

IMPLEMENTASI *SPELLING CORRECTION* PADA OTOMATISASI REGISTRASI SURAT MASUK DENGAN METODE PETER NORVIG DAN DAMERAU-LEVENSHTEIN *DISTANCE*

Alamanda Septiara, Za'imatun Niswati, Fitrianti

^{1,3}Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Indraprasta

²Program Studi Sistem Informasi, Fakultas Teknik dan Ilmu Komputer, Universitas Indraprasta

Jl. Raya Tengah No. 80, Kel. Gedong Kec. Pasar Rebo, Jakarta Timur

alamandasptara@gmail.com, zaimatunnis@gmail.com, fitriantifl1@gmail.com

Abstrak

Proses registrasi surat masuk masih banyak dilakukan secara manual sehingga informasi pada surat perlu diinput kembali ke dalam sistem. *Teknologi Optical Character Recognition (OCR)* dapat digunakan untuk mengotomatisasi proses tersebut, namun hasil ekstraksi teks masih rentan terhadap kesalahan karakter akibat *noise* pada dokumen, seperti logo, cap, dan tanda tangan. Penelitian ini menerapkan metode *spelling correction* berbasis algoritma Peter Norvig, Damerau-Levenshtein *Distance*, dan Sorensen-Dice *Coefficient* sebagai *postprocessing* untuk meningkatkan akurasi hasil OCR pada registrasi surat masuk. Sistem meliputi deteksi area teks (YOLO), ekstraksi teks (Tesseract OCR), dan koreksi ejaan menggunakan empat kategori kamus (bulan, nama instansi, surat, dan KBBI) yang didukung *confusion mapping* untuk membatasi kandidat kata sesuai pola kesalahan OCR. Pengujian dilakukan terhadap 50 sampel surat, mengukur CER dan WER sebelum dan sesudah koreksi. Hasil menunjukkan 26 dari 50 surat mengalami peningkatan akurasi, terutama pada surat berkualitas rendah, dengan WER turun dari 20,78%-31,98% menjadi 7,36%-13,37%. Sebaliknya, 9 surat mengalami penurunan akurasi akibat *overcorrection*. Metode ini paling efektif menangani kesalahan *substitution* (turun dari 347 menjadi 180 kata), namun kurang efektif pada kesalahan *deletion* (meningkat dari 64 menjadi 80). Efektivitas metode bergantung pada kualitas awal OCR dan jenis kesalahan, sehingga lebih sesuai untuk dokumen dengan tingkat kesalahan OCR tinggi.

Kata Kunci: Damerau-Levenshtein *Distance*, *Optical Character Recognition*, Peter Norvig, *Spelling Correction*, *Character Error Rate*, *Word Error Rate*.

Abstract

The process of registering incoming mail is still largely conducted manually, requiring officers to re-input letter information into the system. *Optical Character Recognition (OCR)* can automate this process; however, text extraction remains prone to character errors caused by document noise, such as logos, stamps, and signatures. This study implements spelling correction based on the Peter Norvig algorithm, Damerau-Levenshtein *Distance*, and Sorensen-Dice *Coefficient* as post-processing to improve OCR accuracy in mail registration. The system includes text area detection (YOLO), text extraction (Tesseract OCR), and spelling correction using four dictionary categories (months, institution names, letter-related terms, and KBBI) supported by *confusion mapping* to constrain candidate words per OCR error patterns. Testing was conducted on 50 letter samples, measuring CER and WER before and after correction. Results show 26 of 50 letters exhibited improved accuracy, particularly low-quality letters, with WER decreasing from 20.78%-31.98% to 7.36%-13.37%. Conversely, 9 letters showed decreased accuracy due to *overcorrection*. The method was most effective for substitution errors (reduced from 347 to 180 words), but less effective for deletion errors (increased from 64 to 80). Effectiveness depends on initial OCR quality and error type, making the method more suitable for documents with high OCR error rates.

Keyword: Damerau-Levenshtein *Distance*, *Optical Character Recognition*, Peter Norvig, *Spelling Correction*, *Character Error Rate*, *Word Error Rate*.

PENDAHULUAN

Digitalisasi administrasi mendorong berbagai organisasi untuk mengotomatisasi proses pengelolaan dokumen guna meningkatkan efisiensi dan kualitas layanan. Salah satu proses yang masih banyak dilakukan secara manual adalah registrasi surat masuk, yaitu dengan memasukkan kembali informasi penting dari dokumen, seperti nomor surat, tanggal, asal surat, perihal dan isi surat, ke dalam sistem. Proses ini memerlukan waktu, terutama ketika volume surat yang diterima tinggi, serta meningkatkan risiko kesalahan pencatatan akibat proses pengetikan ulang.

Salah satu teknologi yang dapat digunakan untuk mengotomatisasi proses tersebut adalah *Optical Character Recognition* (OCR), OCR dapat mengekstraksi informasi teks pada dokumen hasil pindai menjadi teks digital. Namun, akurasi OCR masih dipengaruhi oleh kualitas dokumen dan keberadaan *noise*, seperti logo, cap, tanda tangan, maupun hasil pemindaian yang kurang baik. Kondisi tersebut menyebabkan kesalahan karakter pada hasil ekstraksi sehingga informasi yang diperoleh belum sepenuhnya dapat digunakan secara langsung dalam proses registrasi surat.

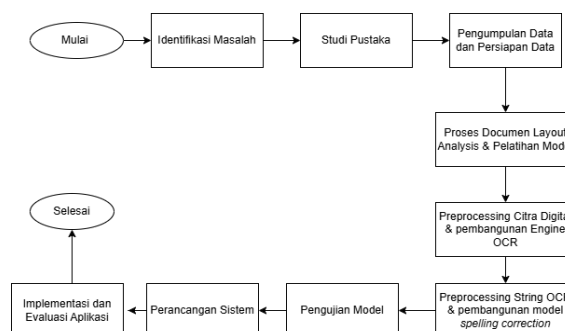
Penelitian Relevan

Penelitian perbandingan *engine* OCR [1] menunjukkan bahwa EasyOCR memiliki performa lebih baik dibandingkan Tesseract OCR, dengan rata-rata *Word Error Rate* (WER) sebesar 25,78% untuk EasyOCR dan 49,69% untuk Tesseract OCR. Selain itu, penelitian mengenai *spelling correction* [2] membuktikan bahwa algoritma Peter Norvig memberikan akurasi lebih baik dibandingkan LSTM dan N-Gram, dengan presentase akurasi masing-masing sebesar 89% untuk Peter Norvig, 75% untuk N-Gram, dan 67%–74% untuk LSTM. Penggunaan Damerau-Levenshtein *Distance* untuk memperbaiki kesalahan ejaan bahasa Indonesia juga menunjukkan tingkat efektivitas hingga 97% [3]. Sementara itu, penelitian mengenai *Document Layout Analysis* [4] menunjukkan bahwa model YOLO mampu mencapai mAP sebesar 92,9%. Meskipun demikian, penerapan *spelling correction* sebagai tahap *post-processing* untuk meningkatkan akurasi hasil OCR pada proses registrasi surat masuk masih terbatas, terutama dalam mengevaluasi pengaruh kualitas hasil OCR terhadap efektivitas proses koreksi.

Berdasarkan penelitian-penelitian tersebut, kajian mengenai efektivitas *spelling correction* sebagai tahap *postprocessing* pada hasil OCR dengan tingkat kualitas yang berbeda masih terbatas. Oleh karena itu, penelitian ini mengimplementasikan metode *spelling correction* berbasis algoritma Peter Norvig, Damerau-Levenshtein *Distance*, dan Sorensen-Dice *Coefficient* untuk memperbaiki hasil ekstraksi OCR pada registrasi surat masuk. Selain mengevaluasi peningkatan akurasi menggunakan metrik *Character Error Rate* (CER) dan *Word Error Rate* (WER), penelitian ini juga menganalisis bagaimana kualitas awal hasil OCR memengaruhi efektivitas proses *spelling correction*.

METODE PENELITIAN

Penelitian dilakukan melalui beberapa tahapan yaitu identifikasi masalah, studi pustaka, pengumpulan data dilakukan melalui observasi, wawancara dan studi literatur dari berbagai sumber seperti Peraturan tentang Tata Naskah Dinas pada surat, serta peraturan pemerintah lain dan referensi jurnal ilmiah yang relevan.



Gambar 1. Tahapan Penelitian

Pengumpulan Dataset

Pada penelitian ini digunakan 2 dataset utama. Dataset pertama terdiri dari 170 dokumen surat masuk *dummy* dengan berbagai *layout* yang berbeda sesuai Tata Naskah Dinas pada beberapa Instansi pemerintah dan *layout* yang umum pada dokumen surat masuk yang digunakan untuk melatih model YOLO dalam mendeteksi area teks pada dokumen. Dataset kedua yaitu pembuatan kamus domain (*domain-specific lexicon*) yaitu kamus yang berisi kosakata yang umumnya sering muncul pada

korespondensi persuratan, seperti kamus domain surat yang di ambil dari 100 data surat masuk, pengumpulan kamus pendukung lain juga dilakukan seperti kamus nama instansi pemerintahan, kamus bulan dan kamus KBBI sebagai acuan pada proses *spelling correction*.

Document Layout Analysis

Document Layout Analysis adalah suatu proses untuk mendeteksi dan melakukan pelabelan terhadap struktur *layout* sebuah dokumen untuk memudahkan proses ekstraksi informasi pada sebuah dokumen dengan cara mengidentifikasi blok-blok homogen didalam dokumen serta menentukan relasi atau hubungan antar blok tersebut [5]. Sejalan dengan hal tersebut, [6] menjelaskan bahwa *Document Layout Analysis* berperan penting dalam mengubah informasi semi-terstruktur pada dokumen menjadi representasi yang terstruktur, sekaligus mengekstraksi informasi kunci yang terkandung di dalamnya.

Pada penelitian ini peneliti mengembangkan dan melatih model YOLO [7] untuk mendeteksi area-area penting pada dokumen surat yang berisi informasi utama, seperti nama instansi, metadata surat, isi surat, nama jabatan, nama pejabat, dan tembusan. Hasil deteksi tersebut kemudian digunakan sebagai dasar proses *cropping* sehingga hanya area yang mengandung informasi penting yang diproses pada tahap *preprocessing* citra dan OCR. Dengan pendekatan ini, sistem dapat mengabaikan objek lain yang tidak diperlukan, seperti logo, cap, tanda tangan, maupun elemen nonteks lainnya, sehingga gangguan (*noise*) pada proses ekstraksi teks dapat diminimalkan dan kualitas hasil OCR menjadi lebih baik.

Model YOLO yang dilatih pada penelitian ini mencapai mAP50 sebesar 0,995 dan mAP50-95 sebesar 0,799, *Precision* sebesar 0,989 dan *Recall* sebesar 0,998 pada data uji, sehingga model mampu mendeteksi area informasi penting pada dokumen surat secara akurat sebelum masuk ke tahap OCR.



Gambar 2. Hasil Deteksi Area Informasi Teks dengan YOLO

Preprocessing Citra

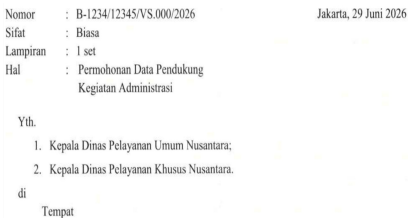
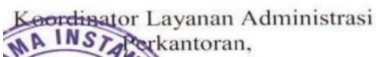
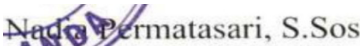
Tahapan *preprocessing* citra bertujuan untuk meningkatkan kualitas citra dan mempersiapkannya agar dapat diproses secara optimal pada tahap *Optical Character Recognition* (OCR). Pada penelitian ini, tahapan *preprocessing* meliputi koreksi kemiringan dokumen (*skew correction*), konversi citra berwarna menjadi citra skala keabuan (*grayscale*), pengurangan gangguan (*noise reduction*) menggunakan *median blur*, serta proses binarisasi (*binarization*) menggunakan metode *adaptive Gaussian thresholding* untuk memisahkan teks dari latar belakang sehingga karakter lebih mudah dikenali oleh OCR [8].

Optical Character Recognition (OCR)

Optical Character Recognition adalah teknologi yang digunakan untuk mengekstraksi dan memproses teks dari citra digital secara otomatis [8]. Pada penelitian ini, proses ekstraksi teks

dilakukan dengan menggunakan PyTesseractOCR. Hasil deteksi area teks dari model YOLO diekstraksi menjadi teks digital yang selanjutnya digunakan sebagai masukan pada tahap *postprocessing*.

Tabel 1. Hasil *Cropping* Deteksi Objek dan Hasil Teks OCR

No	Label Area	Hasil YOLO	Hasil ekstraksi OCR
1	Nama_instansi		kementerian instansi dummy republik idn pusat pelatihan dan pengembangan asn
2	metadata_surat		nomor : b-1234/12345/vs.000/2026 sifat : biasa lampiran : 1 set hal : permohonan data pendukung kegiatan administrasi jakarta, 29 juni 2026 yth. 1. kepala dinas pelayanan umum nusantara; 2. kepala dinas pelayanan khusus nusantara. di tempat
3	nama_jabatan		Koordinator Layanan Administrasi Perkantoran,
4	nama_pejabat		Nadia Permatasari, S.Sos

Sumber: Dokumen Pribadi

Preprocessing String

Preprocessing string adalah tahapan dalam *postprocessing* hasil OCR. Sebelum proses *spelling correction*, teks hasil OCR terlebih dahulu melalui tahap *preprocessing*. *Preprocessing* bertujuan untuk membersihkan dan menormalisasi teks agar memiliki format yang konsisten sebelum diproses oleh model [9].

Pada penelitian ini, tahapan *preprocessing* meliputi tokenisasi, normalisasi, dan *confusion mapping*. Pada tahap tokenisasi, teks dipecah menjadi token-token berupa unit kata. Selanjutnya, dilakukan normalisasi dengan mengubah spasi ganda menjadi spasi tunggal serta mengonversi seluruh token menjadi huruf kecil (*lowercase*) untuk menjaga konsistensi representasi teks. Berbeda dengan *preprocessing* pada umumnya, karakter simbol yang muncul akibat kesalahan OCR tidak langsung dihapus, melainkan dipertahankan agar dapat dimanfaatkan dalam proses pembangkitan kandidat kata. Hal ini dilakukan karena OCR sering kali salah mengenali karakter huruf sebagai simbol atau angka. Oleh karena itu, peneliti menyusun tabel pemetaan (*confusion mapping*) berdasarkan pola kesalahan karakter yang umum terjadi. Pada tahap *confusion mapping*, setiap token diperiksa dan karakter yang teridentifikasi sebagai hasil kesalahan OCR diganti dengan karakter yang sesuai berdasarkan tabel pemetaan tersebut. Sebagai contoh, token "**st@t1stik**" dipetakan menjadi "**statistik**".

Tabel 2. Proses Tokenisasi, Normalisasi dan *Confusion Mapping*

Fungsi	Semula	Menjadi	Keterangan
Tokenisasi	'Badan Pusat St@t1stik Jakarta,'	'Badan', 'Pusat', 'St@t1stik', 'Jakarta,'	Memecah satu kalimat menjadi token per kata
Normalisasi	'Badan', 'Pusat', 'St@t1stik', 'Jakarta,'	'badan', 'pusat', 'st@t1stik', 'jakarta'	merubah token kata menjadi lowercase (huruf kecil)

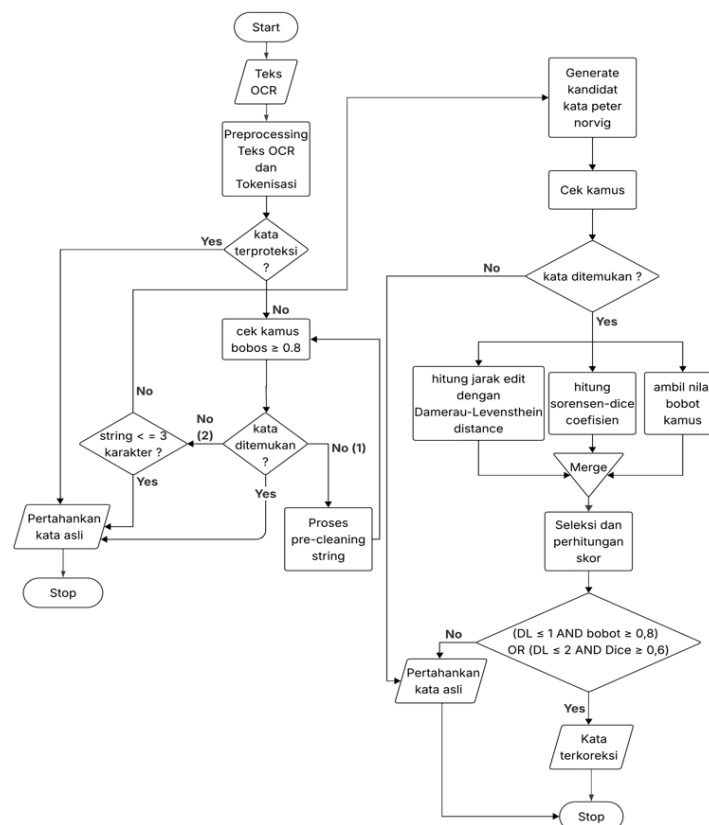
Fungsi	Semula	Menjadi	Keterangan
<i>Confusion Mapping</i>	'badan', 'pusat', 'st@tistik', 'jakarta'	'badan', 'pusat', 'statistik', 'jakarta'	mengganti karakter yang salah terbaca pada OCR menjadi karakter yang benar

Tabel 3. Pemetaan Karakter *Confusion Mapping*

No	Karakter salah terbaca	Karakter Sebenarnya	Karakter salah terbaca	Karakter Sebenarnya
	Karakter huruf menjadi angka		Karakter berdempetan	
1	'0'	'o'	'm'	'm'
2	'1'	'i', 'l', 't'	'vv'	'w'
3	'2'	'z'	'cl'	'd'
4	'3'	'e'	'nn'	'm'
5	'4'	'a'	'li'	'h'
6	'5'	's'	Karakter Simbol	
7	'6'	'b'	'@'	'a', 'e'
8	'7'	't', 'r'	'!'	'i'
9	'8'	'b'	' '	'i', 'l'
10	'9'	'g'	'('	'o', 'c'

Spelling Correction

Setelah *preprocessing* teks dilakukan, tahap selanjutnya adalah *spelling correction*. *Spelling correction* adalah metode untuk memperbaiki kesalahan penulisan pada *string*. Kesalahan dalam penulisan karena adanya karakter berlebih, kurang atau penempatan karakter yang tidak tepat sehingga menghasilkan bentuk kata yang tidak sesuai dengan kosakata baku (*vocabulary*).



Gambar 3. Flowchart Spelling Correction

Token terproteksi

Sebelum dilakukan proses koreksi, setiap token terlebih dahulu diperiksa apakah termasuk kata terproteksi, yaitu token yang berupa angka, kombinasi angka dengan karakter "/" atau "." seperti nomor surat, tanggal, angka atau nominal. Token yang termasuk kategori ini tidak akan diproses lebih lanjut dan langsung dianggap valid. Mekanisme ini diterapkan untuk mencegah sistem melakukan koreksi terhadap informasi yang bersifat tetap, seperti angka, nominal, nomor surat, atau kode dokumen, sehingga keakuratan informasi penting tetap terjaga. Selanjutnya, token yang tidak termasuk kata terproteksi diperiksa terhadap kamus prioritas tinggi, yaitu kamus bulan (bobot 1,0), kamus nama instansi (bobot 0,9), dan kamus surat (bobot 0,8). Apabila token ditemukan pada salah satu kamus tersebut, token langsung dinyatakan valid tanpa melalui proses koreksi. Selain itu, token dengan panjang ≤ 3 karakter juga tidak dikoreksi untuk menghindari terjadinya *over-spelling* atau koreksi yang tidak semestinya.

Untuk token dengan panjang > 3 karakter yang tidak ditemukan pada kamus prioritas, sistem melanjutkan proses *spelling correction* melalui beberapa tahapan, yaitu pembangkitan kandidat kata menggunakan algoritma Peter Norvig, perhitungan jarak edit menggunakan Damerau-Levenshtein *Distance*, serta perhitungan kemiripan *string* menggunakan Sorensen-Dice *Coefficient*.

Pembangkitan Kandidat Kata dengan Peter Norvig

Algoritma Peter Norvig menggunakan pendekatan probabilistik untuk mengoreksi kata yang mengalami kesalahan ejaan dengan membangkitkan kandidat kata melalui operasi *edit distance*, yaitu *insertion* (penyisipan karakter), *deletion* (penghapusan karakter), *substitution* (penggantian karakter), dan *transposition* (pertukaran posisi karakter) [2]. Kandidat koreksi kemudian dipilih berdasarkan probabilitas tertinggi sebagaimana dirumuskan pada rumus berikut:

$$\hat{c} = \operatorname{argmax}_{c \in \text{candidates}} P(w|c) P(c) \quad (1)$$

Persamaan (1) terdiri atas empat komponen, yaitu *argmax* sebagai operator untuk memilih kandidat dengan probabilitas tertinggi, $c \in \text{candidates}$ yang menyatakan himpunan kandidat kata, $P(c)$ sebagai probabilitas kemunculan kandidat dalam korpus (*language model*), dan $P(w|c)$ sebagai probabilitas bahwa kata hasil OCR w berasal dari kandidat kata c (*error model*) (Kusuma & Ratnasari, 2023).

Berbeda dengan metode asli yang menggunakan probabilitas korpus sebagai dasar pemilihan kandidat, pada penelitian ini algoritma Peter Norvig hanya digunakan untuk membangkitkan kandidat kata. Kemudian, setiap kandidat dilakukan pengecekan terhadap kamus, jika kata yang dihasilkan terdapat pada kamus maka kata tersebut kemudian dihitung berdasarkan nilai *edit distance* menggunakan Damerau-Levenshtein *Distance*, tingkat kemiripan string menggunakan Sorensen-Dice *Coefficient*, serta bobot kamus untuk menentukan hasil koreksi terbaik.

Perhitungan Jarak Edit dengan Damerau-Levenshtein *Distance*

Pada penelitian ini, algoritma Damerau-Levenshtein *Distance* digunakan untuk menghitung jarak edit (*edit distance*) antara kata hasil OCR dan kandidat kata yang dihasilkan oleh algoritma Peter Norvig. Perhitungan dilakukan menggunakan empat operasi edit, yaitu *insertion*, *deletion*, *substitution*, dan *transposition* [2]. Jarak edit tersebut dihitung menggunakan persamaan rekursif sebagaimana ditunjukkan pada persamaan berikut:

$$d_{i,j} = \min \begin{cases} 0 & \text{if } i = 0, j = 0 \\ d_{i-1,j} + 1 & \text{if } i > 0 \text{ (Deletion)} \\ d_{i,j-1} + 1 & \text{if } j > 0 \text{ (Insertion)} \\ d_{i-1,j-1} + \text{cost} & \text{if } i, j > 0 \text{ (Substitution)} \\ d_{i-2,j-2} + 1 & \text{if } i, j > 1, a[i] = b[j-1] \text{ and } a[i-1] = b[j] \text{ (Transposition)} \end{cases} \quad (2)$$

Nilai *edit distance* diperoleh dengan membandingkan *cost* dari empat operasi edit, yaitu *deletion*, *insertion*, *substitution*, dan *transposition*. Operasi dengan biaya terkecil dipilih sebagai nilai *edit distance* pada setiap langkah perhitungan. Berbeda dengan tiga operasi lainnya, *transposition* hanya diterapkan apabila terdapat dua karakter yang berdekatan saling bertukar posisi, sehingga pertukaran tersebut dihitung sebagai satu operasi edit.

Perhitungan kemiripan *string* dengan Sorensen-Dice Coefficient

Algoritma Sorensen-Dice Coefficient digunakan untuk mengukur kemiripan antar dua buah string dengan pendekatan bigram [10].

Rumus Sorensen-Dice coefficient:

$$S = \frac{2 \times |X \cap Y|}{|X| + |Y|} \quad (3)$$

Keterangan:

- S : Hasil akhir nilai koefisien kemiripan. Dengan rentang kemiripan jika mendekati nilai 1, maka kemiripan semakin tinggi
- X : Himpunan bigram dari kata target
- Y : Himpunan bigram dari kata input.
- $|X \cap Y|$: Jumlah irisan bigram yang sama antar X dan Y .

Pada penelitian ini, Sorensen-Dice Coefficient digunakan untuk mengukur kemiripan *string* antara kandidat kata yang dihasilkan oleh Peter Norvig dengan kata yang terdapat pada kamus.

Penentuan Kandidat Akhir

Kandidat akhir dipilih berdasarkan kombinasi nilai Damerau-Levenshtein Distance, Sorensen-Dice Coefficient, dan bobot kamus. Kandidat diterima sebagai hasil koreksi apabila memenuhi salah satu dari dua kondisi, yaitu (1) nilai Damerau-Levenshtein Distance ≤ 1 dengan bobot korpus $\geq 0,8$ atau (2) nilai Damerau-Levenshtein Distance ≤ 2 dengan Sorensen-Dice Coefficient $\geq 0,6$. Jika tidak terdapat kandidat yang memenuhi kedua kondisi tersebut, sistem mempertahankan kata asli (*fallback*) untuk meminimalkan *false positive*.

```
-> [Filter Kamus] Kandidat valid edit-1: ['surat', 'surati']
[TOKEN 1][suratt] -> Step 5: Kandidat yang lolos kamus -> ['surati', 'surat']
[TOKEN 1][suratt] -> Step 5: Hitung nilai tiap kandidat:
-> Kandidat: 'surati' | Jarak DL: 1 | Sorensen-Dice: 0.8000 | Bobot Korpus: 0.5
-> Kandidat: 'surat' | Jarak DL: 1 | Sorensen-Dice: 0.8889 | Bobot Korpus: 0.8
[TOKEN 1][suratt] -> Step 5: Hasil terbaik: 'surat' (DL: 1, Bobot: 0.8, Dice: 0.8889)
[TOKEN 1][suratt] -> Step 5: Kata dikoreksi -> 'surat'
```

Gambar 4. Penentuan Kandidat Akhir

Metode Evaluasi Akurasi

Metode evaluasi akurasi digunakan untuk mengukur seberapa baik penggunaan metode *spelling correction* dapat meningkatkan akurasi teks hasil OCR. Beberapa metrik utama yang digunakan untuk mengevaluasi performa tersebut, yaitu *Character Error Rate* (CER) dan *Word Error Rate* (WER). Masing-masing matrik memiliki pendekatan dan kegunaan yang berbeda, tergantung pada tingkat detail (*granularity*) dan tujuan evaluasi yang ingin dicapai [1].

Character Error Rate (CER) digunakan untuk melihat tingkat kesalahan pada level karakter. Metrik ini digunakan untuk mengukur akurasi sebelum perbaikan yaitu hasil ekstraksi *engine* OCR dengan kata asli (*Ground truth*) dan mengukur akurasi setelah dilakukan *spelling correction* dengan kata asli (*Ground truth*). Untuk menghitung CER dapat menggunakan persamaan sebagai berikut:

$$CER(\%) = \frac{S_{char} + D_{char} + I_{char}}{N_{char}} \times 100\% \quad (4)$$

Keterangan:

S_{char} (*Substitution*) : Jumlah karakter yang salah dikenali atau tertukar (contoh: 'O' terbaca '0')
 D_{char} (*Deletion*) : Jumlah karakter yang hilang atau tidak terbaca oleh sistem
 I_{char} (*Insertion*) : Jumlah karakter tambahan yang muncul namun tidak ada pada teks asli.
 N_{char} : Total jumlah seluruh karakter yang ada pada teks asli (*ground truth*).

Word Error Rate (WER) adalah metrik yang digunakan untuk mengukur tingkat kesalahan sistem pada level kata. Berbeda dengan CER yang berfokus pada detail karakter, WER menilai keberhasilan sistem dalam menghasilkan kata yang utuh dan benar.

Rumus perhitungan WER:

$$WER(\%) = \frac{S_{word} + D_{word} + I_{word}}{N_{word}} \times 100 \quad (5)$$

Keterangan:

S_{word} (*Substitution*) : Jumlah kata yang salah dikenali
 D_{word} (*Deletion*) : Jumlah kata yang hilang atau tidak terbaca
 I_{word} (*Insertion*) : Jumlah kata tambahan yang tidak ada pada data asli.
 N_{word} : Total jumlah kata pada data asli (*ground truth*).

Persentase penurunan digunakan untuk mengetahui besarnya penurunan nilai CER dan WER setelah penerapan *spelling correction* dibandingkan dengan nilai sebelum penerapan *spelling correction*. Persentase penurunan dihitung dengan membandingkan selisih nilai sebelum dan sesudah *spelling correction* terhadap nilai sebelum *spelling correction*.

$$P = \frac{CER_{OCR} - CER_{SC}}{CER_{OCR}} \times 100\% \quad (6)$$

$$P = \frac{WER_{OCR} - WER_{SC}}{WER_{OCR}} \times 100\% \quad (7)$$

HASIL DAN PEMBAHASAN

Evaluasi dilakukan terhadap 50 sampel surat dengan menggunakan metrik *Character Error Rate* (CER) dan *Word Error Rate* (WER), dihitung dengan membandingkan teks hasil OCR dan teks hasil *spelling correction* terhadap *ground truth*. Selanjutnya, dilakukan analisis terhadap perubahan nilai CER dan WER sebelum dan sesudah penerapan *spelling correction*. Analisis tambahan dilakukan terhadap distribusi jenis kesalahan *substitution* (S), *deletion* (D), dan *insertion* (I) untuk mengidentifikasi penyebab perubahan WER pada tiap kategori.

Hasil rata-rata CER dan WER dari 50 sampel surat sebelum dan sesudah penerapan *spelling correction*, beserta persentase penurunannya, disajikan pada Tabel 4.

Tabel 4 . Rata-Rata CER dan WER Sebelum dan Sesudah *Spelling Correction*

No	Metrik	Sebelum <i>Spelling Correction</i>	Setelah <i>Spelling Correction</i>	Selisih	Persentase Penurunan Error
1	CER	1.85%	1.54%	0.31%	16.55%
2	WER	6.75%	4.46%	2.29%	33.88%

* **Character Error Rate (CER):** menunjukkan tingkat kesalahan pada level karakter.
 * **Word Error Rate (WER):** menunjukkan tingkat kesalahan pada level kata
 * Semakin kecil nilai CER dan WER menunjukkan akurasi sistem yang semakin baik, sedangkan semakin besar nilainya menunjukkan tingkat kesalahan sistem yang semakin tinggi

Berdasarkan hasil pengujian CER dan WER pada Tabel 4, dari 50 sampel dokumen yang diuji, sebanyak 26 dokumen menunjukkan peningkatan akurasi setelah diterapkan *spelling correction*. Peningkatan terbesar terjadi pada dokumen 1–6 yang memiliki nilai CER dan WER hasil OCR relatif tinggi, sehingga menunjukkan bahwa metode yang diusulkan lebih efektif pada hasil OCR yang masih mengandung banyak kesalahan. Sebaliknya, sebanyak 9 dokumen mengalami penurunan akurasi pada kedua metrik. Sementara itu, dokumen lainnya menunjukkan pola perubahan CER dan WER yang tidak konsisten, yaitu 6 dokumen dengan nilai CER tetap tetapi WER meningkat, 1 dokumen dengan CER menurun tetapi WER meningkat, 1 dokumen dengan CER meningkat tetapi WER tetap, 1 dokumen dengan CER menurun tetapi WER tetap, serta 3 dokumen dengan CER tetap tetapi WER menurun, dan 3 dokumen lainnya dengan CER dan WER yang sama-sama tidak mengalami perubahan (tetap). Hasil tersebut menunjukkan bahwa perubahan pada tingkat karakter tidak selalu sejalan dengan perubahan pada tingkat kata. Oleh karena itu, analisis selanjutnya difokuskan pada metrik WER sebagai indikator utama untuk mengevaluasi kualitas hasil OCR dan *spelling correction*. Untuk memperoleh pemahaman yang lebih rinci mengenai penyebab perubahan tersebut, dilakukan analisis distribusi jenis kesalahan *substitution* (S), *deletion* (D), dan *insertion* (I) pada hasil OCR maupun hasil *spelling correction*, sebagaimana disajikan pada Tabel 5.

Tabel 5. Rekapitulasi Distribusi Jenis Kesalahan *Word Error Rate* (WER)

No	Jenis Kesalahan	Surat Terdampak	Total S/D/I (OCR)	Total S/D/I (SC)	Turun	Naik	Tetap
1	<i>Substitution</i>	48	347	180	31	8	9
2	<i>Deletion</i>	17	64	80	0	13	4
3	<i>Insertion</i>	18	36	33	3	15	0

Sumber: Dokumen Pribadi

Berdasarkan rekapitulasi pada Tabel 5, kesalahan *substitution* merupakan jenis kesalahan yang paling dominan pada hasil OCR, yaitu 347 kesalahan kata pada 48 dari 50 sampel surat. Setelah diterapkan *spelling correction*, jumlahnya menurun menjadi 180 kesalahan kata, dengan 31 surat mengalami penurunan, 9 surat tetap, dan 8 surat meningkat akibat *overcorrection*. Hasil ini menunjukkan bahwa metode yang diusulkan efektif mengurangi kesalahan *substitution* karena token hasil OCR masih dapat dibangkitkan kandidat koreksinya.

Sebaliknya, kesalahan *deletion* meningkat dari 64 menjadi 80 kesalahan kata, ditemukan pada 17 sampel surat, tanpa adanya surat yang mengalami penurunan. Sementara itu, kesalahan *insertion* sedikit menurun dari 36 menjadi 33 kesalahan kata pada 18 sampel surat, meskipun sebagian besar sampel masih mengalami penambahan kesalahan. Kedua hasil tersebut menunjukkan bahwa metode yang diusulkan belum mampu menangani kesalahan akibat hilangnya token maupun perubahan struktur token, karena koreksi hanya dilakukan pada tingkat ejaan.

Secara keseluruhan, *spelling correction* paling efektif mengurangi kesalahan *substitution*, sedangkan efektivitasnya terbatas pada kesalahan *deletion* dan *insertion*. Temuan ini menjelaskan bahwa peningkatan akurasi pada Tabel 4 terutama berasal dari berkurangnya kesalahan *substitution*,

sedangkan peningkatan WER pada sebagian sampel dipengaruhi oleh *overcorrection* serta keterbatasan metode dalam memperbaiki token yang hilang maupun struktur token yang berubah.

SIMPULAN

Penelitian ini berhasil menerapkan metode *Spelling Correction* menggunakan algoritma Peter Norvig, Damerau-Levenshtein *Distance*, dan Sorensen–Dice *Coefficient* sebagai tahap *post-processing* hasil OCR pada sistem registrasi surat masuk. Metode yang diusulkan didukung oleh *confusion mapping*, pembagian kamus ke dalam empat kategori sesuai konteks persuratan, serta model YOLO untuk mendeteksi area informasi penting sehingga proses OCR lebih terfokus pada bagian yang relevan. Berdasarkan pengujian menggunakan *Character Error Rate* (CER) dan *Word Error Rate* (WER), metode ini mampu meningkatkan akurasi hasil OCR, terutama pada dokumen dengan kualitas pemindaian rendah yang memiliki tingkat kesalahan OCR tinggi. Peningkatan akurasi terutama berasal dari berkurangnya kesalahan *substitution*, sedangkan penurunan akurasi pada sebagian dokumen dipengaruhi oleh *overcorrection*, meningkatnya kesalahan *deletion* dan *insertion*, serta keterbatasan metode dalam menangani perubahan struktur token. Dengan demikian, metode yang diusulkan lebih efektif diterapkan pada dokumen yang masih memiliki tingkat kesalahan OCR relatif tinggi dibandingkan pada dokumen dengan hasil OCR yang telah memiliki akurasi tinggi.

Penelitian selanjutnya disarankan untuk: (1) menambahkan mekanisme deteksi kata yang benar-benar terindikasi salah sebelum dikoreksi guna mengurangi *overcorrection*, khususnya pada dokumen berkualitas baik; (2) memperluas cakupan kamus, terutama istilah teknis, nama instansi, dan nama jabatan; (3) mengembangkan metode ekstraksi metadata surat yang lebih fleksibel dan tidak terlalu bergantung pada hasil OCR; serta (4) mengeksplorasi pendekatan *contextual spelling correction* berbasis *language model* yang mempertimbangkan hubungan antar-kata dalam kalimat, sehingga permasalahan *token splitting/merging* dan pemilihan kata sesuai konteks dapat ditangani lebih baik.

DAFTAR PUSTAKA

- [1] M. N. Darpito, K. Firdausy, and A. Fadlil, “JIP (Jurnal Informatika Polinema) Perbandingan Unjuk Kerja Library Optical Character Recognition (OCR) dalam Pengenalan Teks pada Dokumen Digital,” 2025.
- [2] A. T. A. Kusuma and C. I. Ratnasari, “COMPARISON OF SPELL CORRECTION IN BAHASA INDONESIA: PETER NORVIG, LSTM, AND N-GRAM,” *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 3, pp. 214–220, Dec. 2023, doi: 10.33387/jiko.v6i3.7072.
- [3] A. P. Wibawa, P. Yuliawati, P. Santoso, R. Shalahuddin, and I. M. Wirawan, “Damerau Levenshtein Distance dengan Metode Empiris untuk Koreksi Ejaan Bahasa Indonesia,” *ILKOM Jurnal Ilmiah*, vol. 12, no. 3, pp. 176–182, Dec. 2020, doi: 10.33096/ilkom.v12i3.600.176-182.
- [4] M. Li *et al.*, “DocBank: A Benchmark Dataset for Document Layout Analysis,” Online, 2020. [Online]. Available: <https://github.com/doc-analysis/DocBank>.
- [5] G. M. Binmakhashen and S. A. Mahmoud, “Document layout analysis: A comprehensive survey,” Nov. 30, 2020, *Association for Computing Machinery*. doi: 10.1145/3355610.
- [6] D. L. Li, S. K. Lee, and Y. T. Liu, “Printed document layout analysis and optical character recognition system based on deep learning,” *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-07439-y.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” May 2016, [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [8] K. Abdulwahhab Hamad and M. Kaya, “A Detailed Analysis of Optical Character Recognition Technology,” 2013. [Online]. Available: <http://ijamec.atscience.org>
- [9] A. Miyajiwal, A. Ladkat, S. Jagadale, and R. Joshi, “On Sensitivity of Deep Learning Based Text Classification Algorithms to Practical Input Perturbations,” Jan. 2022, doi: 10.1007/978-3-031-10464-0_42.
- [10] A. H. Pardosi, D. Enda, and D. Direvisi, “Rancang Bangun Aplikasi Latihan Pengucapan Bahasa Inggris Berbasis Speech Recognition dengan Metode Sorensen-Dice Coefficient,” 2025.