ANALISIS SISTEM OPTICAL CHARACTER RECOGNITION (OCR) PADA DOKUMEN DIGITAL MENGGUNAKAN METODE TESSERACT

PERFOMANCE ANALYSIS OF OPTICAL CHARACTER RECOGNITION (OCR) SYSTEM ON DIGITAL DOCUMENTS USING TESSERACT METHOD

Amalia Nur Rahmawati[1], Suryo Adhi Wibowo[2], Unang Sunarya[3]
[1,2]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung
[1]**amalianurr@student.telkomuniversity.ac.id**, [2]**suryoadhiwibowo@telkomuniversity.ac.id**,
[3]**unangsunarya@telkomuniversity.ac.id**

Abstrak

Pada era modern ini, kemampuan teknologi sudah semakin mutakhir. Hampir semua yang dilakukan manusia saat ini menggunakan digital. *Optical Character Recognition* (OCR) merupakan salah satu teknologi yang digunakan untuk mendeteksi karakter pada suatu gambar menjadi bentuk teks yang dapat dibaca oleh mesin komputer. Penelitian OCR sebelumnya meneliti segmentasi dan penerjemah dokumen menggunakan Tesseract OCR. Metode yang akan digunakan dalam penelitian ini adalah dengan metode Tesseract pada dokumen digital karena cocok untuk digunakan pada sistem OCR untuk deteksi karakter pada suatu objek. Sistem dirancang menggunakan *Python*. Pengujian dilakukan pada 10 foto dokumen digital. Konfigurasi sistem uji yang digunakan untuk pengujian adalah konfigurasi sistem terbaik berdasarkan parameter performansi yang digunakan yaitu parameter jarak, rotasi, dan *opacity*. Parameter performansi yang terbaik didapatkan pada jarak 30 cm pada kondisi di luar ruangan sebesar 85,1%, kemudian performansi terbaik rotasi sebesar 85,1%, dan performansi opacity terbaik didapatkan pada jarak 30 cm dengan kondisi di dalam ruangan sebesar 84,5%.

Kata kunci : ***Tesseract, OCR, digital document, image processing***

Abstract

In this modern era, technological capabilities are increasingly up to date. Almost everything that humans do today uses digital. Optical Character Recognition (OCR) is one of the technologies used to detect characters in an image into a form of text that can be read by a computer machine. Previous OCR research examined segmentation and document translators using the Tesseract OCR. The method to be used in this study is by tesseract method on digital documents because it is suitable for use on OCR systems for character detection on an object. The system is designed using Python. Testing was conducted on 10 photos of digital documents. The test system configuration used for testing is the best system configuration based on the performance parameters used i.e. distance, rotation, and opacity parameters. The best performance parameters are obtained at a distance of 30 cm in outdoor conditions of 85.1%, then the best performance rotation is 85.1%, and the best opacity performance is obtained at a distance of 30 cm with indoor conditions of 84.5%

Keywords: ***Tesseract, OCR, digital document, image processing***

1. Pendahuluan

Optical Character Recognition (OCR) merupakan suatu alat yang dapat digunakan untuk mengkonversi cetakan karakter menjadi teks digital, tanpa harus mengetik ulang[1]. Dengan menggunakan OCR ini, gambar yang bertulis tangan, mesin ketik, dapat di cari per kata atau kalimat yang dapat diganti atau dimanipulasi dan diberikan barcode[1]. Dalam mengubah gambar menjadi text,langkah-langkah yang dilakukan oleh algoritma OCR, yaitu: akuisisi gambar, pra-pemrosesan,

segmentasi, ekstraksi fitur, klasifikasi, dan pasca-pemrosesan[2]. Sistem OCR dapat digunakan dalam berbagai aplikasi praktis seperti pengenalan plat nomor, perpustakaan pintar, pengenalan karakter berbagai macam bahasa[2]. OCR merupakan sub bidang dari *Pattern Recognition* (PR) yang berkaitan dengan pengenalan karakter[3].

Permasalahan dalam OCR sering dijumpai karena operasi aritmatika OCR yang belum sepenuhnya terselesaikan, dan layanan *cloud* OCR yang masih buruk kinerjanya pada saat memproses gambar dengan operasi aritmatika tulisan tangan. Dari studi literatur, peneliti Khawaja Ubaid Ur Rehman dan Yaser Daanial Khan meneliti skrip teks Urdu menggunakan teknik OCR dan klasifikasi dengan metode *Cascade Forward Backpropagation Neural Network* melalui pengujian kumpulan data dan lima kali *cross*-validasi yang memberikan tingkat akurasi 96,444% dan 96,922%[3]. Peneliti Wanwei Wang dkk meneliti *License Plate Recognition*(LPR) dalam *Intelligent Transportation System* (ITS) untuk mendeteksi plat nomor dengan menggunakan *Multi-task Convolutional Neural Network* (MTCNN) yang dapat meningkatkan ketepatan dan kecepatan deteksi dengan mengadopsi model *Convolutional Recurrent Neural Network* (CRNN) dan *Connectionist Temporal Classification* (CTC) untuk pengenalan plat lisensi dan menghasilkan tingkat akurasi 98% [4]. Peneliti Yue Yin dkk menggunakan teknik OCR untuk meneliti karakter huruf besar China dengan tujuan meningkatkan akurasi dan mengurangi waktu pemrosesan pada aplikasi *Internet of Things* (IoT) dengan menguji *Convolutional Neural Network* (CNN), *Visual Geometry Group* (VGG), *Capsule Network* (CapsNet) dengan akurasi 99,17%, *Residual Netwrk* (ResNet) dengan akurasi 99,38% dan mengembangkan metode CNN yang menghasilkan akurasi beban jaringan turun 96,5% dengan akurasi 97,70% dan waktu pengujian setiap jaringan rata-rata 20 kali CPU[5]. Peneliti Sahil Thakare, Ajay Kamble dkk mengembangkan sebuah metode yang menggabungkan segmentasi dan translasi bahasa untuk membuat sebuah dokumen dari bahasa tertentu mudah dimengerti oleh pembaca yang tidak fasih dalam bahasa tersebut. Mereka menggunakan OCR untuk mengkonversi dokumen yang digunakan menjadi karakter-karakter yang kemudian ditranslasi ke bahasa tertentu menggunakan API milik Google Translate[6]. Tesseract ini masih mempunyai kelemahan pada penggunaan *Polygonal Approximation* sebagai input ke classifier, akurasi mungkin dapat ditingkatkan secara signifikan dengan penambahan dari model n-gram karakter *Hidden-Markov-Model-based* [7].

Dikarenakan metode Tesseract masih mempunyai kelemahan, maka dalam Tugas Akhir ini mengajukan teknik OCR dengan menggunakan metode *Tesseract* pada dokumen digital yang didukung oleh sistem Android untuk mengidentifikasi berbagai objek seperti huruf, angka, teks dari kamera *smartphone*. Tesseract merupakan salah satu mesin OCR bersifat *open source* yang merupakan fitur analisis pengenalan teks yang fleksibel, dan masih dalam pengembangan aktif oleh Google dan saat ini tersedia dalam versi 4.0[8]. Parameter yang menunjukkan keberhasilan penelitian ini dengan menggunakan metode Tesseract pada dokumen digital ini adalah tingkat akurasi ketepatan yang tinggi, yang kemudian akan di analisis. Perbedaan penelitian yang akan dilakukan ini mengusulkan sistem OCR dengan menggunakan metode Tesseract pada dokumen digital. Sedangkan pada ref[5] tidak dijelaskan metode yang digunakan dan berisi hasil pengujian dari beberapa jenis klasifikasi untuk OCR berbasis IoT.

2. Dasar Teori

2.1 Optical Character Recognition

Optical Character Recognition (OCR) merupakan suatu alat yang digunakan untuk mengkonversi karakter huruf, angka, simbol dan karakter pada suatu gambar dalam berbagai bahasa yang digunakan agar dapat dibaca oleh mesin. Masukkan pada OCR umumnya berupa tulisan tangan atau karakter dari cetakan mesin. Pengenalan karakter tulisan tangan sulit diidentifikasi karena perbedaan tekanan pada karakter yang sama oleh pengguna. Desain sistem OCR terdiri dari beberapa bagian yaitu:

1. *Grayscale*

Grayscale merupakan intensitas setiap piksel, untuk konversi warna berdasarkan model warna RGB menjadi representasi *grayscale* menggunakan persamaan 2.1 [9].

$$Y = 0.2126R + 0.7152G + 0.0722 \quad (2.1)$$

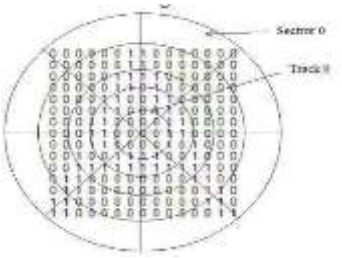
2. *Feature Extraction*

Feature Extraction[10] merupakan proses dalam mendapatkan informasi sebuah atau sekelompok objek dalam proses klasifikasi. Dokumen masukan yang berisi beberapa teks dipisah menjadi karakter tunggal untuk *recognition*[9]. *Feature Extraction* terdiri dari dua bagian yaitu *Convolutional Layer* dan *Pooling Layer*.

3. *Recognition of Pattern*

Recognition of Pattern merupakan hasil dari pencocokan dari format biner pada sebuah karakter. Biner dibagi menjadi beberapa trek yang setiap trek akan dibagi menjadi beberapa sektor. Untuk mengetahui radius dengan menggunakan piksel menggunakan jarak maksimal pusat dengan persamaan[9] 2.2 dan pembagian sektor dan trek pada gambar 2.1.

$$D = \sqrt{((y2 - y1)^2 + (x2 - x1)^2)} \quad (2.2)$$



Gambar 1. Pembagian Trek dan Sektor

4. *Recognition of Output*

Recognition of Output merupakan proses pencocokan template yang terdiri dari setiap nilai sektor dan setiap nilai trek dengan sektor trek yang dihasilkan. Jika semua parameter yang ditemukan cocok dengan nilai template, maka hasilnya adalah karakter yang teridentifikasi.

2.2 Tesseract

Tesseract[11] merupakan salah satu sistem OCR yang bersifat *open source* yang telah diteliti dan digunakan dalam berbagai skrip teks dan bahasa. Tesseract melakukan analisis karakter dengan memperhatikan posisi objek, huruf kecil, huruf besar, ligature, simbol dan karakter yang terhubung maupun yang berubah bentuk karena mengikuti posisi sebuah kata. Tesseract adalah mesin OCR yang mempunyai kemampuan untuk mengenali lebih dari 100 bahasa.

Tesseract kini mempunyai versi terbaru yang telah diimplementasikan sepenuhnya dalam bahasa pemrograman C++ dan mendukung platform Linux dan Windows[8]. Metode umum yang digunakan untuk mengklasifikasi piksel vertikal, masing-masing mempunyai parsial karakter dan menggabungkan klasifikasi model Hidden-Markov yang membentuk batas karakter[12].

2.2.1 Prinsip Kerja Tesseract

Prinsip kerja Tesseract sebenarnya tidak jauh berbeda dengan prinsip kerja dari metode OCR yang lainnya. Tesseract yang digunakan OCR mempertimbangkan data pelatihan yang dapat diperoleh dari beberapa langkah yang menghasilkan file perantaran, yang kemudian dikemas ke hasil tunggal. Nama file umumnya menunjukkan bahasa dan kemungkinan font yang digunakan[8].

2.2.2 Sistem Pelatihan Tesseract

a. *Shape Training*

Metode Tesseract dalam melakukan pengenalan karakter menggunakan pendekatan fitur ekstraksi dengan pengelompokan berdasarkan gambar karakter dan pembuatan set karakter (alfabet).

b. *Dictionary Training*

Tesseract menggunakan format khusus, yaitu: *Directed Acyclic Word Graph* (DAWG) yang dibuat daftar kata dalam format teks, dan kebalikannya yang daftar kata diekstraksi dari file DAWG yang berguna untuk memperbarui kamus yang sudah ada. Berbagai jenis kamus yang digunakan Tesseract membantu berbagai kombinasi karakter yang mungkin digunakan.

2.2.3 Tahapan Utama OCR di Tesseract

Beberapa tahapan utama yang digunakan OCR di Tesseract sebagai berikut:

1. *Line and Word Finding*

Tahap pertama pada OCR adalah mencari garis teks menggunakan algoritma. Algoritma ini akan mencari objek serupa yang berada di dalam suatu garis. Setelah garis teks ditemukan, selanjutnya akan dibuat garis batas yang menentukan garis batas atas, garis tengah, dan garis bawah dari objek yang sudah diidentifikasi.

2. *Word Recognition*

Salah satu bagian proses dari mesin pengenal karakter adalah untuk mengidentifikasi bagaimana kata-kata bisa disegmentasi ke karakter. Jika ada karakter-karakter yang tergabung karena adanya kerning, maka karakter-karakter tersebut dipotong terlebih dahulu dengan teknik *chopping*. Hasil cetak yang terpisah-pisah digabungkan dengan teknik *associating broken characters*.

3. *Static Character Classifier*

Pada tahap ini, Tesseract menggunakan *machine learning* untuk mengidentifikasi huruf. Huruf mempunyai banyak variasi dari segi font, ukuran, dan attributes. Belum lagi ditambah kualitas gambar yang berbeda-beda.

4. *Linguistic Analysis*

Kapan saja modul pengenalan kata menemukan segmentasi baru, modul linguistik akan memilih string kata yang terbaik melalui beberapa kategori. Segmentasi yang akan dipilih adalah yang mempunyai kata dengan *total rating distance* yang paling rendah, dengan kategori yang telah dikalkulasi dengan konstan tertentu.

5. *Adaptive Classifier*

Adaptive classifier lebih sensitif untuk *font*. Tesseract tidak memiliki template untuk classifier, tetapi menggunakan *feature* dan *classifier* yang sama dengan *static classifier*. Teknik ini memberi kemudahan dalam membedakan huruf besar dan kecil, juga meningkatkan immunity untuk *noise specs*.

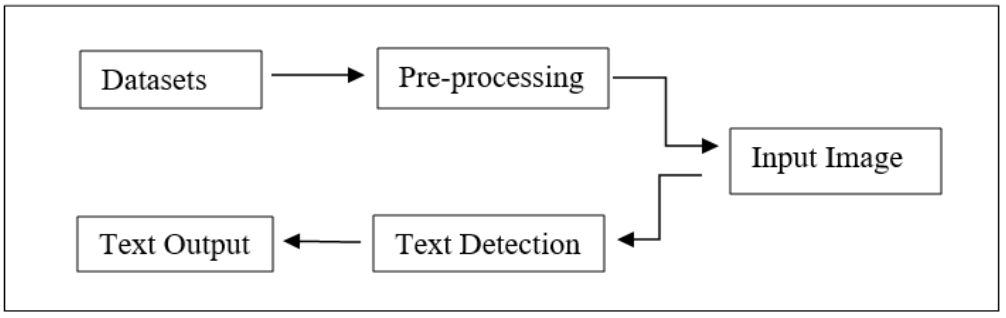
2.3 Bahasa Pemrograman Python

Bahasa pemrograman yang digunakan pada Tugas Akhir ini adalah Python. Bahasa pemrograman Python merupakan bahasa yang mendukung *multi- paradigma* yang mendukung kode program prosedural dan pemrograman object (OOP) yang dilengkapi dengan *library* yang kompeherensif, sehingga dapat digunakan untuk membuat aplikasi seperti *big data*, *data mining*, *deep learning*, *data science*, dan *machine learning*. Sintaks Python sederhana dan mudah untuk dipelajari.

3. Pembahasan

3.1 Gambaran Umum Sistem

Tugas Akhir ini mempunyai gambaran umum sistem OCR yang akan dirancang dan diimplementasikan sebagai berikut.



Gambar 2. Gambaran Umum Sistem OCR

Berdasarkan Gambar 2 dapat dilihat bahwa:

1. *Datasets*: mengumpulkan foto dokumen digital yang akan di deteksi.
2. *Image Pre-Processing*: pada tahap ini, gambar asli akan dilakukan koreksi gambar mengenai karakter yang miring maupun sejenisnya.
3. *Input Image*: memasukkan gambar yang akan dideteksi.
4. *Text Detection*: menemukan area teks pada gambar yang dideteksi.
5. *Text Output*: mengidentifikasi teks yang dideteksi.

3.2 Desain Sistem

Sistem pada Tugas Akhir ini menggunakan teknologi pengolahan citra digital (*digital image processing*) untuk mendeteksi tulisan dengan masukan berupa citra yang didapatkan dari foto dan keluaran berupa teks yang dapat dibaca oleh mesin komputer atau *smartphone*. Sistem deteksi objek ini menggunakan metode Tesseract.

Deteksi pada sistem ini mempunyai beberapa tahapan, yaitu: akuisisi gambar atau masukan citra digital, *pre-processing*, segmentasi, ekstraksi, klasifikasi, pasca pemrosesan. Tahapan tersebut mempunyai fungsi masing-masing dapat ditunjukan pada diagram alir 3 sebagai berikut.

Gambar 3. Diagram Alir Sistem OCR

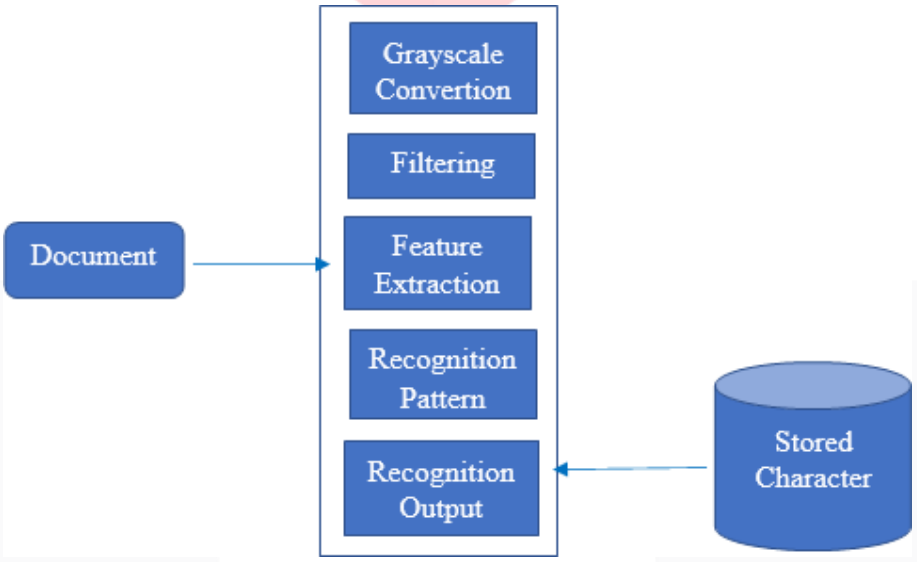
Berdasarkan pada Gambar 3 dapat dilihat bahwa:

1. Akuisisi gambar merupakan data masukan berupa citra digital. Pada tahap ini, terjadi proses kuantisasi yaitu mengubah gambar menjadi bentuk biner dan dikompresi.
2. *Pre-processing* berfungsi untuk meningkatkan kualitas gambar yang dideteksi dan manghaluskan karakter digital [16].
3. Segmentasi berfungsi untuk memisahkan gambar menjadi unsur-unsur karakter.
4. Ekstraksi berfungsi untuk mengekstraksi fitur-fitur karakter pada gambar, baik fitur geometris maupun fitur statistik yang akan mengurangi dimensi gambar.

3.2.1 Desain Sistem OCR

Di dalam sistem OCR, terdapat beberapa tahapan yang ditunjukkan pada Gambar 4 untuk mengkonversi gambar yang dideteksi sebagai berikut:

1. *Grayscale Conversion*: hasil pengukuran setiap piksel.
2. *Filtering*: memfilter gambar dalam bentuk piksel yang baik.
3. *Feature Extraction*: informasi dari suatu atau sekelompok objek.
4. *Recognition Pattern:* matriks yang dihasilkan berisi nilai unik untuk setiap font karakter sehingga mudah untuk diidentifikasi masing-masing.
5. *Recognition Output*: sektor dan trek yang sesuai dengan matriks yang dihasilkan kemudian mengidentifikasi jumlah piksel dalam setiap daerah. Hal-hal tersebut dapat ditunjukkan pada gambar 4 sebagai berikut.



Gambar 4. Gambar Ulang Diagram Blok Sistem OCR

3.3 Parameter Perfomansi

Parameter performansi yang akan diujikan pada Tugas Akhir ini, sebagai berikut:

1. Akurasi

Akurasi merupakan tolak ukur ketepatan gambar yang dideteksi. Akurasi direpresentasikan dengan N yang merupakan banyaknya data dengan T yang merupakan banyaknya pengujian. Maka dari itu, akurasi dijabarkan dengan persamaan 3.1.

$$A = \frac{N}{T} \; x \; 100 \tag{3.1}$$

4. Hasil dan Analisis

4.1 Skenario Pengujian

Skenario ini bertujuan untuk menguji fungsionalitas sistem apakah dengan menggunakan metode Tesseract ini mendapatkan hasil OCR yang akurat. Skenario pengujian ini dilakukan dengan membuat dataset berupa foto 10 dokumen, dengan setiap dokumen diambil pada jarak 30 cm, 40 cm, dan 50 cm dengan keadaan *indoor* dan *outdoor*. Sehingga skenario pengujian ini mempunyai dataset total 60 gambar.

1. Skenario 1: Pengujian terhadap parameter Jarak

Pengujian parameter jarak bertujuan untuk mengetahui performansi keseluruhan sistem yang sudah dirancang seberapa akurat dalam melakukan deteksi teks terhadap *dataset* yang sudah disediakan.

Pada tahap ini, sistem yang akan diuji untuk melakukan pengujian pada parmeter jarak 30 cm, 40 cm, dan 50 cm dengan *dataset* sebanyak 60 foto dokumen dengan keadaan *indoor* dan *outdoor*.

2. Skenario 2: Pengujian terhadap parameter Rotasi

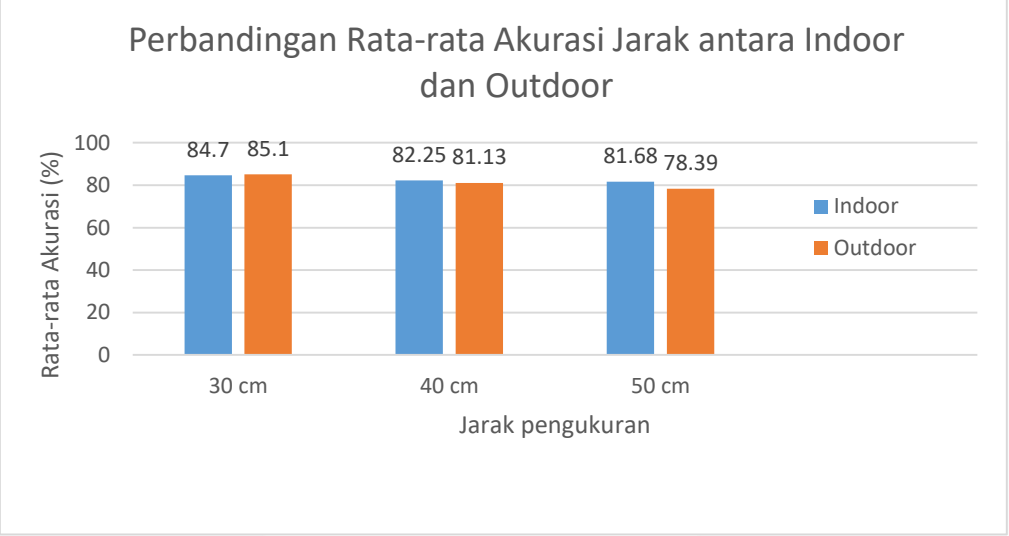
Pengujian parameter rotasi bertujuan untuk mengetahui performansi keseluruhan sistem yang sudah dirancang seberapa akurat dalam melakukan deteksi teks terhadap *dataset* yang sudah disediakan. Pada tahap ini, sistem yang akan diuji untuk melakukan pengujian dengan *dataset* sebanyak 30 foto diambil dari performansi jarak yang terbaik yaitu pada keadaan 30 cm *outdoor*.

3. Skenario 3: Pengujian terhadap parameter *Opacity*

Pengujian parameter opacity bertujuan untuk mengetahui performansi transparansi gambar secara keseluruhan sistem yang sudah dirancang seberapa akurat dalam melakukan deteksi teks terhadap *dataset* yang sudah disediakan. Pada tahap ini, sistem yang akan diuji untuk melakukan pengujian dengan *dataset* sebanyak 30 foto diambil dari performansi jarak yang terbaik yaitu pada keadaan 30 cm *outdoor*.

4.2 Hasil Pengujian

a. Parameter Jarak



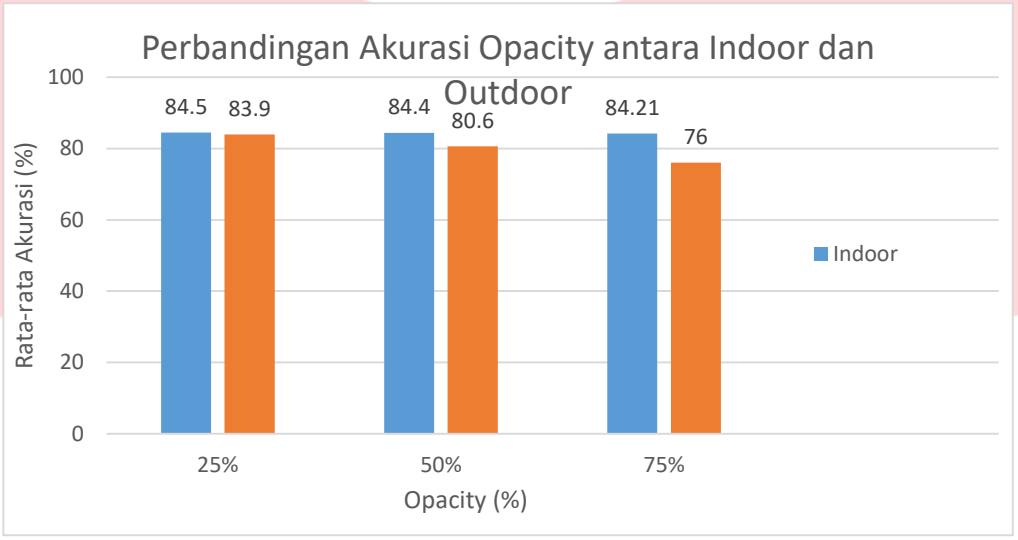
Gambar 5. Perbandingan Rata-rata akurasi jarak antara Indoor dan Outdoor

b. Parameter Rotasi

Akurasi terbaik pada parameter jarak yaitu pada saat 30 cm *outdoor* yang kemudian di rotasi untuk menghitung parameter rotasi dengan 90˚, 180˚, dan 270˚ tidak berpengaruh pada deteksi teks yang dihasilkan, untuk hasil indoor sebesar 84,7% dan outdoor sebesar 85,1%.

1. Indoor
   Rata-Rata Dokumen 90deg: 0.8470640770297264
   Rata-Rata Dokumen 180deg: 0.8470640770297264
   Rata-Rata Dokumen 270deg: 0.8470640770297264
2. Outdoor
   Rata-Rata Dokumen 90deg: 0.8510699486750924
   Rata-Rata Dokumen 180deg: 0.8510699486750924
   Rata-Rata Dokumen 270deg: 0.8510699486750924

c. Parameter Opacity



Gambar 6. Perbandingan Akurasi Opacity Indoor dan Outdoor

4.3 Analisis

Pada pengujian yang telah dilakukan, hasil performa Tesseract dari setiap parameter dinilai cukup baik. Hal ini dibuktikan dari rata-rata akurasi pada setiap parameter. Pada parameter jarak, rata-rata akurasi terbaik didapatkan ketika pada keadaan *outdoor* dengan jarak sebesar 30 cm yaitu sebesar 85,1%. Sementara rata-rata akurasi terendah didapatkan ketika pada jarak 50 cm dengan keadaan *outdoor* yaitu sebesar 78,39%. Semakin jauh gambar maka semakin berpengaruh pada akurasinya. Hal ini dipengaruhi kemiringan baris, spasi pada baris yang baru, dan cahaya yang ada pada foto.

Kemudian untuk parameter rotasi tidak ditemukan perbedaan pada rata-rata akurasi dari setiap parameter. Rata-rata akurasi terbaik didapatkan pada keadaan *outdoor* yaitu sebesar 85,1%. Sementara rata-rata akurasi terendah didapatkan pada keadaan *indoor* yaitu sebesar 84,7%.

Pada parameter opacity mendapatkan rata-rata akurasi terbaik pada saat keadaan *indoor* dengan transparansi 25% yaitu sebesar 84.5%, sedangkan rata-rata akurasi terendah didapatkan pada keadaan *outdoor* dengan transparansi 75% yaitu sebesar 76%.

5. Kesimpulan

Setelah dilakukan pengujian, performansi sistem OCR pada dokumen digital menggunakan metode Tesseract didapatkan kesimpulan sebagai berikut:

1. Rata-rata akurasi terbaik berdasarkan parameter jarak didapatkan dengan jarak 30 cm dan dengan kondisi outdoor sebesar 85,1%.
2. Rata-rata akurasi terendah didapatkan pada jarak 50 cm dengan keadaan outdoor yaitu sebesar 78,39%.
3. Rotasi tidak berpengaruh terhadap akurasinya. Rata-rata akurasi terbaik berdasarkan parameter rotasi didapatkan pada kondisi outdoor 30 cm sebesar 85,1%
4. Rata-rata akurasi terbaik berdasarkan parameter opacity didapatkan pada transparansi 25% dengan kondisi *indoor* yaitu sebesar 84.5%.
5. Rata-rata akurasi terendah berdasarkan parameter opacity didapatkan pada transparansi 75% dengan kondisi outdoor yaitu sebesar 76%

6. Referensi

[1] E. S. Pusat Penelitian dan Pengembangan Telekomunikasi and Pusat Penelitian Elektronika dan Telekomunikasi (Indonesia), "Jurnal elektronika dan telekomunikasi.," *J. Elektron. dan Telekomun.*, vol. 17, no. 2, pp. 56–62, Dec. 2017.

[2] N. Islam, Z. Islam, and N. Noor, "A Survey on Optical Character Recognition System," 2016.

[3] K. U. U. Rehman and Y. D. Khan, "A Scale and Rotation Invariant Urdu Nastalique Ligature Recognition Using Cascade Forward Backpropagation Neural Network," *IEEE Access*, vol. 7, pp. 120648–120669, Aug. 2019, doi: 10.1109/access.2019.2936363.

[4] W. Wang, J. Yang, M. Chen, and P. Wang, "A Light CNN for End-to-End Car License Plates Detection and Recognition," *IEEE Access*, vol. 7, pp. 173875–173883, 2019, doi: 10.1109/ACCESS.2019.2956357.

[5] Y. Yin, W. Zhang, S. Hong, J. Yang, J. Xiong, and G. Gui, "Deep Learning-Aided OCR Techniques for Chinese Uppercase Characters in the Application of Internet of Things," *IEEE Access*, vol. 7, pp. 47043–47049, 2019, doi: 10.1109/ACCESS.2019.2909401.

[6] S. Thakare, A. Kamble, V. Thengne, and U. R. Kamble, "Document Segmentation and Language Translation Using Tesseract-OCR," *2018 13th Int. Conf. Ind. Inf. Syst. ICIIS 2018 - Proc.*, pp. 148–151, Jul. 2018, doi: 10.1109/ICIINFS.2018.8721372.

[7] R. Smith, "An Overview of the Tesseract OCR Engine." Accessed: Feb. 10, 2020. [Online]. Available: http://code.google.com/p/tesseract-ocr.

[8] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "Efficient and effective OCR engine training," *Int. J. Doc. Anal. Recognit.*, vol. 23, no. 1, pp. 73–88, Oct. 2019, doi: 10.1007/s10032-019-00347-8.

[9] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical Character Recognition Implementation Using Pattern Matching," *Int. J. Comput. Sci. Inf. Technol.*, vol. Vol. 5 (2), pp. 2088–2090, 2014, Accessed: Apr. 07, 2020. [Online]. Available: www.ijcsit.com.

[10] N. Mani and B. Srinivasan, "Application of artificial neural network model for optical character recognition," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1997, vol. 3, pp. 2517–2520, doi: 10.1109/icsmc.1997.635312.

[11] "GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository)." https://github.com/tesseract-ocr/tesseract (accessed Feb. 23, 2020).

[12] L. D. . Smith R, Antonova D, "Adapting the Tesseract open source OCR engine for multilingual OCR," *ACM International Conference Proceeding Series*, 2009. https://dl.acm.org/doi/pdf/10.1145/1577802.1577804?accessTab=true (accessed Feb. 25, 2020).

[13] "What is Python? Executive Summary | Python.org." https://www.python.org/doc/essays/blurb/ (accessed Apr. 06, 2020).