai dengan analisis citra dengan melakukan pra-pengolahan dan Ekstraksi kata menggunakan *Optical Character Recognition (OCR)*. Hasil OCR ini kemudian dimasukkan ke pengenalan *Hamming* untuk dilakukan pengenalan Kata. Hasil penelitian menunjukkan kinerja OCR dalam mengenali kata memberikan akurasi tertinggi pada kelompok B yaitu 40%, sedangkan pengenalan kata menggunakan *Hamming distance* memberikan akurasi 85% pada kelas yang sama. Baik OCR maupun *Hamming* belum mampu mengenali kata dari tulisan tangan dengan baik pada kelompok A dan C.

Kata Kunci: optimasi, pencocokan_kata, hamming_distance, optical_character_recognition

ABSTRACT

Image is a two-dimensional representation of objects from the visual world, involving various disciplines including art, human vision, astronomy, engineering, and so on. In this study, *Hamming Distance* was applied to *Optical Character Recognition* using *Matlab* and *Python*. The words used in this study were handwritten in Indonesian, which were divided into three groups with ages 10-15 years (group A), 20-25 years (group B) and over 50 years (group C). The study began with image analysis by pre-processing and word extraction using *Optical Character Recognition (OCR)*. The results of this OCR were then entered into *Hamming* recognition to perform word recognition. The results showed that OCR performance in recognizing words provided the highest accuracy in group B, which was 40%, while word recognition using *Hamming distance* provided 85% accuracy in the same class. Both OCR and *Hamming* were not able to recognize words from handwriting well in groups A and C.

Keywords: optimization, word matching, hamming distance, optical character recognition, tesseract engine

1. PENDAHULUAN

Tulisan Tangan setiap orang tidak selalu sama dan terorganisir sehingga banyak terjadi masalah dalam pembacaan, hal tersebut menyebabkan masalah dalam kehidupan sehari-hari hingga dunia kerja. Mulai dari tulisan anak kecil yang kadang tidak mudah untuk dibaca oleh orang tua bahkan Ketika disekolah mungkin tidak terbaca oleh sang guru. Hal tersebut menghasilkan masalah dalam interpretasi kata karena jika berbeda dapat menyebabkan salah paham dan sebagainya.

Pada penelitian yang dilakukan oleh Vijaya Madane, Kajal Ovhal dan Mrunali Bhong yang berjudul *Handwriting Recognition Using Artificial Intelligence Neural Network And Image Processing*. Pada Penelitian sebelumnya yang dilakukan oleh Wahyuddin, Askar Hakim yang berjudul *Aplikasi Ekstraksi Data Kartu Vaksin Berbasis Web Menggunakan Metode Optical Character Recognition (OCR)*. Pada Penelitian yang dilakukan oleh Wu Wena, Xiaobo Xueb, Ya Lia, Peng Gua, Jianfeng Xu yang berjudul *Code Similarity Detection using AST and Textual Information*. Pada penelitian yang dilakukan oleh Asha Rani Mishra, V.K Panchal, Pawan Kumar yang berjudul *Similarity Search based on Text Embedding Model for detection of Near Duplicates* [1], [2], [3], [4]. Berdasarkan Penelitian tersebut berhasil melakukan Ekstraksi fitur dengan beberapa metode seperti Neural network dan OCR. Juga pada penelitian berhasil untuk mengecek tingkat kesamaan kata menggunakan beberapa metode seperti AST, Euclidean distance dan lainnya.

Penelitian ini mengusulkan penggunaan Hamming Distance dalam mengenali kata dari tulisan tangan hasil dari OCR. Penelitian ini akan dilakukan dalam dua tahap utama yaitu analisis tulisan tangan dan pengenalan kata menggunakan OCR dan tahap kedua pengenalan lanjutan menggunakan Hamming Distance.

2. TINJAUAN PUSTAKA

Sebagai usaha untuk menguatkan topik penelitian, penulis melakukan analisis dari hasil riset penelitian terdahulu yang berkaitan dengan topik penelitian. Berikut adalah hasil dari penelitian tersebut :

Wahyuddin, Askar Hakim (2023) dengan judul penelitian “Aplikasi Ekstraksi Data Kartu Vaksin Berbasis Web Menggunakan Metode OCR”. Penelitian tersebut membahas tentang penggunaan metode OCR sebagai metode rekognisi dari gambar yang diambil lewat kamera untuk menghasilkan aplikasi ekstraksi data kartu vaksin dan dimunculkan dalam bentuk website. Berdasarkan kesimpulan penelitian tersebut hasil ekstraksi kartu vaksin menggunakan OCR berhasil dengan catatan gambar yang diunggah memiliki pencahayaan yang baik dan jelas[1].

Syahri Muharom (2019) dengan judul penelitian “Pengenalan Nomor Ruangan Menggunakan Kamera Berbasis OCR Dan Template Matching”. Penelitian tersebut membahas tentang pengaplikasian metode OCR untuk dapat mengenali nomor ruangan dan menjadi alternatif metode baru untuk dapat mengenali nomor ruangan menggunakan kamera. Berdasarkan kesimpulan penelitian tersebut memiliki Tingkat akurasi yang cukup tinggi sebesar 93,75% [5].

Susan Siti Nurhaliza, Lussiana ETP (2022) dengan judul penelitian “Sistem Pengenalan Karakter Dokumen Secara Otomatis Menggunakan Metode Optical Character Recognition”. Penelitian tersebut membahas tentang pengaplikasian metode OCR untuk mengenali huruf pada dokumen distribusi izin alat Kesehatan. Berdasarkan kesimpulan penelitian tersebut penerapan metode OCR berhasil mengenali karakter sebanyak 312 sehingga Tingkat akurasi pengujian adalah 98.78% dengan rata-rata waktu proses untuk mengenali karakter sebesar 1.29 detik [6].

Muhammad Rizal Toha, Agung Triayudi (2022) dengan judul “Penerapan Membaca Tulisan di dalam Gambar Menggunakan Metode OCR Berbasis Website pada e-KTP”. Penelitian tersebut membahas tentang penerapan metode OCR untuk membangun sebuah sistem informasi pembaca teks dari gambar foto e-KTP dan melakukan pengujian dan analisis pada hasil sistem informasi yang dibangun. Berdasarkan kesimpulan penelitian tersebut bahwa perangkat lunak yang dirancang berhasil untuk mengekstrak semua data pada E-KTP dengan nilai akurasi yang cukup bagus [7].

Hasan Nindya Murwato, Suci Aulia, Atik Novianti (2020) dengan judul penelitian “Perancangan Translator Image to Text Dengan Menggunakan Metode Optical Character Recognition Berbasis Matlab”. Penelitian tersebut membahas tentang penerapan metode Optical Character Recognition untuk sistem translator yang dapat menerjemahkan kata dalam gambar lalu menampilkannya dengan kata yang bisa dimengerti oleh wisatawan. Berdasarkan kesimpulan penelitian tersebut bahwa metode OCR berhasil untuk melakukan translasi dengan nilai akurasi rata-rata yang cukup tinggi dengan Cahaya yang memadai karena hal tersebut sangat mempengaruhi proses deteksi karakter [8].

Yunimawar Niati Gulo (2022) dengan judul penelitian “Penerapan Algoritma Hamming Distance Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Indonesia”. Penelitian tersebut membahas tentang penerapan metode Hamming distance untuk membantu pencarian teks sehingga memudahkan pencarian teks yang sesuai dengan konteks yang dicari. Berdasarkan kesimpulan penelitian tersebut pencarian teks menggunakan metode Hamming distance bisa diterapkan sesuai dengan penelitian yang dilakukan [9].

Mayesti Anggelina, Lucia Dwi Krisnawati, Danny Sebastian (2022) dengan judul penelitian “Penerapan Simhash dan Hamming distance dalam Deteksi kemiripan Teks Berita”. Penelitian tersebut membahas tentang cara penerapan simhash dengan algoritma Local Sensitive Hashing juga metode Hamming Distance untuk membuat sistem yang bisa menampilkan dokumen yang memiliki kesamaan pada bagian nama hingga paragraf dengan Tingkat kemiripan tertentu. Berdasarkan kesimpulan penelitian tersebut Bahwa penerapan algoritma simhash dan Hamming Distance berhasil dengan Tingkat akurasi rata-rata 27% dan recall 80%[10].

Mudawil Qulub, Rifqi Hammad, Pahrul Irfan, Yuliana (2023) dengan judul penelitian “Improvement of Spelling Correction Accuracy in Indonesian Language through the Application

of Hamming Distance Method”. Penelitian tersebut membahas tentang penerapan dan Analisa metode Hamming Distance terhadap kesalahan kata baku dan tidak baku dalam Bahasa Indonesia untuk mengetahui Tingkat koreksi pada kesalahan kata bahasa Indonesia. Berdasarkan Kesimpulan penelitian tersebut pada pengujian 1 dengan kata salah atau perbedaan 1 dan 2 karakter, metode Hamming distance menghasilkan akurasi sebesar 98,33%. Untuk perbedaan 1 karakter 100% akurasi dan perbedaan 2 karakter 96.67% [11].

Aria Novitra (2023) dengan judul penelitian “Penerapan Algoritma Approximate String Matching Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Teknologi Komputer”. Penelitian tersebut membahas tentang cara penerapan metode Approximate String Matching dan Hamming distance ke sebuah sistem untuk membantu pengguna sistem mencari kesalahan dalam teks. Berdasarkan kesimpulan penelitian tersebut bahwa metode Hamming Distance dapat diterapkan di sistem tersebut [12].

Susi Rianti, Riza Adrianti Supono (2023) dengan judul penelitian “Perbandingan Algoritma Edit Distance, Levenshtein Distance, Hamming Distance, Jaccard Similarity Dalam Mendeteksi String Matching”. Penelitian tersebut membahas tentang cara membandingkan beberapa algoritma untuk menentukan mana yang terbaik. Dengan Levenshtein Distance di tempat pertama dengan nilai MAP 13,125ms dan Hamming Distance dengan nilai MAP 14,25ms [13].

Berdasarkan penelitian yang telah dilakukan sebelumnya, memperkuat dimungkinkannya menemukan atau memberikan alternatif kata yang paling sesuai dengan citra tulisan tangan (tulisan dalam gambar).

3. METODOLOGI

3.1. Dataset

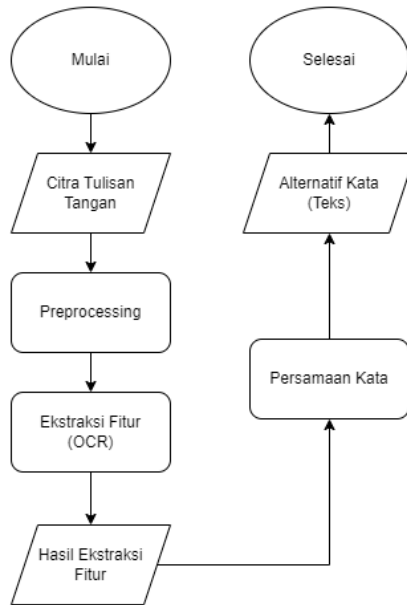
Untuk penelitian ini menggunakan data yang didapat dari tulisan tangan orang, data yang digunakan dibagi menjadi 3 kelas. Untuk kelas A menggunakan data tulisan tangan dari orang yang berumur 10-15 tahun, untuk kelas B menggunakan data tulisan tangan dari orang yang berumur 20-25 tahun dan Kelas C menggunakan data tulisan tangan dari orang yang berumur 50+ tahun. Tiap kelas berisi 20 data gambar tulisan tangan dengan total 60 gambar.

Tabel 1. Dataset Tulisan Tangan

Data Citra	Tipe Data	Jumlah
Citra Tulisan Tangan	Tulisan Tangan Usia 10 - 15	20
	Tulisan Tangan Usia 20 - 25	20
	Tulisan Tangan Usia 50+	20
	Jumlah	60

3.2. Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran secara umum tentang sistem yang akan dibuat sehingga kebutuhan sistem dapat diketahui sebelumnya. Fungsi flowchart ialah memberi gambaran tentang program yang akan dibuat pada penelitian ini. Dalam proses rekognisi dan pelengkap kata tulisan tangan terdapat beberapa proses yang perlu dilalui. Gambar 2.1 memperlihatkan cara alur kerja sistem atau Flowchart sistem secara visual dari awal hingga akhir.

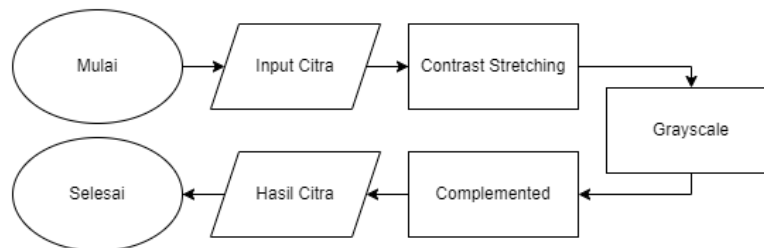


Gambar 1 Flowchart Pencocokan Tulisan Tangan

Gambar 1 menjelaskan proses sistem pengecekan tulisan tangan yang dilakukan dengan beberapa tahapan, berawal dari memasukkan data awal citra untuk setelahnya dilakukan preprocessing atau pemrosesan awal menggunakan contrast stretching, grayscale dan complement. Dilanjut dengan proses pengenalan citra menggunakan Optical Character Recognition (OCR). Hasil OCR berupa teks (karakter) yang digunakan sebagai data masukan pada google colab. Teks masukan pada google colab tersebut kemudian dilakukan pembersihan hasil dari karakter yang tidak diinginkan seperti karakter selain huruf. Setelah dibersihkan, dilakukan proses pencarian kata untuk menemukan kata yang kurang cocok hingga cocok menggunakan metode Hamming Distance. Setelah dilakukan pengecekan akan didapat sebuah hasil yang menunjukkan kata mana yang lebih cocok dari tulisan yang dicari dengan tingkat akurasi tiap kata yang ditemukan dengan kata awal.

3.3. Preprocessing

Preprocessing adalah proses di mana citra di rubah untuk menyesuaikan format citra yang digunakan menjadi lebih sederhana untuk dilakukan analisis lebih lanjut. Dalam penelitian ini untuk Bagian Preprocessing terdiri dari beberapa metode untuk urutannya akan dijelaskan dengan Flowchart di bawah ini.



Gambar 2 Flowchart Preprocessing

Gambar 2 menjelaskan Langkah-langkah pra-pemrosesan untuk penelitian ini. Penjelasan mengenai tahapan pengerjaannya pra-pemrosesan adalah sebagai berikut.

Pertama, dilakukan perbaikan kualitas masukan citra berupa peregangan kontras atau contrast stretching. Fungsi tersebut mampu meningkatkan kontras gambar sehingga menjadi lebih jelas. memberikan gambaran hasil dari imadjust. imadjust akan mengubah nilai intensitas citra masukan menjadi lebih cerah karena intensitas citra masukan awal akan di tingkatkan pada rentang nilai intensitas keluaran.

Peninggian kontras memiliki intensitas tiga warna atau RGB yaitu merah (red), hijau

(green) dan biru (blue). Untuk itu dibutuhkan 2x3 matriks yaitu matriks RGB untuk bagian low-in dan juga high-in. dalam penelitian ini menggunakan 0.9 untuk low-in disama ratakan untuk semua warna RGB dan 1 untuk high-in disama ratakan untuk semua warna RGB dengan itu menghasilkan warna yang cocok untuk menaikkan intensitas warna objek hitam.

Setelah dilakukan proses peregangan kontras dihasilkan citra Grayscale. Grayscale adalah proses di mana citra akan diubah mode warnanya dari misal RGB ke warna ke abuan. Di bawah ini akan diberikan contoh gambar dengan mode warna RGB.

Untuk mengubah menjadi Grayscale dilakukan perhitungan dengan persamaan sederhana dengan contoh nilai rata-rata kecerahan dalam tiga warna RGB yang berbeda menggunakan rumus persamaan:

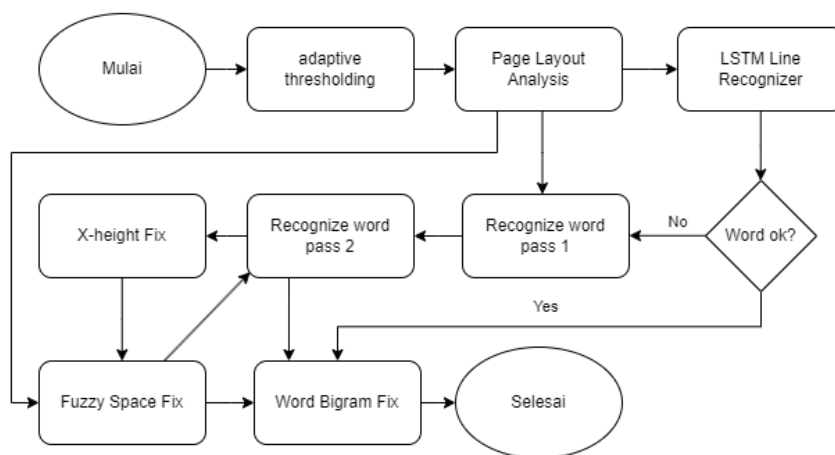
3.4. Optical Character Recognition

Optical Character Recognition, pengenalan karakter atau yang bisa disebut OCR adalah salah satu bidang dalam computer vision dan hubungan komputer manusia. OCR bekerja dengan cara memisahkan kalimat menjadi kata dan simbol huruf sebelum dilakukan pengenalan. Karena kualitas pengenalan sangat bergantung terhadap kualitas gambar dokumen tersebut maka dilakukan beberapa pre-processing untuk membuat algoritma dapat menangkap deteksi lebih baik seperti Pre-processing [14].

Ekstraksi fitur bertujuan untuk mengidentifikasi dan mengekstraksi atribut-atribut dari karakter yang akan digunakan untuk pengenalan. Terdapat berbagai metode yang dapat digunakan untuk mengekstraksi fitur dari citra teks, termasuk ekstraksi bentuk, tekstur, dan statistik dari karakter yang terdeteksi.

Salah satu metode ekstraksi fitur yang umum digunakan dalam OCR adalah ekstraksi bentuk dan ukuran karakter. Pada tahap ini, karakter yang telah diidentifikasi diproses untuk mengekstraksi atribut-atribut geometris seperti luas, tinggi, lebar, dan bentuk karakter. Informasi ini dapat membantu dalam membedakan karakter yang berbeda serta memperbaiki keandalan pengenalan karakter.

Untuk penggunaan OCR, matlab menggunakan Tesseract engine. Untuk penjelasan cara kerja engine tersebut akan dijelaskan dalam bentuk visual menggunakan flowchart.

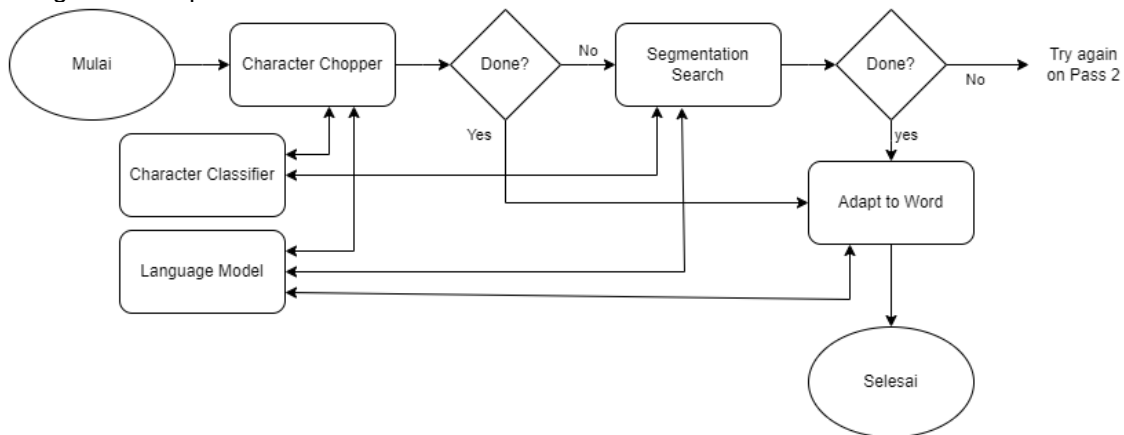


Gambar 5 Flowchart Tesseract Engine

Gambar 2.7 menjelaskan tentang cara kerja Tesseract Engine atau Tesseract System Architecture. Untuk cara kerja Engine tersebut secara pipeline dimulai dari adaptive thresholding.

Adaptive thresholding dalam Tesseract Engine adalah teknik yang digunakan untuk mengubah citra menjadi citra biner sebelum proses pengenalan teks (OCR). Metode ini memungkinkan Tesseract untuk bekerja lebih baik dalam kondisi cahaya yang bervariasi atau ketika latar belakang citra tidak homogen. Dengan adaptive thresholding, citra dibagi menjadi beberapa wilayah kecil, dan Tesseract menghitung nilai ambang untuk setiap wilayah berdasarkan statistik lokal seperti rata-rata intensitas piksel di sekitarnya. Setiap wilayah kemudian diubah menjadi citra biner menggunakan ambang yang telah dihitung, memungkinkan Tesseract untuk lebih akurat memisahkan teks dari latar belakang.

Page layout analysis dalam Tesseract Engine adalah proses krusial di mana sistem mengenali dan memahami struktur halaman secara menyeluruh sebelum melakukan OCR (Optical Character Recognition). Ini melibatkan identifikasi dan segmentasi elemen-elemen seperti teks, gambar, dan tabel dalam sebuah dokumen. Tesseract mengidentifikasi area-area di mana teks terletak, termasuk baris teks, paragraf, dan blok teks, serta gambar dan tabel yang ada. Segmentasi halaman juga dilakukan dengan membagi halaman menjadi bagian-bagian yang lebih kecil, seperti kolom atau sel, terutama saat mengenali tabel. Proses ini juga memperhitungkan margin dan padding di sekitar teks dan elemen lainnya untuk memahami batas-batas antara elemen-elemen tersebut. Setelah page layout analysis dilanjutkan dengan recognize word pass 1 dan 2.



Gambar 6 Flowchart Recognize Word pass 1 dan 2

x-height fix merupakan proses yang bertujuan untuk mengatasi masalah pengenalan teks yang terkait dengan tinggi huruf (x-height) yang bervariasi. X-height adalah tinggi huruf bagian tengah, seperti huruf "x" atau "o", dan perbedaan dalam x-height dapat memengaruhi pengenalan teks. Proses x-height fix bekerja dengan cara menyesuaikan tinggi huruf dalam proses pengenalan teks, terutama untuk huruf-huruf yang memiliki x-height yang tidak standar. Dengan memperbaiki tinggi huruf ini, Tesseract dapat meningkatkan akurasi pengenalan teks, terutama pada dokumen yang memiliki variasi tinggi huruf yang signifikan.

Setelah proses x-height fix dilanjut oleh proses Fuzzy space fix. proses tersebut digunakan untuk mengatasi masalah spasi yang tidak jelas atau tidak konsisten antara kata-kata dalam teks yang diakui. Ketika teks diakui, terkadang terjadi kesalahan dalam penentuan spasi antara kata-kata, terutama saat ada teks yang bergabung atau tumpang tindih. Proses ini menggunakan pendekatan "fuzzy", yang berarti mendekati atau kurang pasti, untuk menentukan spasi yang seharusnya antara kata-kata. Tesseract Engine akan memeriksa konteks teks dan karakter di sekitar spasi yang ambigu, lalu membuat perkiraan atau estimasi untuk menambahkan atau mengurangi spasi yang diperlukan. Dengan melakukan ini, fuzzy space fix membantu meningkatkan konsistensi dan kejelasan teks yang dihasilkan setelah proses OCR.

Proses yang terakhir sebelum selesai adalah word bigram fix. proses tersebut digunakan untuk memperbaiki kesalahan pengenalan teks yang terjadi pada kata-kata yang berdekatan, atau yang disebut dengan bigram. Bigram adalah dua kata yang berurutan dalam teks, dan word bigram fix bertujuan untuk mengoreksi kesalahan pengenalan pada pasangan kata-kata ini. Dalam beberapa kasus, Tesseract dapat salah mengenali bigram karena kemungkinan variasi karakter atau kata yang mirip.





Proses ini bekerja dengan memeriksa konteks dan makna dari bigram yang terdeteksi, kemudian mencocokkan dengan kata-kata yang mungkin benar berdasarkan dictionary atau model bahasa. Jika terdapat bigram yang tidak sesuai atau tidak cocok dengan kata-kata yang umumnya muncul berdampingan, word bigram fix akan mencoba untuk mengoreksi dan menyesuaikan pasangan kata tersebut. Hal ini membantu meningkatkan akurasi pengenalan teks, terutama dalam kasus di mana ada variasi dalam urutan kata atau kesalahan pengenalan pada pasangan kata yang sering muncul bersama.

Untuk LSTM line recognizer baru ditambahkan pada beberapa tahun ke belakang, LSTM digunakan karena mempunyai hasil yang sama dengan HMM dan DNN serta lebih baik dari metode lainnya. Untuk integrasi LSTM dengan tesseract mulai dari kode LSTM didasari oleh implementasi OCROpus python, memperluas kemampuan termasuk 2-d dan ukuran input

bervariasi, sepenuhnya berintegrasi dengan tesseract pada Tingkat kelompok kata yang mirip, visualisasi dengan API viewer yang ada dan lainnya. Hasil integrasi tersebut membuat sebuah jaringan tesseract dengan string yang ringkas, kemampuan terbatas tapi sangat fleksibel dalam Batasan tertentu dan mudah digunakan serta sedikit yang perlu di pelajari.

Untuk output yang dihasilkan adalah dalam bentuk tulisan digital yang didapat dari gambar tulisan yang dimasukkan. Contoh hasil dari ekstraksi fitur menggunakan data dummy berada di bawah ini.

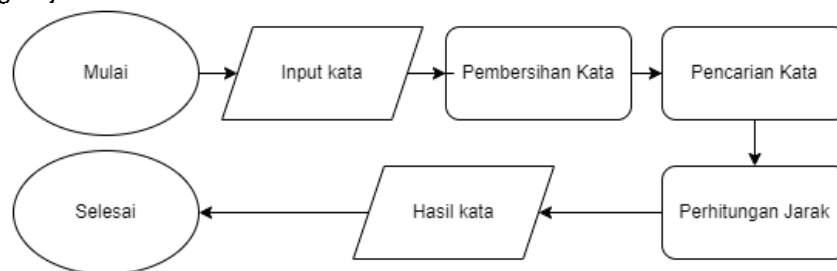
Tabel 2 Hasil ekstraksi Fitur (OCR)

No	Gambar awal	Gambar akhir	Hasil teks
1	ALMOND		ALMOND
2	APPLE		APFKE
3	banana		9fifAna
4	BANANA		BAN/Awx

3.5. Persamaan kata

Pada hasil yang telah didapat setelah melakukan proses Ekstraksi Fitur (OCR) menggunakan Tesseract, dengan menggunakan matlab, akan dihasilkan hasil deteksi citra berupa teks. teks tersebut akan dijadikan masukan untuk mendapatkan Alternatif kata yang lebih tepat. kata masukan tersebut akan diproses melalui beberapa tahapan terlebih dahulu sebelum masuk ke proses persamaan kata menggunakan metode Hamming distance.

Hamming distance adalah salah satu algoritma mengukur kedekatan item. Jika nilai jarak makin kecil, maka kedua item itu semakin dekat dan berlaku sebaliknya. Yang biasanya dibandingkan adalah kata dan bilangan biner [15]. Flowchart di bawah ini akan menjelaskan tahapan yang terjadi.



Gambar 7 Flowchart Persamaan Kata

Tahap awal dimulai dengan membersihkan input kata yang didapat dari ekstraksi fitur. Input kata dibersihkan dari simbol hingga angka yang terbawa lewat ekstraksi dan diganti dengan spasi untuk mempermudah perhitungan jarak.

Setelah proses pembersihan kata selesai, dilanjut dengan proses pencarian kata. Input kata dimasukkan dalam algoritma untuk mencari kata yang sesuai, untuk penelitian ini input kata yang dipakai adalah buah karena tidak berubah-ubah. ada 3 tahapan dalam proses ini yang

pertama dengan cara mencocokkan kata awal dengan kata yang ada di basis data, yang kedua dengan cara mencocokkan kata awal yang terpotong secara utuh dan tidak terpisah dengan basis data dan yang ketiga dengan cara mencocokkan kata yang dipisah per huruf dan sesuai posisi huruf dengan database.

Setelah melakukan pencarian kata dilanjut dengan perhitungan jarak menggunakan metode hamming distance. Cara kerja metode hamming distance yaitu memecah kata awal dan kata yang ditemukan sesuai menjadi huruf satu persatu untuk dibandingkan katanya per huruf dan sesuai posisi. Jika kata yang ditemukan memiliki beberapa perbedaan huruf dari kata awal maka total huruf dari kata awal akan dikurangi dengan total huruf dari kata yang sesuai dengan syarat posisi huruf sama dan diberi nilai persentase untuk membantu penilaian.

3.6. Akurasi

Langkah selanjutnya adalah menghitung akurasi prediksi kata yang ditemukan dengan persamaan akurasi di bawah ini:

$$Akurasi = \frac{Kata\ benar}{Total\ Kata} \times 100\% \quad (1)$$

4. HASIL DAN PEMBAHASAN

Bagian ini mengulas tentang hasil dari penelitian yang didapat dan dilakukan juga pembahasan yang akan dibagi menjadi beberapa bagian yaitu implementasi dan pengujian. Dalam beberapa bagian tersebut juga akan dijelaskan bagaimana sistem pada penelitian ini bekerja.

4.1. Implementasi

Contrast Stretching

Kode Program 1. Contrast Stretching pada Input Gambar

```
1 % meningkatkan kontras citra
2 constr = imadjust(i3,[.8 .8;.9 .9]);
```

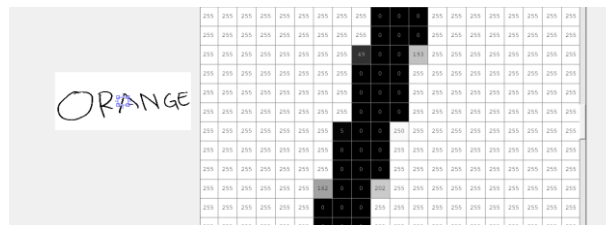


Gambar 8 Citra awal (a) dan Hasil Contrast Stretching (b)

Grayscale

Kode Program 2. Grayscale pada Hasil Contrast Stretching

```
1 % meningkatkan kontras citra
2 gray = rgb2gray(constr);
```



Gambar 9 citra hasil grayscale

Complemented

Kode Program 1. Complement pada Hasil Grayscale

```
1 % meningkatkan kontras citra
2 Imcom = imcomplement(gray);
```




Gambar 10 Hasil grayscale (a) dan complemented (b)

Ekstraksi Fitur OCR

Kode Program 1. Complement pada Hasil Grayscale

```

1 % mengubah citra menjadi teks
2 ocrResults = ocr(enc); % ekstraksi karakter dan bentuk dari citra
3 recognizedText = ocrResults.Text; % mengubah hasil ekstraksi menjadi teks
4 app.HasilTeksEditField.Value = num2str(recognizedText); % output hasil teks

```



Gambar 11 Hasil OCR

Persamaan Kata Hamming Distance

Kode Program 1. Complement pada Hasil Grayscale

```

1 # Menghitung Nilai Hamming Distance
2 hamming = panjang_awal - huruf_sama

```

4.2. Pengujian

Tujuan pengujian sistem ini adalah untuk membuktikan apakah sistem yang sudah diimplementasikan ini telah memenuhi rancangan yang telah dibuat sebelumnya. Pengujian sistem ditujukan untuk mengetahui hasil kinerja metode yang digunakan dalam mengenali kata. Hasil kinerja sistem ditampilkan pada tabel 3 untuk hasil OCR dan tabel 4 untuk hasil hamming distance.

Tabel 3 Hasil Implementasi OCR

No.	Gambar Awal	Gambar Akhir	Hasil Teks	Ground Truth	Sesuai
1.	Apple		EVVFI 9	Apple	Tidak
2.	Apple		-	Apple	Tidak
3.	AVOCADO		;;,L/z- pace Jo	Avocado	Tidak

60.	strawbery		-	Strawberry	Tidak

Untuk Kelas A:

$$Akurasi = \frac{0}{20} \times 100\% = 0\%$$

Untuk Kelas B:

$$Akurasi = \frac{8}{20} \times 100\% = 40\%$$

Untuk Kelas C:

$$Akurasi = \frac{6}{20} \times 100\% = 30\%$$

Hasil OCR tidak dapat mengenali kata yang ditulis oleh anak usia 10-15 tahun. Akurasi pengenalan OCR terhadap tulisan tangan usia 20-25 tahun dapat dikenali lebih baik yaitu sebesar 40% dan untuk kata yang ditulis oleh kelompok berusia lebih dari 50 tahun hanya mampu dikenali 30% akurat.

Tabel 4 Hasil Implementasi Hamming Distance pada OCR

No.	Gambar Awal	Hasil Akhir	Ground Truth	Sesuai
1.	APPLE	Apple,Guomi,Uvaia	Apple	Iya
2.	APPLE	-	Apple	Tidak
3.	AVOCADO	Apricot,, Avocado	Avocado	Iya

60.	strawbery	-	Strawberry	Tidak

Untuk kelas A:

$$Akurasi = \frac{2}{20} \times 100\% = 10\%$$

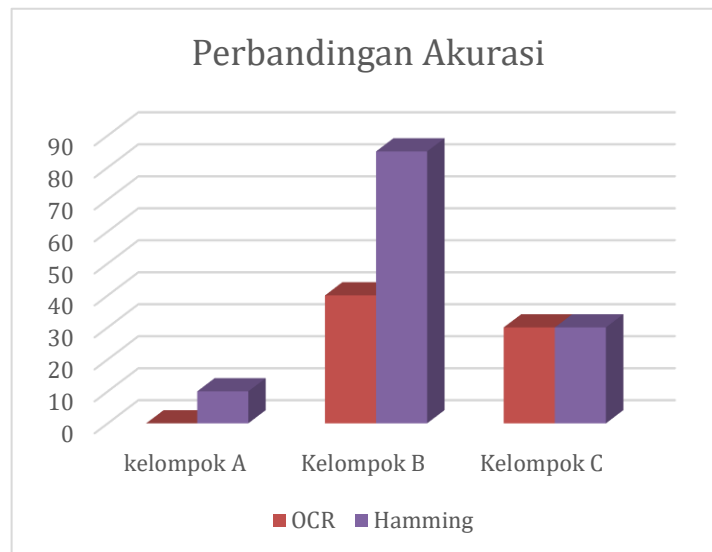
Untuk Kelas B:

$$Akurasi = \frac{17}{20} \times 100\% = 85\%$$

Untuk Kelas C:

$$Akurasi = \frac{6}{20} \times 100\% = 30\%$$

Pengenalan kata lanjutan dengan hamming distance memberikan akurasi 10% pada kelompok A dengan usia 10-15 tahun dan pengenalan kata tulisan tangan pada kelompok B memberikan akurasi 85% sedangkan pengenalan hamming terhadap kata tulisan tangan pada kelompok C dengan usia lebih dari 50 tahun memberikan akurasi pengenalan sebesar 30%.



Gambar 12. Hasil Optimasi Hamming Distance pada Kinerja OCR

5. KESIMPULAN

Hasil pengujian pada penelitian ini menunjukkan bahwa penggunaan hamming distance pada pencocokan kata memberikan hasil pengenalan lebih baik dibanding hanya menggunakan OCR. Penambahan metode Hamming Distance dapat meningkatkan akurasi hingga 75%. Tulisan tangan pada kelompok B dengan usia 20-25 tahun memberikan akurasi pengenalan paling baik dibanding kelompok usia lain dengan akurasi: 40% dari OCR dan 85% dari Hamming Distance. Rendahnya nilai akurasi dikarenakan engine yang digunakan hanya bisa mendeteksi beberapa kata saja dalam Bahasa yang ditentukan dan tulisan yang baku serta tidak miring.

Untuk pengembangan ke depannya, disarankan untuk melakukan penelitian lebih lanjut guna meningkatkan akurasi. Salah satu peluang yang masih terbuka adalah dalam meningkatkan akurasi dari OCR. Selain itu bisa juga dengan meneliti pada tahap post processing seperti menggunakan metode levenshtein, damerau-levenshtein, jaccard, dsb.

DAFTAR PUSTAKA

- [1] Wahyuddin and A. Hakim, "APLIKASI EKSTRAKSI DATA KARTU VAKSIN BERBASIS WEB," *JURNAL SINTAKS LOGIKA*, vol. 3, no. 2, 2023, [Online]. Available: <https://jurnal.umpar.ac.id/index.php/sylog>
- [2] W. Wen, X. Xue, Y. Li, P. Gu, and J. Xu, "Code Similarity Detection using AST and Textual Information," *International Journal of Performability Engineering*, vol. 15, no. 10, pp. 2683–2691, 2019, doi: 10.23940/ijpe.19.10.p14.26832691.
- [3] A. R. Mishra, V. K. Panchal, and P. Kumar, "Similarity Search based on Text Embedding Model for detection of Near Duplicates," *International Journal of Grid and Distributed Computing*, vol. 13, no. 2, pp. 1871–1881, 2020, [Online]. Available: <https://www.researchgate.net/publication/353036503>
- [4] V. Madane, K. Ovhal, and M. Bhong, "HANDWRITING RECOGNITION USING ARTIFICIAL INTELLIGENCE NEURAL NETWORK AND IMAGE PROCESSING," *International Research Journal of Modernization in Engineering Technology and Science*, Mar. 2023, doi: 10.56726/irjmet34395.
- [5] S. Muharom, "Pengenalan Nomor Ruangan Menggunakan Kamera Berbasis OCR Dan Template Matching," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 4, no. 1, Oct. 2019, doi: <https://doi.org/10.25139/inform.v4i1.1371>.
- [6] S. S. Nurhaliza and L. ETP, "Sistem Pengenalan Karakter Dokumen Secara Otomatis Menggunakan Metode Optical Character Recognition," *PETIR*, vol. 15, no. 1, pp. 166–175, Feb. 2022, doi: 10.33322/petir.v15i1.1610.

- [7] M. Rizal Toha and A. Triayudi, "Penerapan Membaca Tulisan di dalam Gambar Menggunakan Metode OCR Berbasis Website pada e-KTP," *Jurnal Sains dan Teknologi*, vol. 11, pp. 175–183, 2022, doi: 10.23887/jst-undiksha.v11i1.
- [8] H. Nindya Murwato, S. Aulia, and A. Novianti, "PERANCANGAN TRANSLATOR IMAGE TO TEXT DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION BERBASIS MATLAB IMAGE TO TEXT TRANSLATOR DESIGN USING MATLAB BASED OPTICAL CHARACTER RECOGNITION METHOD," *e-Proceeding of Applied Science*, vol. 6, no. 1, Apr. 2020.
- [9] Y. N. Gulo, "Penerapan Algoritma Hamming Distance Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Indonesia," *JoGTC: Journal Global Tecnology Computer*, vol. 1, no. 2, pp. 50–54, 2022.
- [10] M. Anggelina, L. D. K. Dwi Krisnawati, and D. Sebastian, "Penerapan Simhash dan Hamming distance dalam Deteksi kemiripan Teks Berita," *Jurnal Terapan Teknologi Informasi*, vol. 6, no. 2, pp. 131–141, Oct. 2022, doi: 10.21460/jutei.2022.62.216.
- [11] M. Qulub, R. Hammad, and P. Irfan, "Improvement of Spelling Correction Accuracy in Indonesian Language through the Application of Hamming Distance Method," 2023. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [12] A. Novitra, "Penerapan Algoritma Approximate String Matching Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Teknologi Komputer," *Journal Global Tecnology Computer*, vol. 2, no. 2, pp. 61–66, 2023.
- [13] S. Rianti and R. A. Supono, "PERBANDINGAN ALGORITMA EDIT DISTANCE, LEVENSHTAIN DISTANCE, HAMMING DISTANCE, JACCARD SIMILARITY DALAM MENDETEKSI STRING MATCHING."
- [14] A. Kumar Siliwangi and D. Prabowo, "Pencarian Informasi Berbasis Teks dalam Komik Digital Menggunakan OCR," 2022.
- [15] L. P. Sari, R. Saptono, and E. Suryani, "Computation of Scientific References Using Vector Space Model over Cosine Similarity and Hamming Distance (Case Study: Department of Informatics UNS)," *Jurnal Ilmiah Teknologi dan Informasi*, vol. 8, no. 1, 2019.