

DETEKSI TEKS MENGGUNAKAN *TEXT FLOW* PADA SERTIFIKAT

Indra Rianto¹, Ednawati Rainarli²

^{1,2}Program Studi Teknik Informatika – Universitas Komputer Indonesia,
Jl. Dipatukukur 112-114 Bandung
Email: indrarianto9@gmail.com¹, ednawati.rainarli @email.unikom.ac.id²

ABSTRAK

Penelitian ini memanfaatkan sebuah hasil scan sertifikat untuk mendeteksi teks yang ada didalamnya, dimana pada sebuah sertifikat, terdapat ukuran huruf yang berbeda dan jenis huruf yang ada bukan jenis huruf yang biasa dipakai pada dokumen biasanya melainkan jenis huruf yang cukup mirip dengan tulisan tangan. Hasil pendeteksian teks tersebut dipengaruhi oleh *preprocessing* serta data uji yang digunakan.

Dalam penelitian ini metode pengenalan teks yang digunakan adalah *Text Flow* serta bantuan klasifikasi huruf menggunakan *Tesseract*. Sebelum melakukan proses deteksi, citra tersebut mengalami *preprocessing*. Kemudian sample yang digunakan untuk data uji yaitu tiga buah sertifikat. Berdasarkan hasil pengujian yang dilakukan dengan menggunakan tiga kombinasi, maka didapatkan akurasi terbaik untuk *Recall* sebesar 67% *Precision* sebesar 58% dan *F-Score* sebesar 62% dengan kombinasi yang digunakan adalah nilai *threshold* 128 dengan *min confidence score* 50, jarak *horizontal* kurang dari 2, jarak *vertical* kurang dari 0.2 dan besar antara huruf sebesar kurang dari 2. Hasil penerapan metode *Text Flow* pada sertifikat ini masih kurang dari 70% tetapi mampu untuk tidak memilih objek selain teks seperti tanda tangan dan logo.

Kata kunci: Deteksi teks, Sertifikat, Text Flow, Min-Cost Flow, Segmentasi.

1. PENDAHULUAN

Kebutuhan untuk mendeteksi teks pada sertifikat sangat penting, salah satu faktor yang mendorong kebutuhan untuk mendeteksi teks pada sertifikat adalah kemajuan teknologi. Pada saat ini sertifikat tidak hanya dibutuhkan secara cetak (*hardcopy*), namun dibutuhkan juga file (*softcopy*) dari sertifikat tersebut untuk menunjang beberapa hal seperti kecepatan pengiriman, keamanan dan keutuhan dari info yang terdapat pada sertifikat. Sertifikat mengandung informasi penting yang umumnya terdiri dari kop, nomor surat, nama peserta, waktu dan tempat, serta pengesahan [1]. Penelitian untuk pendeteksian teks pada sertifikat sebelumnya telah dilakukan oleh Reza Yogi Andria yang menyimpulkan bahwa terdapat masalah pada

bagian segmentasi. Penggunaan metode *Connected Component* pada penelitian tersebut mengakibatkan terdeteksinya gambar latar atau background pada bagian sertifikat akan terdeteksi sebagai objek/huruf.

Penelitian tentang Optical Character Recognition (OCR) oleh Raden Sofian Bahri dan Irfan Maliki yang berjudul Perbandingan Algoritma *Template Matching* dan *Feature Extraction* pada *Optical Character Recognition* menyimpulkan bahwa berdasarkan akurasi, pengembangan, dan waktu, algoritma *feature extraction* lebih unggul dibandingkan algoritma *template matching* pada OCR [2].

Segmentasi memiliki proses pendekatan yang berbeda bergantung pada jenis input yang diterima, tiap input memiliki pendekatan yang berbeda pada penerapan proses segmentasi. Meskipun memiliki perbedaan, tujuan yang ingin dicapai adalah sama yaitu identifikasi atau pengenalan. Dalam sertifikat, pengenalan atau identifikasi objek bertujuan untuk mengenali teks yang terdapat pada sebuah sertifikat. Pengenalan teks pada umumnya mempunyai empat langkah berurutan yang terdiri dari deteksi karakter kandidat, penghapusan karakter kandidat tak terpakai, ekstraksi baris teks, dan verifikasi baris teks. Namun, banyak kesalahan terjadi sepanjang masing-masing langkah berurutan ini yang sering menyebabkan kinerja deteksi rendah. Untuk mengatasi masalah ini, sistem deteksi teks terpadu *Text Flow* dapat mengurangi tingkat kesalahan, dengan memanfaatkan *minimum cost (min-cost) flow network model*.

Dengan pendeteksian karakter kandidat oleh *cascade boosting*, model jaringan aliran biaya minimum (*min-cost flow network*) mengintegrasikan tiga langkah terakhir menjadi satu proses yang memecahkan akumulasi kesalahan di kedua level karakter dan tingkat garis teks secara efektif. *Text Flow* telah diuji pada tiga dataset publik, yaitu, dataset ICDAR2011, ICDAR2013 dataset, dataset ini berisikan foto yang mengandung teks dan dataset multibahasa dan itu mengungguli metode terancang pada ketiga dataset dengan *recall* dan *F-score* yang jauh lebih tinggi yaitu metode Baseline, Huang et al, dan Neuman and Matas [3]. Oleh karena itu, pada penelitian ini akan digunakan metode *Text Flow* dengan tujuan mengukur keberhasilan dari metode *textflow* pada pendeteksian karakter.

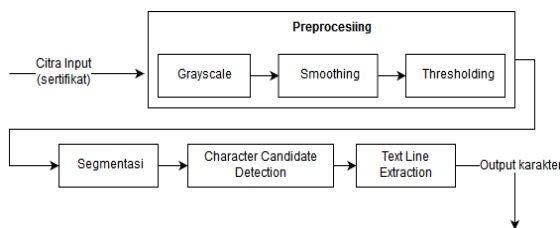
2. ISI PENELITIAN

2.1. Analisis Masalah

Adapun masalahnya yaitu bagaimana cara mendeteksi teks yang ada pada sertifikat karena tidak semua sertifikat menggunakan huruf yang biasa digunakan untuk menulis di suatu dokumen melainkan jenis huruf yang cukup mirip dengan tulisan tangan. Metode yang digunakan untuk klasifikasi pada huruf sertifikat tersebut adalah *Text Flow*. Pada penelitian sebelumnya *Text Flow* sudah diimplementasikan untuk pengenalan teks pada citra hasil foto [3]. *Text Flow* telah diuji pada tiga dataset publik, yaitu, dataset ICDAR2011, ICDAR2013 dataset, dataset ini berisikan foto yang mengandung teks dan dataset multibahasa dan itu menggunakan metode terancang pada ketiga dataset dengan recall dan F-score yang jauh lebih tinggi yaitu metode Baseline, Huang et al, dan Neuman and Matas [3].

2.2. Analisis Sistem

Sistem yang akan dibangun memiliki masukan, proses, dan keluaran. Masukan yang terdiri dari data latih dan data uji. Berikut adalah proses yang terjadi pada sistem dilihat pada gambar 1.



Gambar 1. Block Diagram Proses Utama Sistem

2.3. Analisis Data Masukan

Pada sistem, data masukan tersebut merupakan suatu citra. Data masukan merupakan citra input berupa hasil scan sertifikat yang digunakan sebagai data uji. Berikut ini contoh data masukan pada gambar 2.



Gambar 2. Contoh Data Latih

2.4. Analisis Metode

Pada bagian ini yaitu menjelaskan analisis metode yang terjadi pada implementasi metode *Text Flow* pada sertifikat.

2.4.1. Analisis Preprocessing

Tahapan dari *preprocessing* terdiri dari *grayscale*, *smoothing*, *thresholding* dan segmentasi.

a. Grayscale

Proses grayscale dilakukan untuk mendapatkan citra grayscale atau citra dengan warna keabuan. Rumus yang digunakan adalah "luma" atau "luminance". Berikut adalah rumus *grayscale* pada persamaan 1 dibawah ini.

$$Gray = 0,299 * red + 0,587 * green + 0,114 * blue \quad (1)$$

b. Smoothing

Smoothing dilakukan menggunakan citra integral dengan tujuan untuk memperbaiki kualitas citra dan menghilangkan derau. Pada tahap ini digunakan sebuah mask sebagai rentang perhitungan dengan nilai ketetanggaan [4]. Pembuatan data matriks citra integral $g_{x,y}$ dari citra grayscale $G_{i,j}$ dengan ukuran $M \times N$ menggunakan persamaan 2 dibawah.

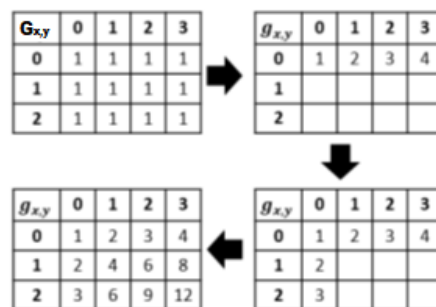
$$g_{x,y} = \sum_{i=0}^x \sum_{j=0}^y G_{i,j}, x = 0,1,2,\dots,M-1, y = 0,1,2,\dots,N-1 \quad (2)$$

Penyederhanaan akan dilakukan pada persamaan 2 untuk komputasi yang lebih efektif. Berikut adalah hasil penyederhanaan dari persamaan 2.

$$g_{0,y} = G_{0,y} + g_{0,y-1} \quad y=1,2, \dots, N-1 \quad (3)$$

$$g_{x,y} = G_{x,0} + g_{x-1,0} \quad x=1,2,\dots,M-1 \quad (4)$$

$$g_{x,y} = G_{x,y} + g_{x,y-1} + g_{x-1,y} - g_{x-1,y-1}, x=1,2,\dots,M-1, y=1,2,\dots,N-1 \quad (5)$$



Gambar 3. Ilustrasi Citra Integral

Menentukan dimensi mask yang akan mempengaruhi jumlah intensitas yang diperoleh dari area mask tersebut. Ukuran mask 3×3 , dengan ukuran ini diperoleh tetangga sebanyak 8 pixel.

Perhitungan nilai rata-rata citra integral pada rentang dimensi mask dengan persamaan 6 dibawah.

$$\begin{aligned} \text{mean}(x,y) &= \frac{s(x,y)}{wx \times wy}, \\ x &= 0,1,2..M-1, \quad y = 0,1,2..N-1 \end{aligned} \quad (6)$$

Dimana $s(x,y)$ dapat dihitung dengan persamaan 7 dibawah.

$$\begin{aligned} s(x,y) &= (g_{x+dx-1,y+dy-1} + g_{x-dx,y-dy}) \\ &\quad - (g_{x-dx,y+dy-1} + g_{x+dx-1,y-dy}) \end{aligned} \quad (7)$$

Dengan $dx = \text{round}(\frac{wx}{2})$, $dy = \text{round}(\frac{wy}{2})$,

pada penelitian ini, diusulkan suatu teknik pencarian cell atau piksel pada mask yang tidak memiliki nilai dengan menambahkan satu fungsi artifisial (x,y) pada persamaan 6 sehingga menjadi seperti pada persamaan 8.

$$\begin{aligned} \text{mean}(x,y) &= \frac{s(x,y)}{wx \times wy} + a(x,y), \\ x &= 0,1,2..M-1, \quad y = 0,1,2..N-1 \end{aligned} \quad (8)$$

Nilai dari fungsi $a(x,y)$ dapat dihitung dengan persamaan 9.

$$a(x,y) = \begin{cases} \frac{nw(x,y)}{wx \times wy}, & \text{if } \left(\begin{array}{l} (x-nr < 0 \text{ or } x+nr \geq M) \\ \text{or } (y-nc < 0 \text{ or } y+nc \geq N) \end{array} \right) \\ 0, & \text{sebaliknya} \end{cases} \quad (9)$$

$$\text{Dengan } nr = \text{trunc}(\frac{wx}{2}) \text{ dan } nc = \text{trunc}(\frac{wy}{2})$$

Dimana nilai fungsi $nw(x,y)$ dapat dihitung dengan persamaan 10 berikut.

$$nw(x,y) = \begin{cases} (rp(x) \times wy + cp(y) \times (wx - rp(x))) \times 255, & \text{if } \left(\begin{array}{l} (x-nr < 0 \text{ and } y-nc < 0) \\ \text{or } (x-nr < 0 \text{ and } y+nc \geq N) \\ \text{or } (x+nr \geq M \text{ and } y-nc < 0) \\ \text{or } (x+nr \geq M \text{ and } y+nc \geq N) \end{array} \right) \\ rp(x) \times wy \times 255, & \text{if } \left(\begin{array}{l} (x-nr < 0 \text{ or } x+nr \geq M) \\ \text{and } (y > 0 \text{ and } y < N) \end{array} \right) \\ cp(y) \times wx \times 255, & \text{if } \left(\begin{array}{l} (y-nc < 0 \text{ or } y+nc \geq N) \\ \text{and } (x > 0 \text{ and } x < M) \end{array} \right) \end{cases} \quad (10)$$

(x) = Row padding, jumlah baris pada mask yang berada di luar dimensi citra dihitung berdasarkan jarak terluar.

(y) = Column padding, jumlah kolom pada mask yang berada di luar dimensi citra dihitung berdasarkan jarak terluar.

Dengan fungsi $rp(x)$, dan $cp(y)$ yang dapat dilihat pada persamaan 11.

$$\begin{aligned} rp(x) &= \begin{cases} |x-nr|, & \text{if } (x-nr < 0) \\ x+nr-M+1, & \text{if } (x+nr \geq M) \end{cases} \\ cp(y) &= \begin{cases} |y-nc|, & \text{if } (y-nc < 0) \\ y+nc-N+1, & \text{if } (y+nc \geq N) \end{cases} \end{aligned} \quad (11)$$

c. Thresholding

Proses thresholding yaitu pengolahan citra *grayscale* menjadi citra biner atau citra hitam putih. Nilai ambang batas yang digunakan pada penelitian ini adalah 128 [5]. Berikut algoritma yang digunakan pada persamaan 12 dibawah ini.

$$\begin{aligned} &\text{if } (\text{gray} > 128) \{ \\ &\quad \text{val} = 0 \\ &\} \text{ else } \{ \\ &\quad \text{val} = 1 \\ &\} \end{aligned} \quad (12)$$

Pertama-tama lakukan pengambilan piksel di mulai dari piksel (0,0). Setelah mengambil piksel maka lakukan perhitungan menggunakan persamaan 2 diatas. Lakukan hingga piksel terakhir pada citra.

d. Segmentasi

Proses segmentasi bertujuan untuk membagi citra ke dalam basis elemen sesuai dengan kriteria yang ditentukan. Metode yang digunakan pada proses segmentasi adalah *Connected Component Labeling* (CCL) [6]. Berikut adalah langkah-langkah dari metode tersebut.

- 1) Definisikan variabel pendukung.
Matriks untuk menyimpan koordinat objek = $pt_{i,j}$
Matriks untuk menyimpan batas koordinat objek = $br_{i,j}$
Array 1D untuk menyimpan nilai koordinat x dan y sementara = c
- 2) Pencarian dimulai dari koordinat (0,0) sampai menemukan matriks bernilai satu (1).
- 3) Simpan koordinat tersebut pada matriks $pt_{i,j}$, lalu ubah nilai matriks bernilai satu(1) dengan nilai nol(0).
- 4) Simpan batas koordinat objek kedalam matriks $br_{0,j}$. karena variabel tersebut saat ini belum memiliki nilai, maka nilai batas atas dan batas bawah akan diisi dengan nilai koordinat x, batas kiri dan batas kanan akan diisi dengan koordinat y.
- 5) Selanjutnya cari nilai piksel matriks tetangga bernilai satu dengan koordinat pusat (3,3) dengan *mask* berdimensi 5 X 5.
- 6) Simpan koordinat piksel bernilai satu (1) kedalam matriks $pt_{i,j}$ lalu inialisasikan nilai pada koordinat matriks satu (1) dengan nilai nol (0).
- 7) Gunakan koordinat dengan nilai satu pada matriks tetangga, inialisasikan koordinat x pada variable C_0 dan koordinat y pada variabel C_1 . Selanjutnya lakukan perbandingan nilai pada matriks $br_{0,j}$ dari data sebelumnya dengan koordinat dengan nilai satu pada matriks tetangga. Proses ini memperoleh nilai batas terluar untuk suatu objek.
- 8) Proses pada langkah nomor 5 sampai 7 akan diulang dengan menggunakan koordinat selanjutnya pada matriks $pt_{0,j}$ hingga tidak ada lagi koordinat yang dapat diproses pada matriks tersebut.

2.4.2. Analisis Min-Cost Flow Network.

Pada tahap ini akan dibahas perancangan, penentuan cost dari setiap calon kandidat karakter dan implementasi dari *Min-Cost Flow Network*. Calon karakter yang terdeteksi oleh proses segmentasi akan diproses oleh tesseract sebelum memasuki tahapan *Min-Cost Flow Network*, hal ini dilakukan untuk mengecek setiap calon karakter, apakah merupakan sebuah karakter atau bukan. Jika hasil dari tesseract menyatakan sebuah calon karakter adalah benar karakter, maka karakter tersebut akan ditandai dengan kotak berwarna hijau, sedangkan jika dianggap bukan karakter maka akan ditandai dengan kotak berwarna merah. Selanjutnya kotak yang berwarna hijau lah yang akan diproses untuk proses *Min-Cost Flow Network*.

1. Sebelum pembentukan teks akan dilakukan dulu pengecekan, apakah citra input memiliki lebih dari satu baris calon teks atau tidak. Pengecekan dilakukan berdasarkan sumbu Y pada karakter hasil segmentasi pada proses sebelumnya. Berikut adalah contoh untuk melakukan pengecekan baris dengan contoh pada Tabel 1 Berikut.

Tabel 1 Calon Karakter Kandidat I sebagai A (kotak merah) dan Calon Karakter Kandidat N sebagai B (kotak kuning)

x/y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
4	0	0	1	1	0	0	0	1	1	1	0	0	1	1	0
5	0	0	1	1	0	0	0	1	1	1	0	0	1	1	0
6	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0
7	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0
8	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0
9	0	0	1	1	0	0	0	1	1	0	0	1	1	1	0
10	0	0	1	1	0	0	0	1	1	0	0	1	1	1	0
11	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Dari tabel dapat dimisalkan bahwa b1 ditandai dengan kotak berwarna merah dan b2 ditandai dengan kotak berwarna kuning.

- a. Keterangan b1

$$b1.y = 3$$

$$b1.h = 9$$

- b. Keterangan b2

$$b2.y = 3$$

$$b2.h = 9$$

Berikut ini adalah fungsi yang digunakan untuk melakukan pengecekan baris.

Box boxA; Box boxB;

if (b1.y > b2.y) {

boxA = b2; boxB = b1;

} else {

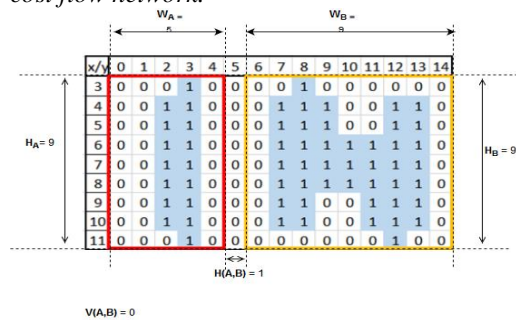
boxA = b1; boxB = b2;

}

return (boxB.y >= boxA.y && boxB.y <= (boxA.y + boxA.h)) || (((boxB.y + boxB.h) >= boxA.y && (boxB.y + boxB.h) <= (boxA.y + boxA.h)));

jika fungsi ini menghasilkan nilai true maka kedua objek yang masuk kedalam fungsi akan dianggap sebagai satu baris, jika tidak maka akan dianggap sebagai baris yang berbeda.

2. Berdasarkan asumsi bahwa semua baris teks dimulai dari kiri ke sebelah kanan, semua kandidat karakter akan disortir terlebih dahulu untuk mendapatkan koordinat horizontal mereka. Kandidat karakter didapatkan dari hasil segmentasi yang dilakukan sebelumnya. Berikut adalah gambar untuk perancangan *min-cost flow network*.



Gambar 4 Ilustrasi Jarak dan Persamaan

Untuk setiap kandidat karakter A, kandidat karakter berikutnya B, yang dapat disambungkan oleh A, harus dibatasi oleh kondisi tertentu untuk mengurangi kesalahan dalam penghitungan jarak antara karakter A dan B. Berikut adalah kondisi yang harus dipenuhi.

Tabel 2 Kondisi Jarak Antara Kandidat Karakter

No	Kondisi	Syarat
1.	Jarak Horizontal antara A dan B	$\frac{H(A,B)}{\min(W_A, W_b)} < T_H$
2.	Jarak vertikal Antara A dan B	$\frac{V(A,B)}{\min(W_A, W_b)} < T_V$
3.	Ukuran dari A dan B	$\frac{ W_A - W_b }{\min(W_A, W_b)} < T_S$

Setelah semua pengecekan dilakukan, dan semua syarat terpenuhi maka dapat diartikan bahwa dua karakter A dan B tersebut adalah tetangga dan dapat diartikan sebuah teks.

2.5. Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai ketepatan atau kecocokkan dari hasil aplikasi pengenalan teks pada sertifikat.

Dalam penelitian ini pengujian akurasi digunakan beberapa kombinasi parameter pada data uji. Tabel dibawah ini merupakan kombinasi parameter yang digunakan.

Tabel 3 Kombinasi Parameter Pengujian

Pengujian Ke-	Min Confidence Score	Nilai T_H , T_V dan T_S
1	65	(2, 0.6, 0.2)
2	60	(2, 0.6, 1)
3	50	(2, 0.6, 2)

2.5.1 Hasil Pengujian Akurasi Kombinasi ke-1

Pengujian akurasi kombinasi ke-1 dilakukan dengan menggunakan kombinasi parameter sebagai berikut.

Nilai Min Confidence Score : 65
 Nilai T_H , T_V dan T_S : 2, 0.6, 0.2

Berdasarkan pengujian yang dilakukan terhadap aplikasi dengan menggunakan parameter diatas, maka tabel 4 dibawah ini merupakan hasil pengujiannya.

Tabel 4. Hasil Pengujian Kombinasi ke-1

No	Nama Citra	Recall	Precision	F-score
1	dataUji1	0	0	0
2	dataUji2	0.09	0.11	0.099
3	dataUji3	0.09	0.11	0.099
Rata-rata		0.06	0.07	0.066

2.5.2 Hasil Pengujian Akurasi Kombinasi ke-2

Pengujian akurasi kombinasi ke-2 dilakukan dengan menggunakan kombinasi parameter sebagai berikut.

Nilai Min Confidence Score : 60
 Nilai T_H , T_V dan T_S : 2, 0.6, 1

Berdasarkan pengujian yang dilakukan terhadap aplikasi dengan menggunakan parameter diatas, maka tabel 5 dibawah ini merupakan hasil pengujiannya.

Tabel 5. Hasil Pengujian Kombinasi ke-2

No	Nama Citra	Recall	Precision	F-score
1	dataUji1	0.125	0.14	0.12
2	dataUji2	0.047	0.11	0.06
3	dataUji3	0.5	0.66	0.56
Rata-rata		0.224	0.3	0.25

2.5.3 Hasil Pengujian Akurasi Kombinasi ke-3

Pengujian akurasi kombinasi ke-3 dilakukan dengan menggunakan kombinasi parameter sebagai berikut.

Nilai Min Confidence Score : 50
 Nilai T_H , T_V dan T_S : 2, 0.6, 2

Berdasarkan pengujian yang dilakukan terhadap aplikasi dengan menggunakan parameter diatas, maka tabel 6 dibawah ini merupakan hasil pengujiannya.

Tabel 6. Hasil Pengujian Kombinasi ke-3

No	Nama Citra	Recall	Precision	F-score
1	dataUji1	0.57	0.57	0.57
2	dataUji2	0.66	0.54	0.59
3	dataUji3	0.77	0.63	0.69
Rata-rata		0.67	0.58	0.62

Pengujian dilakukan satu persatu pada setiap data uji, sehingga didapatkan rata-rata akurasi dari setiap pengujian. Berikut ini merupakan tabel hasil pengujian yang didapatkan.

Tabel 7. Hasil Akurasi Setiap Kombinasi

Pengujian ke-	Nilai Confidence Score, T_H , T_V dan T_S	Recall	Precision	F-score
1	(65, 2, 0.6, 0.2)	0.06	0.07	0.066
2	(60, 2, 0.6, 1)	0.224	0.3	0.25
3	(50, 2, 0.6, 2)	0.67	0.58	0.62

Berdasarkan tabel diatas, Adapun kombinasi parameter terbaik adalah kombinasi parameter ke-3 dengan akurasi rata-rata *Recall* sebesar 67% *Precision* sebesar 58% dan *F-Score* sebesar 62%, Kombinasi parameter yang digunakan yaitu nilai threshold 128 dengan min confidence score 50, jarak horizontal (T_H) 2, jarak vertical (T_V) 0.2 dan besar antara huruf (T_S) sebesar 2. Nilai besar antara huruf (T_S) pada penelitian ini berbeda dengan penelitian yang dilakukan sebelumnya yaitu 0.2 [3], hal ini kemungkinan besar diakibatkan oleh berbedanya jenis huruf yang digunakan dan berbedanya ukuran di sebuah sertifikat.

Hasil temuan yang didapatkan pada saat pengujian adalah terdapat beberapa kekurangan dari metode pengenalan karakter atau huruf. Berikut adalah kekurangan dari metode segmentasi yang diimplementasikan pada sistem ini:

1. Pada penelitian ini seharusnya objek yang dikenali itu hanya huruf dan angka saja, akan tetapi setelah dilakukan pengujian, *noise* yang ada pada sertifikat dikenali sebagai sebuah objek juga.
2. Metode ini tidak dapat menangani tulisan yang berdempet. Karena pada sertifikat, jenis huruf

yang ada terkadang ada huruf yang berdempet dengan huruf lainnya. Hasilnya apabila ada beberapa huruf yang berdempet maka akan dibaca sebagai satu objek/huruf saja.

3. Metode ini tidak dapat menangani sebuah huruf yang memiliki dua objek yang terpisah, contohnya seperti huruf i (I kecil) dan huruf j (J kecil). Oleh karena itu, huruf tersebut tidak dikenali sebagai satu objek melainkan dua objek.

3. PENUTUP

3.1. Kesimpulan

Berdasarkan hasil pengujian sistem Pengenalan Teks Menggunakan Metode *Text Flow* Pada Sertifikat maka diperoleh akurasi terbaik masing-masing untuk *Recall* sebesar 67% *Precision* sebesar 58% dan *F-Score* sebesar 62%. Akurasi ini dipengaruhi oleh *preprocessing*, pengenalan karakter serta data uji yang digunakan. Karena pada setiap sertifikat memiliki ukuran huruf yang berbeda untuk setiap teksnya sehingga mengakibatkan sulitnya menentukan jarak yang tepat untuk membentuk sebuah teks. Pendeteksian karakter sebelum memasuki proses pendeteksian teks juga sangat berpengaruh karena, apabila ada satu huruf yang tidak terdeteksi ditengah-tengah sebuah teks maka teks tersebut akan dianggap menjadi dua bagian teks yang berbeda.

3.2. Saran

Agar penelitian selanjutnya tentang Pengenalan Teks Menggunakan Metode *Text Flow* Pada Sertifikat memiliki akurasi yang lebih tinggi, maka berikut adalah saran yang dapat dijadikan pertimbangan, yaitu memperbaiki metode segmentasi dan pengenalan huruf yang berdempet, serta mengenali sebuah huruf yang memiliki dua objek yang terpisah.

DAFTAR PUSTAKA

- [1] F. A. Prabowo and M. Syani, "Sistem Informasi Pengolahan Sertifikat Berbasis Web di Divisi Training Seamolec," *Jurnal Masyarakat Informatika Indonesia*, vol. 2, no. 1, pp. 73-81, 2017.
- [2] R. S. Bahri and I. Maliki, "Perbandingan Algoritma Template Matching dan Feature Extraction pada Optical Character Recognition.," *KOMPUTA: Jurnal Komputer dan Informatika*, vol. 1, no. I, pp. 29-35, 2012.
- [3] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu and a. C. L. Tan, "Text Flow: A Unified Text Detection System in Natural Scene Images," in *IEEE International Conference on Computer Vision*, 2015.
- [4] T. Singh, S. Roy, O. Singh, T. Sinam and K. Singh, "A New Local Adaptive Thresholding Technique in Binarization," *International Journal of Computer Science*, vol. 8, no. 6, No 2, p. 271, 2011.
- [5] P. Hidayatullah, *Pengolahan Citra dan Aplikasinya*, Bandung: Informatika, 2017.
- [6] K. S. FU and J. K. MUI, "A Survey on Image Segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3-16, 1981.
- [7] A. Bissacco, M. Cummins, Y. Netzer and H. Neven, "Photo to OCR: Reading text in uncontrolled condition.," *International Conference on Computer Vision (ICCV)*, pp. 785-792, 2003.
- [8] R. K. Ahuja, T. L. Magananti and J. B. Orlin, *Network flows: Theory, algorithm, and application*, 1993.
- [9] X. Chen and A. L. Yuille, "Detecting and Reading Text in Natural Scenes," *Computer Vision and Pattern Recognition (CVPR)*, no. 2004, pp. 366-373, 2010.
- [10] Y. Liu, C. Luo and L. Jin, "Tightness-aware Evaluation Protocol for Scene Text Detection," 2019.