

PERBANDINGAN PYTESSERACT DAN TEMPLATE MATCHING UNTUK OTOMATISASI INPUT DATA KTP

Teresa Octaviani¹, Hendry Setiawan², Oesman Hendra Kelana³

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Ma Chung

Villa Puncak Tidar Blok N No. 1, Kab. Malang 65151, Jawa Timur, Indonesia

Email: ¹teresaocta04@gmail.com, ²hendry.setiawan@machung.ac.id, ³oesman.hendra@machung.ac.id

Abstract. *Perbandingan Pytesseract dan Template Matching untuk Otomatisasi Input KTP. ID Cards serve several purposes, including identification, registration, management, and accessing public services. Until now, ID Cards registration has been done by typing so it takes more time and effort, and often causes errors in typing so that the data stored does not match. Therefore, an application was developed with OCR-based automation for inputting ID card information. The method used for OCR is determined by testing the accuracy of Pytesseract and Template Matching in condition using a smartphone camera with dark, bright, and very bright lighting, and using a laptop camera. The average of the accuracy of the four conditions obtained by Pytesseract is 98.33%, while the average obtained by Template Matching is 67.33%. Based on these results, the OCR system developed using Pytesseract.*

Keywords: *Optical Character Recognition (OCR), Pytesseract, Template Matching*

Abstrak. *KTP memiliki banyak fungsi, seperti sebagai kartu identitas, dalam proses pendaftaran, dalam proses kepengurusan, serta untuk mengakses layanan. Hingga saat ini, pendaftaran KTP dilakukan dengan diketik sehingga tidak hemat waktu dan tenaga, serta sering menyebabkan kesalahan dalam pengetikan sehingga data yang disimpan tidak sesuai. Oleh karena itu, dikembangkan aplikasi dengan fitur otomatisasi pengisian data KTP menggunakan OCR. Metode OCR yang akan digunakan ditentukan dengan pengujian akurasi metode Pytesseract dan template matching pada kondisi menggunakan kamera smartphone dengan pencahayaan gelap, terang, terang sekali, dan menggunakan kamera laptop. Rata-rata tingkat akurasi dari empat pengondisian yang didapatkan oleh metode Pytesseract adalah 98,33%, sedangkan rata-rata yang didapatkan oleh metode template matching adalah 67,33%. Berdasarkan hasil ini, sistem OCR yang dikembangkan menggunakan metode Pytesseract.*

Kata Kunci: *Optical Character Recognition (OCR), Pytesseract, Template Matching*

1. Pendahuluan

Bagi warga negara Indonesia, KTP dianggap sebagai kartu identitas dan tanda pengenal. KTP diperlukan dalam berbagai proses pendaftaran, seperti pendaftaran CPNS dan pendaftaran calon anggota TNI/Polri. KTP juga diperlukan ketika ingin mengakses layanan publik, program pemerintah, dan bantuan sosial. Hingga saat ini, pendaftaran KTP dilakukan dengan mengisi data KTP dengan cara diketik satu per satu. Hal ini membuat sistem pendaftaran menjadi lebih memakan waktu dan tenaga. Cara ini juga rawan terhadap kesalahan dalam pengetikan yang menyebabkan ketidaksesuaian pada data yang disimpan. Hal ini menyebabkan ketidakefektifan dalam proses kerja. Oleh karena itu, program ini dibuat untuk membantu proses pendaftaran KTP dengan mengembangkan otomatisasi input data KTP. Otomatisasi input KTP dilakukan dengan membaca KTP menggunakan OCR. Proses otomatisasi ini dapat membantu menghemat waktu dan tenaga dan meminimalisir terjadinya kesalahan pengetikan.

OCR dapat membantu mengenali karakter pada gambar. Penelitian OCR menggunakan *Pytesseract* dan *template matching* sudah pernah dilakukan sebelumnya. Menurut Diwanti, OCR menggunakan *Pytesseract* berhasil memberikan akurasi 80% dari 25 plat untuk pembacaan plat secara *real time* menggunakan CCTV [1]. Menurut Kurniawan, OCR menggunakan *Template Matching* berhasil memberikan akurasi 80,25% dari 30 plat [2]. Berdasarkan dua penelitian ini

dicari metode terbaik, antara *Pytesseract* dan *Template Matching*, untuk sistem OCR yang akan dikembangkan. Oleh karena itu, penelitian ini dilakukan untuk menentukan metode terbaik untuk dijadikan metode pembacaan OCR pada sistem yang akan dikembangkan.

2. Tinjauan Pustaka

2.1. Kartu Tanda Penduduk Elektronik (E-KTP)

Dalam e-KTP terdapat sistem keamanan yang didasarkan pada basis data kependudukan nasional. Setiap warga negara Indonesia hanya akan memiliki satu e-KTP dan satu NIK seumur hidupnya [3]. Dalam e-KTP juga terdapat *chip* untuk membantu proses autentikasi keaslian KTP. *Chip* ini sendiri dapat menjadi *tag* yang dibaca menggunakan RFID dengan frekuensi 13.56 MHz. UID pada e-KTP terdiri dari 4 bit dalam kombinasi 10 digit [4]. Meskipun begitu, beberapa sistem tidak mendukung untuk proses pembacaan *chip* secara langsung, seperti perpustakaan, sistem pendaftaran acara kecil, dan pendaftaran *online*.

2.2. Optical Character Recognition (OCR)

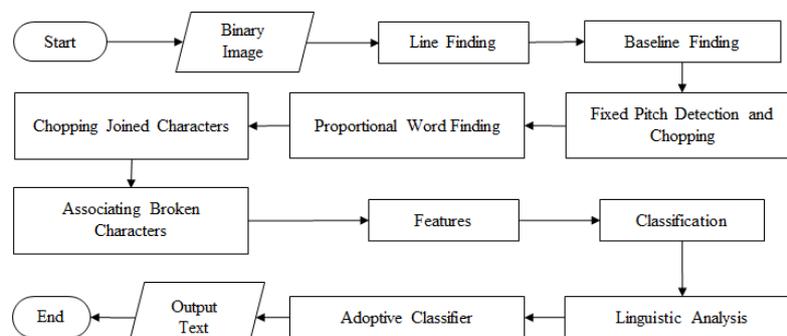
Optical Character Recognition atau yang biasa dikenal sebagai OCR merupakan metode yang dapat mengenali huruf atau karakter pada gambar untuk dikonversi menjadi karakter yang dikenali komputer [5]. Sebelum proses OCR, biasanya gambar diubah menjadi *grayscale* terlebih dahulu. Hal ini dapat membantu meningkatkan akurasi pengenalan OCR. Kemudian dilakukan *filtering* dan *feature extraction* sebelum proses pengenalan [6]. Gambar yang dikenali oleh OCR bisa didapat dengan memfoto teks secara langsung maupun memindai teks menjadi gambar digital, atau melalui gambar digital langsung.

OCR mengubah cara untuk berinteraksi dengan teks. Dengan mengkombinasikan kecerdasan buatan dan pemrosesan gambar yang canggih, OCR menjadi jembatan antara dunia fisik dan dunia digital. OCR dapat memberikan peluang besar dalam mengoptimalkan efisiensi bisnis, meningkatkan akurasi pengolahan data, dan memperluas aksesibilitas informasi. OCR dapat mengurangi ketergantungan pada proses manual yang lambat dan berpotensi menimbulkan kesalahan. Dengan OCR dapat mempercepat alur kerja, meningkatkan produktivitas karyawan, dan menghemat waktu serta sumber daya.

OCR dapat dilakukan menggunakan berbagai metode, seperti *Pytesseract* dan *Template Matching*. Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Oleh karena itu, penelitian ini dilakukan untuk menentukan metode terbaik antara metode *Pytesseract* dan *Template Matching* untuk digunakan dalam proses pembacaan E-KTP.

2.3. Pytesseract

Pytesseract adalah alat yang dimiliki *Python* untuk membantu proses pengenalan karakter atau *Optical Character Recognition* (OCR). *Pytesseract* adalah alat yang membungkus mesin Tesseract-OCR dari Google. Banyak format gambar yang dapat dibaca oleh *Pytesseract*, contohnya *jpeg* dan *png*. *Pytesseract* umumnya dimanfaatkan untuk membantu membaca teks pada gambar digital [7]. Contoh fungsi dalam *library Pytesseract* yang sering digunakan adalah *image_to_string*.



Gambar 1. *Flowchart Pytesseract*

Secara keseluruhan, algoritma *Pytesseract* dapat dilihat pada Gambar 1. Gambar yang dimasukkan dalam proses *Pytesseract* berupa gambar biner. Pada gambar tersebut kemudian dilakukan *line finding* untuk mencari baris pada gambar. Setelah baris ditemukan dilakukan proses *baseline fitting* untuk mencari garis dasar pada setiap baris dengan memasang *quadratic spline* yang membagi karakter menjadi tiga kelompok. Selanjutnya dilakukan proses *pitch detection and chopping* untuk memotong karakter berdasarkan pola yang sudah diuji. Berikutnya adalah proses *proportional word finding* untuk menentukan spasi atau jarak antar kata pada teks yang memiliki pola tidak tetap. Kemudian dilakukan proses *chopping joined characters* untuk memotong karakter yang saling terhubung.

Tahapan selanjutnya adalah *associating broken characters* untuk menghubungkan karakter yang terpisah dengan bantuan asosiator yang menggunakan algoritma A*. Setelah karakter terpisah-pisah dilakukan proses *features* dengan mengekstraksi karakter menjadi 50-100 fitur dan mencocokkannya dengan fitur prototipe dari data training yang diekstraksi menjadi 10-20 fitur. Berikutnya dilakukan tahapan *classification* untuk mengelompokkan karakter berdasarkan vektor prototipe kelas yang sesuai. Selanjutnya akan ada proses *linguistic analysis* untuk mengklasifikasikan kata sesuai dengan yang ada pada modul linguistik. Pengklasifikasian dalam modul linguistik didasarkan pada data training untuk tiap bahasa yang sudah tersedia pada *Pytesseract*. Bahasa untuk modul linguistik ini dapat ditentukan dengan parameter “lang” yang diinputkan ketika memanggil fungsi. *Pytesseract* akan mencocokkan gambar dengan data training yang sesuai dengan bahasa dari nilai parameter lang. Tahapan terakhir adalah *adaptive classifier* yang menggunakan *x-height normalization*. Tahapan *adaptive classifier* lebih peka terhadap font, dapat membedakan antara huruf besar dan huruf kecil, serta memiliki ketahanan yang lebih baik terhadap noise [8].

Dengan *Pytesseract*, OCR dapat diimplementasikan dengan mudah pada program yang menggunakan bahasa pemrograman *Python*. *Pytesseract* dapat memberikan hasil pembacaan dengan presisi yang tinggi. Di dalam *Pytesseract* sudah terdapat proses pengolahan gambar, seperti ekstraksi fitur dan segmentasi teks, yang dapat mempermudah penggunaannya dalam mengolah gambar yang akan dibaca dan meningkatkan akurasi pembacaan. Selain itu, *Pytesseract* memiliki modul linguistik yang dapat membantu mengklasifikasikan kata yang dibaca sesuai data training pada bahasa yang dipilih. *Pytesseract* yang bersifat *open source* mempermudah pengembang dari seluruh dunia untuk berkontribusi dalam pengembangannya sehingga dapat diperbarui secara teratur dan menciptakan inovasi tak terbatas. *Pytesseract* memiliki dokumentasi yang jelas sehingga mudah dipahami. *Pytesseract* sendiri sudah banyak digunakan untuk pengenalan karakter, seperti untuk membaca plat nomor dengan rata-rata akurasi 80% [1], mengenali tulisan tangan pada gambar dengan akurasi sekitar 90% [9], dan berhasil membantu mengidentifikasi obat [10].

2.4. Template Matching

Template matching merupakan salah satu metode yang dapat dimanfaatkan untuk mengenali karakter atau *Optical Character Recognition* (OCR). Prinsip dari *template matching* adalah untuk mencari dan mencocokkan pola pada gambar dengan *template* yang sudah disiapkan [11]. Oleh karena itu, pada metode *template matching* disiapkan karakter *template* yang sesuai terlebih dahulu. Pada gambar yang ingin dikenali kemudian dilakukan *image processing* agar pola pada gambar dapat semakin menyerupai *template*. *Image processing* yang dilakukan seperti mengatur kecerahan pada gambar, mengubah gambar menjadi biner, segmentasi gambar, dan mengatur ukuran gambar. *Template matching* mencocokkan setiap *pixel* dalam suatu gambar dengan *template* yang sudah disiapkan. Penentuan kecocokan gambar ini menggunakan nilai kesamaan dengan mencari jumlah selisih *pixel* terkecil [12].

Template matching cukup fleksibel karena pengguna dapat mengatur *template* sesuai yang diinginkan. *Template* yang dicari harus menyerupai pola yang ingin dikenali. Semakin serupa pola karakter dan *template*, semakin besar akurasi pengenalan pada metode *template matching*. *Template matching* dapat digunakan untuk mengenali teks maupun objek lainnya.

Contoh penggunaan *Template Matching* seperti untuk mengenali plat nomor dengan akurasi 80,25% [2], nomor ruangan dengan akurasi 81,81% [11], dan lain-lain.

2.6. Run-Length Smearing Algorithm (RLSA)

Run-Length Smearing Algorithm (RLSA) adalah metode yang digunakan untuk mencari letak teks pada gambar. Prinsip kerja dari RLSA adalah dengan memindai baris pada gambar baik secara horizontal maupun vertikal. Proses memindai ini disebut sebagai *smearing*. Proses *smearing* pada RLSA dilakukan sebanyak tiga kali. Proses *smearing* pertama dilakukan untuk mencari jarak setiap karakter yang diwakili oleh *pixel* berwarna hitam. Proses *smearing* kedua dilakukan untuk mencari lebar karakter yang diwakili oleh *pixel* berwarna putih. Proses *smearing* ketiga dilakukan untuk mencari besar lokasi teks secara utuh yang diwakili oleh *pixel* berwarna hitam. Kemudian semua *pixel* berwarna putih digabungkan menjadi suatu area yang terdiri dari dua kata atau lebih [13].

2.7. Levenshtein Distance

Levenshtein distance adalah algoritma untuk menghitung perbedaan antara dua *string*. Perbedaan antar *string* dihitung dengan mencari langkah minimum yang dilakukan untuk mengubah suatu *string* menjadi *string* yang lain. Dalam algoritma ini terdapat tiga operasi utama yaitu operasi penyisipan karakter (*insertion*) dengan memasukkan suatu karakter dalam *string*, penghapusan karakter (*deletion*) dengan menghapus suatu karakter pada *string*, dan penukaran karakter (*substitution*) dengan mengganti suatu karakter menjadi karakter lainnya [14]. Algoritma *levenshtein distance* dapat digunakan sebagai mekanisme *autocorrect*. Untuk mekanisme *autocorrect* diperlukan kata-kata yang akan dijadikan *template* atau acuan *string*. Dengan mencari pasangan *string input-template* yang memiliki selisih terkecil, *template* yang ditetapkan menjadi *output* untuk menggantikan *input* yang tidak tepat.

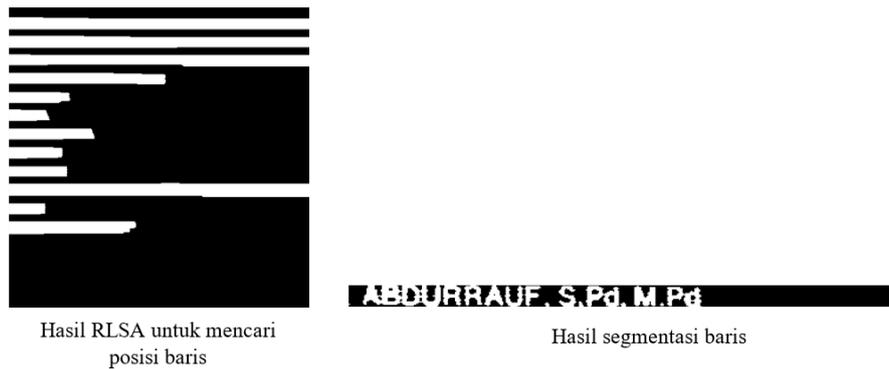
3. Metodologi

Proses perancangan aplikasi dimulai dengan merancang metode *Optical Character Recognition* (OCR) yang akan digunakan. Kemudian proses dilanjutkan dengan proses perancangan aplikasi dan fitur-fitur yang akan dikembangkan. Selain itu, dipilih juga gambar KTP yang akan ditampilkan untuk uji coba. Gambar KTP yang digunakan diambil dari *website gurusiana.id* [15]. Pada proses perancangan ini dirancang juga proses pengujian yang akan dilakukan.

3.1 Perancangan Metode OCR

Perancangan metode OCR dilakukan dengan merancang metode *Pytesseract* dan *Template Matching* yang akan dikembangkan. Dari hasil pembacaan menggunakan *Pytesseract* dan *Template Matching* ini nantinya akan dicari hasil akurasi terbaik. Metode dengan hasil terbaik akan dijadikan metode pembacaan OCR dalam sistem. Setelah metode terbaik ditentukan akan ditambahkan algoritma *levenshtein distance* untuk komponen KTP gender, agama, status perkawinan, dan kewarganegaraan untuk membantu proses *autocorrect* sehingga dapat meminimalisir kesalahan pembacaan.

Baik pada metode *Pytesseract* maupun *template matching*, gambar KTP akan dibagi menjadi gambar NIK dan gambar data pribadi selain NIK. Ukuran pemotongan gambar dibagi menjadi dua yaitu ketika gambar KTP dimasukkan dengan cara *upload* dan ketika difoto langsung menggunakan kamera laptop. Gambar KTP yang diunggah dianggap sudah pas. Untuk gambar yang diunggah, ukuran pemotongan gambar NIK adalah 14-23% secara vertikal dan 20-70% secara horizontal, sedangkan ukuran pemotongan untuk gambar data pribadi lainnya adalah 22-100% secara vertikal dan 23-70% secara horizontal. Untuk gambar yang diambil menggunakan kamera laptop, ukuran pemotongan gambar NIK adalah 21-30% secara vertikal dan 20-70% secara horizontal, sedangkan ukura pemotongan gambar pribadi lainnya adalah 29-100% secara vertikal dan 22-70% secara horizontal. Contoh hasil pemotongan KTP dapat dilihat pada Gambar 2.



Gambar 5. Contoh Hasil RLSA untuk Pencarian Posisi Baris dan Hasil Segmentasi Baris



Gambar 6. Contoh Hasil RLSA untuk Pencarian Posisi Kata dan Hasil Segmentasi Kata

3.2. Perancangan Aplikasi

Perancangan aplikasi terdiri dari perancangan fitur-fitur yang akan dikembangkan dan perancangan tampilan aplikasi. Perancangan ini terdiri dari perancangan *use case diagram* untuk menentukan fitur yang akan dikembangkan, perancangan *flowchart* untuk menentukan alur aplikasi, perancangan *Entity Relationship Diagram* (ERD) untuk merancang *database*, dan perancangan *mockup* untuk membantu menentukan gambaran dari tampilan aplikasi yang akan dikembangkan. Perancangan ERD terdiri dari *Conceptual Data Model* (CDM) untuk membantu menentukan relasi antar tabel dan *Physical Data Model* (PDM) untuk membantu menentukan atribut, tipe atribut, dan panjang atribut, serta atribut yang menjadi *foreign key* (FK). Fitur-fitur yang akan dikembangkan pada aplikasi ini adalah pembacaan KTP (foto langsung maupun *upload file*) menggunakan OCR.

3.3. Pengujian

Pengujian yang dilakukan terdiri pengujian metode OCR dan pengujian aplikasi. Pengujian metode OCR terdiri dari pengujian *Pytesseract* dan pengujian *Template Matching*. Pengujian metode dilakukan terpisah untuk mengetahui metode dengan akurasi tertinggi untuk dimasukkan ke dalam sistem pada aplikasi.

Pengujian metode dilakukan pada empat kondisi yaitu menggunakan kamera *smartphone* dengan pencahayaan gelap (<100 LUX), terang (100-200 LUX), dan terang sekali (>200 LUX), serta menggunakan kamera laptop. Pengkondisian pencahayaan ini dilakukan untuk menguji proses pembacaan dan mengetahui kondisi pencahayaan terbaik sebagai perbandingan terhadap metode *Pytesseract* dan *Template Matching*. Untuk itu diambil 20 KTP dalam kondisi baik sehingga mendapatkan 40 data untuk masing-masing metode. Pertama, akurasi pengujian dihitung untuk masing-masing komponen menggunakan rumus 1. Pada rumus ini, jumlah karakter yang terbaca benar akan dibagi dengan jumlah karakter pada komponen kartu tanda Penduduk (KTP), kemudian dikalikan 100%. Jika total jumlah data yang terbaca lebih dari jumlah karakter pada komponen KTP, jumlah karakter yang terbaca akan dikurangi. Setelah semua komponen KTP dihitung akurasinya, dicari akurasi rata-rata untuk setiap KTP menggunakan rumus 2 [16].

$$\text{Keakuratan (\%)} = \frac{\text{jumlah karakter terbaca benar}}{\text{jumlah karakter pada KTP}} \times 100\% \quad (1)$$

$$\text{Rata-rata (\%)} = \frac{\text{total keakuratan dari masing-masing komponen}}{\text{jumlah komponen pada KTP}} \quad (2)$$

4. Hasil dan Diskusi

4.1. Hasil Pengujian Metode

Hasil pengujian 20 KTP dengan tingkat kecerahan berbeda menggunakan metode *Pytesseract* dapat dilihat pada Tabel 1, sedangkan hasil pengujian 20 KTP dengan tingkat kecerahan berbeda menggunakan metode *template matching* dapat dilihat pada Tabel 2. Gambar yang digunakan untuk uji coba pada kondisi pencahayaan diambil menggunakan kamera *smartphone* dengan pencahayaan 10 LUX untuk kondisi gelap, 122 LUX untuk kondisi terang, dan 390 LUX untuk kondisi terang sekali, sedangkan gambar untuk uji coba menggunakan kamera laptop diambil menggunakan pencahayaan 59 LUX. Pengecekan pencahayaan dilakukan menggunakan aplikasi “Pengukur Cahaya” dari *Coolexp* pada *smartphone*.

Tabel 1. Hasil Pengujian

	Rata-Rata Hasil Akurasi <i>Template Matching</i> (%)	Rata-Rata Hasil Akurasi <i>Pytesseract</i> (%)	Rata-Rata Hasil Akurasi <i>Pytesseract</i> dengan Penambahan Mekanisme <i>Autocorrect</i> (%)	Rata-Rata Waktu Pembacaan Metode <i>Template Matching</i> (detik)	Rata-Rata Waktu Pembacaan Metode <i>Pytesseract</i> (detik)
Gelap (<100 LUX)	63,88	98,30	98,36	2,99	1,80
Terang (100 – 200 LUX)	73,57	98,47	98,53	2,94	1,73
Terang Sekali (200 > LUX)	77,25	99,19	99,19	2,98	1,67
Kamera Laptop (59 LUX)	56,31	97,97	98,03	2,96	1,56
Rata-Rata	67,75	98,48	98,53	2,97	1,69

Dari hasil pada Tabel 1 dapat dilihat hasil pengujian antara metode *Pytesseract* dan *template matching*. Rata-rata tingkat akurasi dari keempat pengkondisian yang didapat *Pytesseract* adalah 98,48%, sedangkan rata-rata tingkat akurasi dari keempat pengkondisian yang didapat *Template Matching* adalah 67,75%. Rata-rata ini menunjukkan bahwa akurasi *Pytesseract* lebih baik dibandingkan akurasi *template matching* sehingga pengembangan metode OCR untuk KTP yang digunakan adalah metode *Pytesseract*.



Gambar 7. Contoh Karakter yang Masih Menyatu

Rata-rata hasil akurasi *template matching* lebih rendah dibandingkan rata-rata hasil akurasi *Pytesseract*. Hal ini berkaitan dengan prinsip kerja *template matching* yang benar-benar harus menyerupai *template*, baik dari segi kecerahan, bentuk, maupun ukurannya. Hal ini sudah diminimalisir dengan mengubah karakter dari E-KTP dan karakter *template* menjadi *grayscale*, mencari *template* dengan bentuk karakter atau font yang serupa dengan font karakter pada E-KTP, serta melakukan segmentasi karakter, *morphological operation*, dan *resize* karakter E-KTP sesuai dengan karakter *template*. Kesalahan pembacaan pada metode *template matching* sebagian

besar disebabkan oleh proses segmentasi karakter yang kurang sempurna. Masih terdapat beberapa karakter yang menyatu atau terpotong, seperti pada Gambar 7. Karakter yang menyatu ini menyebabkan proses pencocokan dengan karakter *template* menjadi tidak sesuai.

Pada *Pytesseract* kemudian ditambahkan algoritma *levenshtein distance* untuk komponen KTP gender, agama, status perkawinan, dan kewarganegaraan. Algoritma *levenshtein distance* ini digunakan untuk membantu proses *autocorrect* sehingga dapat meningkatkan akurasi pembacaan OCR. Dalam Tabel 1 dapat dilihat rata-rata hasil pengujian metode *Pytesseract* yang sudah ditambahkan mekanisme *autocorrect* menggunakan metode *levenshtein distance* pada komponen gender, agama, status perkawinan, dan kewarganegaraan. Berdasarkan hasil tersebut, rata-rata hasil akurasi yang meningkat setelah ditambahkan mekanisme *autocorrect* menggunakan metode *levenshtein distance*. Peningkatan ini dapat dilihat di rata-rata akurasi pada pengkondisian menggunakan kamera *smartphone* dengan pencahayaan gelap (<100 LUX) dari 98,30% menjadi 98,36%, rata-rata akurasi pada pengkondisian menggunakan kamera *smartphone* dengan pencahayaan terang (100-200 LUX) dari 98,47% menjadi 98,53%, dan rata-rata akurasi pada pengkondisian menggunakan kamera laptop dari 97,97% menjadi 98,03%. Walaupun tidak terdapat perbedaan yang signifikan, penambahan mekanisme *autocorrect* ini dapat meningkatkan akurasi pembacaan dari OCR.

Dalam Tabel 1 juga dapat dilihat rata-rata waktu yang dibutuhkan untuk proses pembacaan setiap metode pada setiap kondisi. Berdasarkan hasil tersebut, rata-rata waktu pembacaan yang dibutuhkan oleh *Pytesseract* adalah 1,69 detik. Hasil ini lebih cepat jika dibandingkan dengan rata-rata hasil pembacaan oleh *template matching* yaitu 2,97 detik. Hal ini dapat diakibatkan karena pada metode *template matching*, program akan mengolah *template* terlebih dahulu sebelum mengolah E-KTP dan melakukan proses pencocokan. Semakin banyak *template* yang diolah, semakin lama proses pembacaan *template matching*.

4.2. Hasil Pengujian Aplikasi

Aplikasi dikembangkan menggunakan *framework* Angular untuk bagian *frontend* dan *framework* Django untuk bagian *backend*. *Database* yang digunakan adalah MariaDB yang diakses menggunakan XAMPP. Aplikasi yang dikembangkan terdiri dari dua menu yaitu *Home* dan *Customer*. Menu *Home* hanya berfungsi sebagai tampilan awal ketika menjalankan aplikasi pertama kali. Untuk menambahkan data, *user* dapat masuk ke menu *Customer*.

Penambahan data dapat dilakukan secara otomatis menggunakan OCR. Pada *form customer* mode *Add* terdapat tiga tombol yaitu tombol *Camera* untuk menyalakan kamera dan mengambil foto KTP secara langsung, tombol *Choose File* untuk *upload* KTP, dan tombol *Upload* untuk memanggil API dan menjalankan OCR. Untuk fitur pengambilan gambar KTP secara langsung sudah disediakan *frame* pada tampilan kamera agar dapat membantu *user* menyesuaikan dengan ukuran pemotongan yang sudah ditentukan dalam program. Gambar 8 adalah contoh hasil pembacaan KTP menggunakan OCR dengan cara *upload*. Pada cara *upload* ini, KTP yang di-*upload* dalam kondisi baik dan ukurannya sudah sesuai sehingga tidak memerlukan pemotongan lagi. Hasil OCR akan ditampilkan pada *form* dan *user* dapat mengeceknya terlebih dahulu. Jika data yang ingin disimpan sudah sesuai, *user* dapat menekan tombol *Save* untuk menyimpan data.

Pada Gambar 8, jika dibandingkan antara data pada *form* dengan KTP yang di-*upload*, terdapat beberapa komponen KTP yang tidak sesuai. Contoh komponen KTP yang tidak sesuai seperti pada komponen nama dan alamat. Hal ini sesuai dengan hasil pengujian *Pytesseract* pada Tabel 1 yang belum bisa memberikan akurasi 100%. Oleh karena itu, *user* dapat mengecek kembali hasil pembacaan yang sudah dimunculkan pada *form*.

HOME CUSTOMER

CAMERA CHOOSE FILE UPLOAD



KTP5.JPEG(186.63 KB)



CUSTOMER MAIN INFORMATION

NAME	ABDURRAUF SPD MPD	GENDER	MALE
BIRTH PLACE	CELLENGENGE	BIRTH DATE	25/10/1972
NIK	7312042510720002	ID EXPIRED DATE	DD/MM/YYYY
RELIGION	ISLAM	MARITAL STATUS	KAWIN
NATIONALITY	WNI	PROFESSION NAME	PEGAWAI NEGERI SIPIL (PNS)

ADDRESS INFORMATION

ADDRESS	J MERDEKANO 43	PROVINSI	SULAWESI SELATAN
RT / RW	001 004	KOTA	KAB. SOPPENG
		KECAMATAN	LALABATA
		KELURAHAN	BILA

CANCEL SAVE

Gambar 8. Form customer dan hasil pembacaan KTP menggunakan OCR

5. Kesimpulan dan Saran

Aplikasi berbasis *website* yang dikembangkan dapat melakukan input KTP menggunakan OCR. Metode OCR yang digunakan adalah *Pytesseract*. Fitur OCR ini dapat memberikan rata-rata hasil akurasi 98,33%. Fitur *autocorrect* yang dikembangkan menggunakan algoritma *levenshtein distance* juga dapat meningkatkan tingkat akurasi pembacaan dari rata-rata tingkat akurasi 98,33% menjadi 98,54% walaupun tidak memberikan hasil yang signifikan karena pembacaan oleh *Pytesseract* sudah memberikan tingkat akurasi yang tinggi pada komponen KTP gender, agama, status perkawinan, dan kewarganegaraan. Meskipun begitu, dengan adanya mekanisme *autocorrect* ini dapat lebih menjamin keakuratan pembacaan pada komponen KTP gender, agama, status perkawinan, dan kewarganegaraan.

Saran yang diberikan untuk pengembangan aplikasi selanjutnya adalah untuk lebih meningkatkan akurasi pembacaan OCR, baik menggunakan *Pytesseract*, *Template Matching*, atau metode lainnya. Selain itu, diharapkan nantinya ditambahkan fitur untuk membaca data lain seperti kartu keluarga dan tanda pengenal lainnya.

Referensi

- [1] H. Diwanti, S. Sumaryo, C. Setianingsih. "Real Time Smart CCTV Untuk Mendeteksi Plat Nomor Kendaraan Menggunakan Optical Character Recognition," *e-Proceeding of Engineering*, vol. 6(2), pp. 3045-3052, Agustus, 2019.
- [2] B. S. Kurniawan, S. R. Sentinuwo, O. A. Lantang. "Aplikasi Pengenal Citra Nomor Kendaraan Bermotor Menggunakan Metode Template Matching," *Jurnal Teknik Informatika*, vol. 8(1), pp. 7-12, Mei, 2016.

- [3] M. P. Febriharini. "Pelaksanaan Program e KTP Dalam Rangka Tertib Administrasi Kependudukan," *Serat Acitya*, vol. 5(2), pp. 17-30, 2016.
- [4] T. Puasandi. "Sistem Akses Kontrol Kunci Elektrik Menggunakan Pembacaan E-KTP," *Jurnal Mahasiswa TEUB*, Januari, 2014.
- [5] N. V. Rao, A. S. C. S. Sastry, A. S. N. Chakravarthy, K. P. "Optical Character Recognition Technique Algorithms," *Journal of Theoretical & Applied Information Technology*, vol. 83(2), pp. 275-282, Januari, 2016.
- [6] F. Mohammad, J. Anarase, M. Shingote, P. Ghanwat. "Optical Character Recognition Implementation Using Pattern Matching," *International Journal of Computer Science and Information Technologies*, vol. 5(2), pp. 2088-2090, 2014.
- [7] R. M. I. Indrakusuma, A. S. Ahmadiyah, N. F. Ariyani. "Pengenalan dan Klasifikasi Tulisan pada Nota Pembelian Material (Studi Kasus Proyek Konstruksi)," *Jurnal Teknik ITS*, vol. 10(2), pp. A478-A483, Agustus, 2021.
- [8] M. Hasnat, M. Chowdhury and M. Khan, "An OpenSource Tesseract Based Optical Character Recognizer for Bangla Script", *Ieeexplore.ieee.org*, 2021. [Online]
- [9] S. Salar, S. Roy, S. Verma. "Automate Identification and Recognition of Handwritten Text from an Image," *Turkish Journal of Computer and Mathematics Education*, vol. 12(3), pp. 3800-3808, 2021.
- [10] C. K. Gomathy, V. Geetha, G. S. V. P. Praneetha, M. S. Sucharitha. "Medicine Identification Using Open CV," *Journal of Pharmaceutical Negative Results*, vol. 13, pp. 3718-3723, 2022.
- [11] S. Muharom. "Pengenalan Nomor Ruangan Menggunakan Kamera Berbasis OCR dan Template Matching," *Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 4(1), pp. 27-32, Januari, 2019.
- [12] S. Widodo dan Gunawan. "Template Matching pada Citra E-KTP Indonesia," *Prosiding SNATIKA*, vol. 3, pp. 30-35, November, 2015.
- [13] E. D. Putra, M. Utami, M. H. Rifqo. "Identifikasi Text Meteran Air Menggunakan Metode Run-Length Smearing Algorithm (Rlsa)," *Jurnal Ilmu Komputer dan Informatika*, vol. 4(2), pp. 98-106, Desember, 2021.
- [14] R. Adawiyah dan N. E. Saragih. "Implementasi Algoritma Levenshtein Distance Dalam Mendeteksi Plagiarisme," *Journal Computer Science and Information Technology (JCOinT)*, vol. 5(1), pp. 54-63, 2022.
- [15] A. R. Hanif, E. Nasrullah, and F. X. A. Setyawan, "DETEKSI KARAKTER PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION (OCR)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 1, Jan. 2023, doi: 10.23960/jitet.v11i1.2897.
- [16] M. Rizal Toha and A. Triayudi, "Penerapan Membaca Tulisan di dalam Gambar Menggunakan Metode OCR Berbasis Website pada e-KTP," *Jurnal Sains dan Teknologi*, vol. 11, pp. 175–183, 2022, doi: 10.23887/jst-undiksha.v11i1.