PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

Optimalisasi Image Thresholding Pada Optical Character Recognition Pada Sistem Digitalisasi dan Pencarian Dokumen

Ridwan Rismanto[1]; Arief Prasetyo[2]; Dyah Ayu Irawati[3]

[1, 2, 3]Politeknik Negeri Malang
[1]rismanto@polinema.ac.id

ABSTRACT

The administration activity in an institute is largerly done using a paper based mailing and document as a media. Therefore, a great effort needs to be performed in the case of management and archiving, in the form of providing storage space through the categorization system. Digitalization of document by scanning it into a digital image is one of the solution to reduce the effort to perform the work of archiving and categorizing such documents. It also provide searching feature in the form of metadata, that is manually written during the digitalization process. The metadata can contains the title of document, summary, or category. The needs to manually input this metadata can be solved by utilizing Optical Character Recognition (OCR) that converts any text in the document into readable text storing in the database system. This research focused on the implementation of the OCR system to extract text in the scanned document image as the input for the additional metadata. Thresholding optimization of the pre-processing stage was done by testing the threshold value with the best limit value is 0.6. Experiment performed by processing text extraction towards several scanned document and achieving accuration rate of 92.568%.

Keywords: *OCR, Image Processing, Document Digitalization*

ABSTRAK

Kegiatan administrasi pada suatu institusi dilakukan melalui media dokumen dan persuratan berbasis kertas (paper based). Untuk itu dibutuhkan effort yang besar dalam hal manajemen dan pengarsipannya mulai dari penyediaan tempat penyimpanan hingga pengkategoriannya. Digitalisasi dokumen adalah salah satu solusi untuk mempermudah pekerjaan pengarsipan dokumen tersebut. Proses pencarian dokumen yang sudah disimpan dalam bentuk file citra dimungkinkan melalui metadata yang harus diinputkan sebelumnya saat proses scan dokumen. Ketergantungan terhadap metadata tersebut menyebabkan keterbatasan pada fitur pencarian, karena metadata tidak mencakup keseluruhan isi dokumen. Optical Character Recognition (OCR) dapat diimplementasikan untuk mengekstrak teks yang ada di citra dokumen agar dapat disimpan dan diolah lebih lanjut melalui antar muka aplikasi, sehingga dimungkinkan untuk melakukan pencarian pada keseluruhan isi dokumen. Penelitian ini memfokuskan pada implementasi OCR untuk ekstraksi keseluruhan teks dari citra dokumen sebagai input untuk metadata tambahan. Optimalisasi thresholding pada proses pre-processing dilakukan dengan menguji nilai ambang dengan hasil nilai ambang terbaik adalah 0.6. Pengujian dilakukan dengan membandingkan tingkat kemiripan antara teks asli yang ada di dokumen dengan teks hasil ekstraksi, yang menghasilkan akurasi yaitu rata-rata kemiripan 92.568%.

Kata kunci: *OCR, Image Processing, Digitalisasi Dokumen*

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

1. PENDAHULUAN

Proses digitalisasi dokumen sudah banyak membantu dalam hal pengarsipan dokumen, mulai dari dokumen berupa buku, manuskrip kuno, mapun kegiatan surat menyurat atau administrasi. Digitalisasi dokumen dapat menghasilkan efisiensi dalam hal tempat penyimpanan dan pengorganisasiannya. Selain itu dokumen yang sudah diarsipkan secara digital akan terjaga keawetannya karena tidak tergantung pada bentuk fisik yang rawan rusak jika terjadi bencana seperti banjir, kebakaran, dan juga meminimalisir resiko kehilangan, karena proses *backup* dapat dengan mudah dilakukan.

Dokumen yang telah disimpan dalam bentuk digital notabene berbentuk file gambar hasil dari proses scan. Pencarian yang dapat dilakukan terhadap dokumen tersebut hanya dimungkinkan melalui metadata yang telah diinputkan sebelumnya seperti judul dokumen, tahun pembuatan, penulis, dan sebagainya tergantung dari jenis dokumennya. Pencarian terhadap isi dokumen itu sendiri tidak bisa secara langsung dilakukan dikarenakan format dokumen yang berupa gambar. *Optical Character Recognition* (OCR) merupakan salah satu solusi untuk mengekstrak teks yang ada didalam gambar menjadi teks yang dapat disimpan dan diolah lebih lanjut oleh program komputer.

Penelitian sebelumnya mengenai sistem pengarsipan dokumen BAAK menggunakan OCR pada STMIK Palcolmtech Palembang, menerapkan OCR untuk mengambil segmen tertentu pada form administrasi mahasiswa untuk diekstrak teks-nya kedalam database [6]. Penelitian tersebut menghasilkan error rate sebesar 2.14%. Penelitian lain mengenai OCR dilakukan pada aplikasi text to speech berbasis backpropagation pada aplikasi Android, dapat mengekstrak teks yang diambil melalu kamera smartphone untuk kemudian diubah menjadi suara menggunakan API dari *Google Text to Speech*. Akurasi yang dihasilkan sebesar 97.58% dengan rata-rata 94.7% dengan kondisi pengambilan gambar tertentu [7]. Penelitian berikutnya mengimplementasikan template matching correlation untuk OCR untuk mengenali karakter teks dengan jenis font tertentu yaitu Arial, Times New Roman, Comic Sans MS, Cambria, dan Courier New dengan tingkat keberhasilan 92.90% [8]. Mengacu pada beberapa penelitian tersebut, OCR sangat memungkinkan untuk mengekstrak teks dari citra dokumen dengan tingkat keberhasilan yang cukup baik.

Penelitian ini akan mengimplementasikan OCR dan melakukan optimalisasi *Image Thresholding* untuk mendapatkan nilai akurasi tertinggi pada proses ekstraksi teks.

2. METODE/PERANCANGAN PENELITIAN

2.1. *Optical Character Recognition*

Optical Character Recognition (OCR) adalah sebuah metode yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk dikonversi ke dalam bentuk file tulisan [9]. Sistem pengenal huruf ini dapat meningkatkan fleksibilitas atau kemampuan dan kecerdasan sistem komputer. Sistem pengenal huruf yang cerdas sangat membantu usaha besar-besaran yang saat ini dilakukan banyak pihak yakni usaha digitalisasi informasi dan pengetahuan, misalnya dalam pembuatan koleksi pustaka digital, koleksi sastra kuno digital, dan lain-lain.

Secara umum proses OCR dapat dilihat pada Gambar 1, dengan penjelasan sebagai berikut:

1. File Input
   File input berupa file citra digital denganformat *.bmp atau *.jpg.
2. *Preprocessing*
   Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

3. Segmentasi
   Segmentasi adalah proses memisahkan area pengamatan (*region*) pada tiap karakter yang dideteksi.
4. Normalisasi
   Normalisasi adalah proses merubah dimensi region tiap karakter dan ketebalan karakter.
5. Ekstraksi ciri
   Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.
6. *Recognition*
   Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.

Gambar 1. Proses OCR Secara Umum

OCR secara garis besar bekerja dengan mengkonversi gambar secara elektronik atau dengan kata lain sebagai contoh adalah tulisan tangan yang di-encode menjadi sebuah transkrip yang dipahami secara utuh oleh mesin dalam bentuk gambar manuskrip atau dari teks yang deskriptif [3][4]. Jenis proses ini secara luas berfungsi untuk entri data dari data cetak yang mencakup brosur seperti ID, berbagai pembelian atau jenis faktur lainnya, penerimaan bank yang dihasilkan melalui mesin, cetakan informasi, semua makalah resmi yang sesuai yang merupakan rutinitas mendigitalkan dokumen yang tersedia yang dapat diedit secara otomatis, diperiksa dan disimpan secara lebih profesional dan juga digunakan dalam aksi mesin seperti berbagai konversi mesin, pengaturan *text-to-speech*, kunci eksplorasi data dan teks [5].

Pengenalan karakter adalah bidang penelitian dalam pengakuan berbagai pola pembelajaran, pembelajaran mesin dan bidang visi komputer [1]. Salah satu contoh penerapan OCR adalah berperan dalam pengenalan karakter yang dalamketentuan digit dan huruf serta analisis tulisan tangan [2].

2.2. Binerisasi dan *Image Thresholding*

Binerisasi dilakukan untuk mengubah citra menjadi derajat keabuan (*grayscale*). Pada tahap proses binerisasi, file citra digital dikonversi menjadi citra biner. Citra biner (*binary image*) adalah citra yang hanya memiliki dua nilai derajat keabuan, yaitu hitam dan putih. Pixel-pixel objek bernilai 1 dan pixel-pixel latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi, pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam [10].

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

Konversi dari citra hitam-putih ke citra biner dilakukan dengan menggunakan operasi thresholding. Operasi ini mengelompokkan nilai derajat keabuan setiap pixel ke dalam 2 kelas, yaitu hitam dan putih. Salah satu metode *thresholding* adalah *Otsu Thresholding*.

Otsu Thresholding adalah sebuah teknik *thresholding* yang diperkenalkan oleh Nobuyuki Otsu, yang secara otomatis mencari batas ambang terbaik untuk citra yang diolah [11]. Metode Otsu menghitung nilai ambang T secara otomatis berdasarkan citra masukan. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis Diskriminan berfungsi memaksimumkan variabel tersebut agar dapat memisahkan objek dengan latar belakang [12].

Nilai ambang yang dicari dari suatu citra hitam-putih dinyatakan dengan k . Nilai k berkisar antara 1 sampai dengan L, dengan nilai L = 255. Probabilitas setiap pixel pada level ke i dapat dinyatakan dalam persamaan 1 [12].

$$p_i = \frac{n_i}{N} \quad (1)$$

Keterangan:
n! menyatakan jumlah pixel pada level ke i
N menyatakan total jumlah pixel pada citra.

Nilai momen kumulatif ke nol $\omega\ k$ dapat dinyatakan dalam persamaan 2.3, momen kumulatif ke satu $\mu\ k$ dapat dinyatakan dalam persamaan 2.4, dan nilai rata-rata μ! Dapat dinyatakan dalam persamaan 2.

$$\omega(k) = \sum_{i=1}^{k} p_i \ldots$$

$$\mu(k) = \sum_{i=1}^{k} i \cdot p_i$$

$$\mu_T = \sum_{i=1}^{L} i \cdot p_i \ldots \quad (2)$$

Nilai ambang k dapat ditentukan dengan memaksimumkan persamaan 3.

$$\sigma_B^2(k^*) = \max_{1 \le k \le L} \sigma_B^2(k) \quad (3)$$

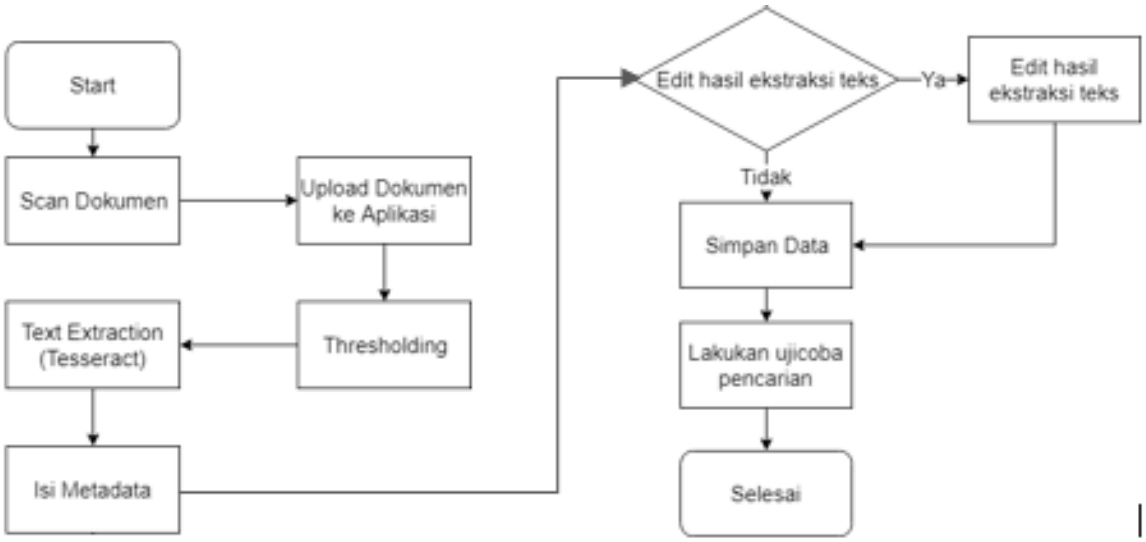
Dimana nilai $\sigma_B^2(k)$ dapat dihitung menggunakan persamaan 4.

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1-\omega(k)]} \quad (4)$$

2.3. Penerapan OCR

Penerapan OCR pada aplikasi ini meliputi proses scan dokumen, upload dokumen kedalam aplikasi, proses OCR (thresholding, text extraction), isi metadata (kategori surat, tahun, deskripsi singkat), cek hasil ekstraksi teks (edit bila diperlukan), simpan data, ujicoba pencarian. Flowchart berikut ini menjelaskan proses diatas.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659



Gambar 2. Flowchart Penerapan OCR

2.4. Pengujian

Pengujian akan dilakukan terhadap *similarity* atau tingkat kemiripan antara teks yang ada pada dokumen asli dengan teks hasil ekstraksi. Algoritma pengecekan kemiripan yang digunakan adalah *Aprroximate String Matching* [13].

Untuk mengukur tingkat akurasi dari kemiripan total semua dokumen, digunakan persamaan sebagai berikut:

$$A = \frac{\sum_{d1}^{dn} Ds}{N} * 100\% \quad (5)$$

Keterangan:
A menyatakan tingkat akurasi dari kemiripan
Ds menyatakan tingkat similaritas dokumen
N menyatakan jumlah data

3. HASIL DAN PEMBAHASAN

3.1. Pengujian *Pre-Processing*

Tahap *pre-processing* ini adalah proses pengolahan data hasil scan input dokumen sebelum diproses oleh modul *optical character recognition*. *Pre-processing* yang dilakukan disini adalah proses *thresholding* pada dokumen hasil scan yang berformat gambar (JPG) dengan metode *Otsu Thresholding*. Tujuan dari dilakukannya pre-processing ini adalah untuk membersihkan gambar/*image* hasil scan sehingga hasil ekstraksi karakternya pun akan lebih baik. Berikut ini adalah hasil *thresholding* pada data scan hasil dokumen dengan resolusi gambar berukuran 1700 x 2338 dengan kedalaman dot per inch yaitu 200 DPI.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

Gambar 3. Citra asli, thresholding dengan nilai ambang 0.3, 0.6 dan 0.9

Dapat dilihat dari Gambar 3 diatas bahwa semakin besar nilai ambang batas maka semakin tinggi nilai kontrasnya namun hasilnya kurang bersih dibandingkan dengan nilai ambang batas kecil. Namun terlalu kecil nilai ambang batas, maka gambar yang dihasilkan juga semakin samar. Maka perlu diuji coba berapakah ambang batas yang paling optimal untuk diproses pada *optical character recognition*.

3.2. Pengujian Ekstraksi Teks

Proses ektraksi teks dilakukan melalui beberapa tahap, antara lain: *line removal*, *zoning*, *tokenization* dan *pattern recognition*.

a. *Line removal*

Proses ini mengidentifikasi objek-objek garis dan marking yang bukan merupakan karakter. Proses ini perlu dilakukan agar proses pengenalan karakter dapat dilakukan dengan baik. Hasil *line removal* dapat dilihat pada Gambar 4 yang merupakan *cropping* dari *image* hasil *thresholding*.

Gambar 4. Hasil proses line removal

b. *Zoning*

Proses ini memisahkan bagian dari gambar menjadi partisi-partisi teks, misal seperti mendeteksi kolom-kolom jika dokumen yang akan diproses merupakan dokumen multi-kolom. Hasil *zoning* dapat dilihat pada Gambar 5 berikut ini.

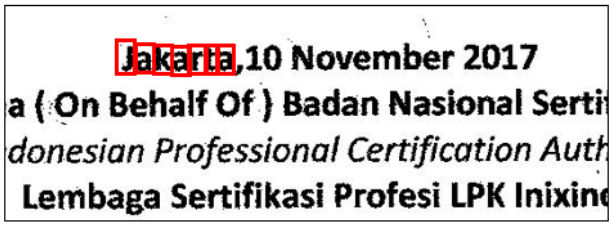
Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

Gambar 5. Zoning

c. *Tokenization*

Pertama-tama, proses ekstraksi teks akan menentukan bagian-bagian dari setiap baris pada gambar yang sudah melalui proses *zoning.* Setiap baris karakter akan diproses satu demi satu.

Pada setiap baris karakter, proses ekstraksi teks akan mengidentifikasi *spacing* atau jarak antar karakter dengan cara memeriksa garis-garis vertikal pada pixel yang bukan merupakan teks. Setiap bagian dari pixel yang berada diantara garis-garis yang bukan merupakan teks, akan ditandai sebagai "*token*" yang merepresentasikan satu buah karakter. Hasil *tokenization* bisa dilihat pada Gambar 6 berikut ini.



Gambar 6. Tokenization

d. *Pattern Recognition*

Proses berikutnya adalah *pattern recognition*. Pada proses ini, setiap token akan dibandingkan per pixel nya dengan sebuah dataset karakter yang sudah disediakan atau disebut juga dengan *glyphs*. Termasuk didalamnya adalah angka, tanda baca, dan simbol-simbol yang lain. *Token* yang paling mirip dengan dataset akan dikenali sebagai karakter yang sesuai dengan dataset tersebut. Teknik ini disebut juga dengan *matrix matching*. Hasil *matriks matching* dapat dilihat pada Gambar 7 berikut ini.

Token							
Karakter	J	a	k	a	r	t	a

Gambar 7. *Pattern Recognition*

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

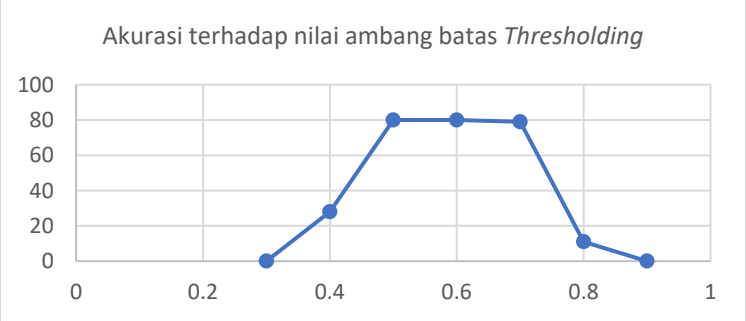
3.3. Nilai Ambang *Thresholding*

Pada penelitian ini diuji tingkat akurasi untuk setiap nilai ambang yang berbeda-beda pada saat proses *pre-processing* yaitu proses *image thresholding*. Dengan percobaan nilai ambang antara lain 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9. Hasil pengujian akurasi untuk beberapa nilai ambang batas dapat dilihat pada Tabel 1 berikut ini. Pengujian ini dilakukan untuk menentukan batas nilai ambang untuk proses thresholding yang paling baik berdasarkan nilai kemiripan teks hasil ekstraksi dengan teks asli dokumen dengan mengabaikan perpindahan baris.

Untuk membandingkan teks asli yang ada pada dokumen dengan teks hasil ekstraksi, digunakan algoritma *Approximate String Matching* [13].

Tabel 1. Hasil pengujian ambang batas

Nilai ambang batas	Jumlah karakter terekstraksi	*Similarity*
0.3	0	0%
0.4	959	28%
0.5	1.025	80%
0.6	1.025	80%
0.7	1.019	79%
0.8	677	11%
0.9	52	0%



Gambar 8. Grafik akurasi terhadap nilai ambang *Thresholding*

Dari tabel diatas dapat diamati bahwa nilai ambang batas yang terlalu besar ataupun terlalu kecil akan berdampak pada penurunan akurasi dari proses ekstraksi teks. Hal ini dikarenakan oleh beberapa faktor. Faktor pertama adalah pada nilai ambang yang terlalu kecil, kontras antara karakter dengan *background* dokumen terlalu tipis, mengakibatkan karakter sulit terdeteksi karena semakin samar. Bahkan pada nilai ambang 0.2, program tidak mengenali teks sama sekali, sehingga nilai similarity 0%.

Faktor berikutnya adalah *noise*. Pada nilai ambang batas yang besar, yaitu 0.9, program juga gagal mengenali teks dan memberikan hasil berupa teks yang jauh dari seharusnya. Hal ini dikarenakan pada nilai ambang yang besar, gambar cenderung lebih gelap, dan malah memperbesar noise yang ada pada dokumen. Sehingga hal ini pun mempengaruhi kemampuan program dalam mengenali teks yang ada dalam dokumen.

Pada nilai ambang batas 0.5 dan 0.6, proses ekstraksi teks menghasilkan tingkat *similarity* tertinggi yaitu 80%. Dapat dilihat pula bahwa baik nilai ambang 0.5 dan 0.6 memberikan hasil yang hampir sama. Hanya saja, pada nilai ambang 0.5 dokumen akan lebih bersih dibandingkan 0.6. Maka untuk pengujian selanjutnya akan digunakan nilai ambang 0.5 dan 0.6 tersebut.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

3.4. Hasil Pengujian Akurasi

Dengan menggunakan nilai ambang yang sudah diuji-cobakan pada pengujian sebelumnya yaitu 0.5 dan 0.6, program akan diuji untuk melakukan ekstraksi teks terhadap beberapa dokumen. Pengujian dilakukan dengan membandingkan kesamaan antara teks hasil ekstraksi program dengan teks yang ada di dokumen dengan mengabaikan karakter ganti baris. Perbandingan antara teks asli pada dokumen dengan teks hasil ekstraksi dilakukan menggunakan algoritma *Approximate String Matching* [13]. Dokumen yang digunakan adalah hasil *scan* dengan kedalaman pixel 200 DPI.

Berikut adalah hasil pengujian *similarity* dengan nilai ambang 0.5 dan 0.6 yang dapat dilihat di Tabel 2 dan 3.

Tabel 2. Hasil pengujian *similarity* ekstraksi teks dengan nilai ambang 0.5

Dokumen	Resolusi	$t_{extraction}$ (s)	% similarity
Dokumen 1	1700 x 2338	6.230	81.52
Dokumen 2	1700 x 2338	3.588	91.4
Dokumen 3	2550 x 3510	2.392	96.12
Dokumen 4	1700 x 2338	2.129	94.03
Dokumen 5	1700 x 2338	2.596	88.5

Tabel 3. Hasil pengujian *similarity* ekstraksi teks dengan nilai ambang 0.6

Dokumen	Resolusi	$t_{extraction}$ (s)	% similarity
Dokumen 1	1700 x 2338	3.782	91.38
Dokumen 2	1700 x 2338	3.440	93.4
Dokumen 3	2550 x 3510	2.665	96.92
Dokumen 4	1700 x 2338	2.502	92.64
Dokumen 5	1700 x 2338	3.287	88.5

Dari hasil ujicoba untuk nilai ambang 0.5 dan 0.6, maka didapatkan nilai akurasi yaitu:

a. Nilai ambang 0.5

$$A = \frac{81.52 + 91.4 + 96.12 + 94.03 + 88.5}{5}$$

$$= 90.314\%$$

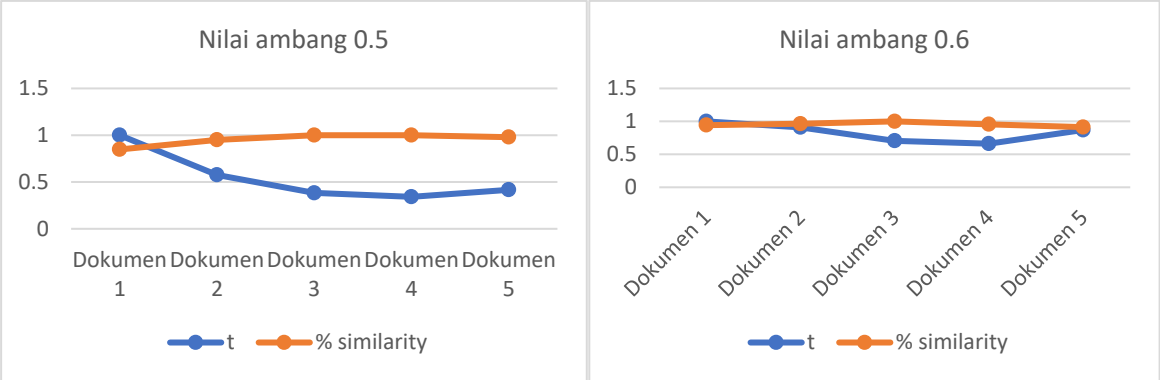
b. Nilai ambang 0.6

$$A = \frac{91.38 + 93.4 + 96.92 + 92.64 + 88.5}{5}$$

$$= 92.568\%$$

Dari perhitungan diatas didapatkan bahwa nilai akurasi OCR untuk nilai ambang *Thresholding* 0.5 adalah 90.314% dan 0.6 adalah 92.568%. Maka nilai ambang terbaik adalah 0.6. Grafik yang menunjukkan korelasi antara waktu proses ekstraksi teks dengan hasil akurasi untuk nilai ambang 0.5 dan 0.6 dapat dilihat pada Gambar 9 berikut ini. Kecenderungan yang ada adalah semakin tinggi akurasi ekstraksi teks, maka semakin cepat waktu eksekusinya.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659



Gambar 9. Grafik korelasi kecepatan ekstraksi dan akurasi dengan nilai ambang 0.5 dan 0.6

4. KESIMPULAN DAN SARAN

Proses ekstraksi teks yang dengan teknik *Optical Character Recognition* dilakukan dengan diawali tahap *pre-processing* yaitu *Image Thresholding* dengan beberapa nilai ambang yaitu 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 dan 0.9. Dari hasil percobaan, nilai ambang yang menghasilkan akurasi tertinggi adalah 0.5 dan 0.6.

Pengujian dengan beberapa dokumen yang semuanya berasal dari digitalisasi dokumen menggunakan alat *scanner*, menunjukkan hasil akurasi tertinggi diperoleh dengan nilai ambang 0.6 dengan nilai akurasi 92.568%.

Metodologi *pre-processing* yang dilakukan pada penelitian ini adalah terbatas pada *Image Thresholding* untuk membersihkan *noise* dan mengubah citra hasil scan dokumen menjadi *grayscale*. Akan lebih baik lagi jika ditambahkan proses *de-skewing* untuk memperbaiki orientasi citra hasil scan agar benar-benar tegak. Dengan demikian maka akurasi akan lebih baik lagi.

DAFTAR PUSTAKA

[1] Shrinivas R. Zanwara , Sandipann P. Naroteb , Abhilasha S. Narotec , Ulhas B. Shindea, 2019, An Effectual Optical Character Recognition Using Efficient Learning System, *International Conference on Sustainable Computing in Science, Technology & Management* (SUSCOM-2019).

[2] H.S. King, HKI Systems and Service LLC, 2018, Processing container images and identifiers using optical character recognition and geo-location, *U.S. Patent Application 10/043,154*, 2018.

[3] M.F. Esmaile, E. Abdulredaa, 2018, Optical Character Recognition Using Active Contour Segmentation. *Journal of Engineering* 24, no. 1, pp: 146-158, 2018.

[4] A. Longacre, 2014, Method for omnidirectional processing of 2D images including recognizable characters, *U.S. Patent* 8,682,077, issued March 25, 2014.

[5] F. Mohammad, J. Anarase, M. Shingote, P. Ghanwat, 2014, Optical character recognition implementation using pattern matching, *International Journal of Computer Science and Information Technologies* 5, no. 2, pp: 2088-2090, 2014.

[6] Triwahyuni, Atin 2011, Aplikasi E-Arsip Pada STMIK Palcomtech Palembang, *Seminar Nasional Informatika, UPN "Veteran" Yogyakarta*, 2 Juli 2011.

[7] Apriyanti, Kristina dkk 2016, Implementasi Optical Character Recognition Berbasis Backpropagation untuk Text to Speech Perangkat Android, *IJEIS*, Vol.6, No.1, pp. 13~24.

[8] Hartanto, Suryo, dkk 2015, Optical Characte Recognition Menggunakan Algoritma Template Matching Correlation, *Jurnal Masyarakat Informatika*, Vol.5, No.9.

Vol. 13, No. 1, Maret 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i1.659

[9] Cheriet M., Kharma N., Liu C., Suen C.Y. 2006, Character Recognition System A Guide for Student and Practioners, *John Willey & Sons. Inc*.

[10] Munir, Rinaldi 2004, Pengolahan Citra Digital dengan Pendekatan Algoritmik, *Informatika*, Bandung.

[11] N. Otsu 1979, A threshold selection method from gray level histograms, *IEEE Trans. Syst. Man Cybern*. SMC-9, 62–66.

[12] Putra, Darma 2010, Pengolahan Citra Digital, *Penerbit Andi*, Yogyakarta.

[13] E. Ukkonen, 1985, Algorithms for approximate string matching, *Inf. Control*. 64 (1-3) (1985) 100–118.