

**OTOMATISASI INPUT DATA KARTU TANDA PENDUDUK (KTP)
MENGGUNAKAN METODE *OPTICAL CHARACTER RECOGNITION*
(OCR) DAN AUTENTIKASI MENGGUNAKAN RFID**

TUGAS AKHIR



TERESA OCTAVIANI

311910013

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2023**

LEMBAR PENGESAHAN
TUGAS AKHIR

**OTOMATISASI INPUT DATA KARTU TANDA PENDUDUK (KTP)
MENGGUNAKAN METODE *OPTICAL CHARACTER RECOGNITION*
(OCR) DAN AUTENTIKASI MENGGUNAKAN RFID**

Oleh:

**TERESA OCTAVIANI
NIM. 311910013**

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS dan TEKNOLOGI
UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom.)

Dosen Pembimbing I,

Hendry Setiawan, ST., M.Kom.

NIP. 20100006

Dosen Pembimbing II,

Ir. Oesman Hendra Kelana, M.Div.,

M.Cs.

NIP. 20110022

Dekan Fakultas Sains dan Teknologi,



Dr. Kestrina Rega Prilianti, M.Si.

NIP. 20120035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan “Otomatisasi Input Data Kartu Tanda Penduduk (KTP) Menggunakan Metode *Optical Character Recognition* (OCR) dan Autentikasi Menggunakan RFID” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 20 Juli 2023



Teresa Octaviani

NIM. 311910013

OTOMATISASI INPUT DATA KARTU TANDA PENDUDUK (KTP) MENGGUNAKAN METODE *OPTICAL CHARACTER RECOGNITION* (OCR) DAN AUTENTIKASI MENGGUNAKAN RFID

Teresa Octaviani, Hendry Setiawan, Oesman Hendra Kelana

Abstrak

Kartu tanda penduduk (KTP) wajib dimiliki oleh masyarakat Indonesia yang sudah berusia 17 tahun ke atas. KTP berfungsi sebagai kartu identitas, tanda pengenal, dalam proses pendaftaran maupun dalam mengakses layanan publik. Selama ini, data-data dari KTP yang didaftarkan diketik secara manual sehingga kurang efektif karena lebih memakan waktu dan tenaga. Selain itu, banyak tersebarnya KTP palsu menyebabkan sulitnya mengetahui keaslian KTP. Ketidakefektifan ini dapat diatasi dengan mengembangkan aplikasi yang memiliki fitur otomatisasi pengisian data dan fitur autentikasi data.

Proses otomatisasi dilakukan menggunakan *Optical Character Recognition* (OCR). OCR berfungsi untuk mengenali huruf atau karakter pada gambar. Metode yang digunakan untuk OCR ditentukan dengan menguji akurasi metode *Pytesseract* dan metode *Template Matching* pada empat kondisi, yaitu menggunakan kamera *smartphone* dengan pencahayaan gelap, terang, terang sekali, dan menggunakan kamera laptop. Selain proses otomatisasi, aplikasi yang dikembangkan memiliki fitur autentikasi untuk menentukan keaslian KTP. Proses autentikasi dilakukan dengan membaca uid KTP menggunakan *Radio Frequency Identification* (RFID).

Rata-rata tingkat akurasi dari empat pengkondisian yang didapatkan oleh metode *Pytesseract* adalah 98,33%, sedangkan rata-rata tingkat akurasi dari empat pengkondisian yang didapatkan oleh metode *Template Matching* adalah 67,33%. Berdasarkan hasil ini, sistem OCR yang dikembangkan menggunakan metode *Pytesseract*. Fitur autentikasi yang dikembangkan juga berhasil membaca uid KTP.

Kata Kunci : *Optical Character Recognition* (OCR), *Pytesseract*, *Radio Frequency Identification* (RFID), *Template Matching*

ID CARD INPUT AUTOMATION USING OPTICAL CHARACTER RECOGNITION (OCR) AND AUTHENTICATION USING RFID

Teresa Octaviani, Hendry Setiawan, Oesman Hendra Kelana

Abstract

Id Card must be owned by Indonesians aged 17 or above. Id Card is used an identity card, identification, in the registration process, and for accessing public services. So far, the registered data from the Id Card is typed so it is ineffective and takes more time and effort. In addition, the spread of the fake Id Cards makes it difficult to know the authenticity of the Id Card. The ineffectiveness can be overcome by developing applications that have data entry automation feature and data authentication feature.

The automation process is carried out using Optical Character Recognition (OCR). OCR is used for recognizing letters or characters in images. The method used for OCR is determined by testing the accuracy of Pytesseract and Template Matching in four conditions, such as using a smartphone camera with dark, bright, and very bright lightning, and using laptop camera. In addition to the automation process, the applications has an authentication feature to determine the authenticity of the Id Card. The authentication process is carried out by reading the uid of the Id Card using Radio Frequency Identification (RFID).

The average of the accuracy of the four conditions obtained by Pytesseract is 98.33%, while the average of the accuracy of the four conditions obtained by Template Matching is 67.33%. Based on these result, the OCR system developed using Pytesseract. The authentication feature that was developed also managed to read the uid of the Id Card.

Keywords : *Optical Character Recognition (OCR), Pytesseract, Radio Frequency Identification (RFID), Template Matching*

KATA PENGANTAR

Segala puji dan syukur dipanjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan hidayah-Nya, sehingga Laporan Tugas Akhir ini dapat diselesaikan dengan sebaik-baiknya. Laporan ini menjelaskan mengenai hasil proyek Tugas Akhir yang sudah dikerjakan.

Pada kesempatan ini, banyak ucapan terima kasih diucapkan kepada pihak-pihak yang telah membantu selama proses penyelesaian proyek Tugas Akhir, di antaranya :

1. Ibu Dr. Kestrilia Rega Prilianti, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Ma Chung,
2. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung sekaligus Dosen Pembimbing I Tugas Akhir,
3. Bapak Ir. Oesman Hendra Kelana, M.Div, M.Cs selaku Dosen Pembimbing II Tugas Akhir,
4. Bapak Windra Swastika, S.Kom, MT, Ph.D selaku Dosen Pengaji Tugas Akhir,
5. Kedua orang tua dan teman-teman yang telah memberikan dukungan dan semangat selama mengerjakan proyek Tugas Akhir

Semoga laporan ini dapat memberikan gambaran mengenai hasil proyek Tugas Akhir yang sudah dikembangkan. Demikian yang dapat disampaikan, saya ucapan terima kasih.

Malang, 20 Juli 2023

Teresa Octaviani

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Perumusan Masalah	3
1.5 Tujuan	3
1.6 Manfaat	3
1.7 Luaran Penelitian	4
1.8 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1 Data Pribadi	5
2.2 Kartu Tanda Penduduk Elektronik (E-KTP)	6
2.3 <i>Optical Character Recognition (OCR)</i>	6
2.4 <i>Pytesseract</i>	8
2.5 <i>Template Matching</i>	14
2.6 <i>Run-Length Smearing Algorithm (RLSA)</i>	16
2.7 <i>Levenshtein Distance</i>	16
2.8 <i>Radio Frequency Identification (RFID)</i>	17
2.9 Penelitian Terdahulu	19
2.10 <i>Hypertext Markup Language (HTML)</i>	20
2.11 Bootstrap	21
2.12 Angular	22
2.13 Node.js	23
2.14 Typescript	24
2.15 Python	24
2.16 Django	25

2.17	Maria DB	26
2.18	Visual Studio Code	26
2.19	PyCharm	27
2.20	XAMPP	28
BAB III METODE PENELITIAN		29
3.1	Analisis Masalah	30
3.2	Studi Literatur	30
3.3	Analisis Kebutuhan	30
3.4	Perancangan Aplikasi	31
3.5	Pengujian	49
BAB IV HASIL DAN PEMBAHASAN		52
4.1	Hasil Uji Coba	52
4.2	Evaluasi Pengujian	69
BAB V SIMPULAN DAN SARAN		73
5.1	Simpulan	73
5.2	Saran	73
DAFTAR PUSTAKA		74

DAFTAR GAMBAR

Gambar 2.1 Sistem Blok Diagram OCR	7
Gambar 2.2 Contoh hasil <i>grayscale</i>	8
Gambar 2.3 <i>Flowchart Pytesseract</i>	10
Gambar 2.4 Contoh <i>candidate chop points</i>	12
Gambar 2.5 <i>Flowchart Tempalte Matching</i>	15
Gambar 2.6 Contoh <i>before after</i> penggunaan metode RLSA	16
Gambar 2.7 Contoh alat RFID	18
Gambar 2.8 Contoh hasil pembacaan RFID	19
Gambar 3.1 Alur penelitian	29
Gambar 3.2 <i>Flowchart</i> perancangan metode <i>Pytesseract</i>	32
Gambar 3.3 <i>Flowchart</i> proses pengolahan <i>template</i>	33
Gambar 3.4 <i>Flowchart</i> perancangan metode <i>Template Matching</i>	35
Gambar 3.5 <i>Use Case Diagram</i>	37
Gambar 3.6 <i>Flowchart</i> alur penambahan data baru	39
Gambar 3.7 <i>Flowchart</i> alur <i>view/edit</i> data KTP	40
Gambar 3.8 <i>Flowchart</i> alur autentikasi data KTP	41
Gambar 3.9 <i>Conceptual Data Model</i> (CDM)	42
Gambar 3.10 <i>Physical Data Model</i> (PDM)	43
Gambar 3.11 <i>Mock up</i> menu <i>Home</i>	45
Gambar 3.12 <i>Mock up</i> menu <i>Customer</i>	46
Gambar 3.13 <i>Mock up form customer</i>	47
Gambar 3.14 <i>Mock up add</i> autentikasi KTP	48
Gambar 3.15 <i>Mock up view</i> data pribadi <i>customer</i>	48
Gambar 3.16 <i>Mock up edit</i> data pribadi <i>customer</i>	49
Gambar 4.1 <i>Source code</i> pemotongan gambar KTP	53
Gambar 4.2 Contoh hasil pemotongan gambar KTP	53
Gambar 4.3 Contoh hasil <i>image processing</i> NIK KTP	54
Gambar 4.4 Contoh hasil <i>image processing</i> data KTP selain NIK	54
Gambar 4.5 Contoh <i>source code</i> pembacaan <i>Pytesseract</i>	55
Gambar 4.6 <i>Tempalte</i> dan jenis tulisan yang digunakan	56

Gambar 4.7 Contoh hasil segmentasi karakter <i>template</i>	56
Gambar 4.8 Contoh hasil segmentasi karakter NIK KTP	57
Gambar 4.9 Contoh <i>source code</i> untuk proses <i>Template Matching</i>	57
Gambar 4.10 Contoh hasil RLSA untuk mengidentifikasi baris	58
Gambar 4.11 Contoh hasil segmentasi baris	58
Gambar 4.12 Contoh hasil RLSA untuk mengidentifikasi kata	58
Gambar 4.13 Contoh hasil segmentasi kata	58
Gambar 4.14 <i>Source code</i> algoritma <i>levenshtein distance</i>	60
Gambar 4.15 Contoh hasil balikan dari fungsi algoritma <i>levenshtein distance</i> yang sudah dijadikan satu array	60
Gambar 4.16 Menu <i>Home</i>	62
Gambar 4.17 Tampilan awal menu <i>Customer</i>	63
Gambar 4.18 <i>Form</i> penambahan data	63
Gambar 4.19 Contoh hasil pembacaan OCR	64
Gambar 4.20 Tampilan awal <i>Customer Authentication Data</i>	65
Gambar 4.21 Contoh tampilan setelah pembacaan uid KTP oleh RFID	65
Gambar 4.22 Contoh tampilan menu Customer setelah klik “Search”	66
Gambar 4.23 Contoh tampilan <i>view</i> data	67
Gambar 4.24 Contoh tampilan <i>edit</i> data	67
Gambar 4.25 Tampilan pertama ketika masuk menu Authentication	68
Gambar 4.26 Contoh jika uid yang dicari belum terdaftar	68
Gambar 4.27 Hasil uji <i>Shapiro-Wilk</i> untuk uji normalitas <i>Pytesseract</i>	70
Gambar 4.28 Hasil uji <i>Kruskal-Wallis</i> untuk uji nonparametrik <i>Pytesseract</i>	70
Gambar 4.29 Hasil uji <i>Shapiro-Wilk</i> untuk uji normalitas <i>Template Matching</i>	70
Gambar 4.30 Hasil uji <i>Bartlett</i> untuk uji homogenitas <i>Template Matching</i>	71
Gambar 4.31 Hasil uji <i>One Way Anova</i> untuk uji parametrik <i>Template Matching</i>	71
Gambar 4.32 Hasil uji untuk menentukan pasangan beda signifikan antara keempat pengkondisian pada metode <i>Template Matching</i>	71

DAFTAR TABEL

Tabel 2.1 Contoh fungsi dalam <i>library Pytesseract</i>	9
Tabel 3.1 Tabel Alamat	44
Tabel 3.2 Tabel Kelurahan	44
Tabel 3.3 Tabel Kecamatan	44
Tabel 3.4 Tabel Kota	44
Tabel 3.5 Tabel Provinsi	45
Tabel 3.6 Tabel Cust	45
Tabel 3.7 Tabel Pengujian Aplikasi <i>Web</i>	51
Tabel 4.1 Hasil Pengujian Metode <i>Pytesseract</i>	55
Tabel 4.2 Hasil Pengujian Metode <i>Template Matching</i>	59
Tabel 4.3 Hasil Pengujian Autentikasi	61
Tabel 4.4 Hasil Evaluasi Pengujian Aplikasi <i>Web</i>	72

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kartu tanda penduduk (KTP) wajib dimiliki oleh masyarakat Indonesia yang sudah berusia 17 tahun ke atas. KTP berfungsi sebagai kartu identitas dan tanda pengenal masyarakat Indonesia. Selain itu, KTP juga diperlukan ketika masyarakat ingin mengakses layanan publik dan program pemerintah. KTP diperlukan sebagai tanda pengenal dalam proses pendaftaran, seperti pendaftaran calon pegawai negeri sipil (CPNS) atau calon anggota TNI/Polri, pengajuan pernikahan, pembukaan akun tabungan, dan lain-lain. KTP juga diperlukan dalam proses kepengurusan berbagai surat seperti surat kehilangan dan surat kematian. Selain itu, KTP menjadi syarat untuk mendapat bantuan sosial, jaminan sosial, dan lain-lain.

Multifinance sebagai perusahaan pembiayaan memiliki beberapa persyaratan bagi nasabah. KTP menjadi salah satu berkas yang harus dilengkapi oleh nasabah jika ingin mengajukan peminjaman atau pembiayaan. KTP ini berfungsi sebagai data diri nasabah dan membantu *multipfinance* dalam mengingatkan nasabah mengenai pembayaran angsuran. Diketahui bahwa permintaan pembiayaan meningkat dari 8,9% di bulan Juni 2022 menjadi 10,3% di bulan Juli 2022 (Damara, 2022).

Selama ini, data-data dari KTP yang didaftarkan diketik secara manual. Hal ini membuat sistem pendaftaran menjadi lebih memakan waktu dan tenaga. Apalagi jika terdapat banyak KTP yang ingin didaftarkan, maka akan memakan banyak waktu. Selain itu, rawan juga terjadi kesalahan dalam pengetikan yang menyebabkan ketidaksesuaian pada data yang disimpan. Hal ini menyebabkan ketidakefektifan dalam proses kerja.

Selain karena ketidakefektifan kinerja, pada zaman ini juga banyak tersebar KTP palsu. Meskipun sudah terdapat hukum untuk mengatasi kasus manipulasi data kependudukan ini seperti diatur dalam UU No. 24 Tahun 2013, tidak menutup kemungkinan bahwa masih banyak oknum tidak bertanggung jawab yang berani untuk memalsukan dokumen kependudukan seperti KTP dan lain-lain.

Oleh karena itu, membuat proyek Tugas Akhir ini dibuat. Proyek Tugas Akhir ini dibuat untuk membantu otomatisasi input KTP menggunakan OCR dan mengautentikasi kebenaran KTP menggunakan *Radio Frequency Identification* (RFID). Hal ini dapat membantu mengatasi ketidakefektifan kinerja dalam penginputan data pribadi dan meminimalisir masalah KTP palsu.

OCR adalah proses untuk mengenali huruf atau karakter pada gambar. Penelitian mengenai penggunaan OCR sudah pernah dilakukan sebelumnya. Diwanti dkk. (2019) pernah menggunakan OCR untuk mendeteksi plat nomor kendaraan secara *real time* dengan tingkat keberhasilan 80%. Abdullah dan Muhammad (2020) pernah menggunakan metode *Template Matching* untuk OCR dengan tingkat keberhasilan sebesar 85%.

RFID adalah alat untuk mengidentifikasi atau mengambil data menggunakan *magnetic card* atau *barcode* (Bakti, 2019). Sopa dkk. (2019) pernah merancang sistem kehadiran dengan menggunakan RFID pada KTP. Selain itu, Hani dkk. (2019) pernah menggunakan RFID pada KTP untuk merancang sistem akses kunci elektronik pada kotak penyimpanan.

1.2 Identifikasi Masalah

Identifikasi masalah pada penelitian ini adalah banyaknya data KTP yang harus diinput sehingga menimbulkan potensi kesalahan dalam pengetikan serta tidak hemat waktu dan tenaga. Selain itu, identifikasi masalah lainnya adalah sulitnya mengetahui keaslian kartu identitas.

1.3 Batasan Masalah

Batasan masalah yang ditentukan dalam proyek Tugas Akhir ini sebagai berikut

1. Dokumen yang akan dibaca menggunakan metode *Optical Character Recognition* (OCR) adalah kartu tanda penduduk (KTP).
2. Verifikasi data KTP dilakukan berdasarkan kode dalam KTP yang akan dibaca menggunakan *Radio Frequency Identification* (RFID).

3. Verifikasi data KTP dilakukan berdasarkan data yang ada pada *database* lokal karena tidak memiliki akses data ke Direktorat Jenderal Kependudukan dan Pencatatan Sipil (Ditjen Dukcapil).

1.4 Perumusan Masalah

Pada penelitian ini ditemukan dua buah masalah yaitu sebagai berikut.

1. Bagaimana membantu proses input data pribadi yang berasal dari KTP dan meminimalisir terjadinya kesalahan pengetikan, serta menghemat waktu dan usaha.
2. Bagaimana membantu mengautentikasi data pribadi yang berasal dari KTP.

1.5 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut.

1. Melakukan otomatisasi input data kartu tanda penduduk (KTP) pelanggan dengan metode *Optical Character Recognition* (OCR).
2. Melakukan autentikasi terhadap data KTP menggunakan *Radio Frequency Identification* (RFID).

1.6 Manfaat

Manfaat yang diperoleh dari tugas akhir ini adalah sebagai berikut.

1. Bagi mahasiswa

Dengan adanya proyek ini, mahasiswa dapat menerapkan ilmu yang diperoleh selama melakukan perkuliahan di Universitas Ma Chung. Selain itu, proyek ini juga menambah pengalaman dan memperluas wawasan mahasiswa sebagai bekal untuk masuk ke dunia kerja.

2. Bagi Universitas Ma Chung, khususnya Program Studi Teknik Informatika

Dengan adanya proyek ini, Universitas Ma Chung, khususnya Program Studi Teknik Informatika dapat mempersiapkan mahasiswa untuk menjadi lulusan yang kompeten. Selain itu, proses pengerjaan proyek Tugas Akhir juga dapat menjadi bekal mahasiswa agar siap terjun di dunia kerja,

1.7 Luaran Penelitian

Luaran dari penelitian ini adalah sebuah program untuk membaca data dari kartu tanda penduduk (KTP) dan melakukan autentikasi data KTP, serta publikasi jurnal ilmiah

1.8 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini dibagi menjadi beberapa bab seperti berikut.

BAB I	Pendahuluan
	Bab satu ini berisi latar belakang, identifikasi masalah, batasan masalah, perumusan masalah, tujuan, manfaat, luaran Tugas Akhir, dan sistematika penulisan.
BAB II	Tinjauan Pustaka
	Bab dua ini terdiri dari uraian mengenai teori dari data pribadi, KTP, <i>Optical Character Recognition</i> (OCR), <i>Pytesseract</i> , <i>Template Matching</i> , <i>Run-Length Smearing Algorithm</i> (RLSA), serta <i>software</i> yang akan digunakan untuk mengembangkan aplikasi proyek Tugas Akhir.
BAB III	Analisis dan Perancangan Sistem
	Bab tiga ini berisi tentang tahapan perancangan proyek Tugas Akhir, yang terdiri dari analisis kebutuhan, perancangan metode OCR, serta perancangan aplikasi <i>web</i> yang akan dikembangkan.
BAB IV	Hasil dan Pembahasan
	Bab empat ini berisi hasil dan pembahasan dari proyek Tugas Akhir yang sudah dikerjakan, yang terdiri dari hasil dari metode OCR yang dikembangkan, hasil aplikasi <i>web</i> , evaluasi hasil, dan diskusi hasil.
BAB V	Kesimpulan dan Saran
	Bab lima ini berisi simpulan dari hasil proyek Tugas Akhir yang telah dikerjakan dan saran terhadap hal-hal yang dapat diperbaiki dari sistem untuk penelitian berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Data Pribadi

Berdasarkan Pasal 1 Undang-Undang Nomor 27 Tahun 2022, data pribadi adalah data perseorangan yang dapat diidentifikasi tersendiri maupun dikombinasikan dengan informasi lain secara langsung maupun tidak langsung melalui sistem elektronik maupun nonelektronik. UU No. 27 Tahun 2022 ini sendiri adalah undang-undang tentang Perlindungan Data Pribadi. Berdasarkan Pasal 4 UU No. 27 Tahun 2022, data pribadi terbagi menjadi dua berdasarkan sifatnya yaitu spesifik dan umum. Data pribadi yang bersifat spesifik adalah data kesehatan, data anak, data keuangan pribadi, dan data lain yang sudah diatur dalam peraturan perundang-undangan, sedangkan data yang bersifat umum adalah nama lengkap, jenis kelamin, kewarganegaraan, agama, dan data lain yang dapat digunakan untuk mengidentifikasi seseorang (JDIH BPK RI, 2022). Data pribadi yang bersifat umum ini dapat ditemukan dalam kartu tanda penduduk (KTP).

Pesatnya perkembangan teknologi memudahkan masyarakat dunia untuk saling terhubung tanpa mengenal batas wilayah. Teknologi sendiri sudah dimanfaatkan dalam berbagai bidang kehidupan, seperti permerintahan, perdagangan, dan kesehatan. Hal ini menyebabkan data pribadi seseorang menjadi mudah untuk dikumpulkan dan digunakan untuk merugikan pemiliknya. Oleh karena itu dibuatlah UU No. 27 Tahun 2022 ini untuk melindungi data pribadi setiap masyarakat Indonesia. Selain itu, perlindungan data pribadi ini juga merupakan perlindungan hak asasi manusia (HAM).

Data pribadi ini tidak bisa digunakan oleh sembarang orang. Berdasarkan Pasal 65 UU No. 27 Tahun 2022 mengenai Larangan Dalam Penggunaan Data Pribadi, setiap orang tidak boleh mengumpulkan, mengungkapkan, atau menggunakan data pribadi yang bukan miliknya. Pelanggaranya akan dikenai hukuman sesuai pasal 67 UU No 27 Tahun 2022 yaitu paling lama empat hingga lima tahun penjara, serta dapat dikenai denda paling banyak empat hingga lima miliar rupiah (JDIH BPK RI, 2022).

Selain itu, berdasarkan Pasal 66 UU No.27 Tahun 2022, setiap orang dilarang untuk memalsukan data pribadi. Hukuman bagi pelaku pemalsuan data pribadi diatur dalam sesuai pasal 68 UU No 27 Tahun 2022, yaitu pidana penjara paling lama enam tahun, serta dapat dikenai denda sebanyak enam miliar rupiah (JDIH BPK RI, 2022).

2.2 Kartu Tanda Penduduk Elektronik (E-KTP)

Kartu tanda penduduk elektronik atau yang disingkat e-KTP adalah dokumen kependudukan yang berbasis *database* kependudukan nasional dengan sistem keamanan secara administrasi dan teknologi informasi. KTP merupakan kartu identitas yang didalamnya terdapat berbagai data pribadi seperti nama, jenis kelamin, kewarganegaraan, dan agama. Dalam e-KTP terdapat Nomor Induk Kependudukan (NIK) yang menjadi identitas setiap penduduk. Setiap penduduk yang berusia diatas 17 tahun wajib memiliki e-KTP dan hanya dapat memiliki satu NIK semasa hidupnya. Berdasarkan Pasal 13 Undang-Undang Nomor 23 Tahun 2006 tentang Adminduk, NIK inilah yang dijadikan landasan dalam penerbitan Paspor, SIM, NPWP, dan dokumen identitas lainnya (Dispendukcapil Kota Semarang, 2015).

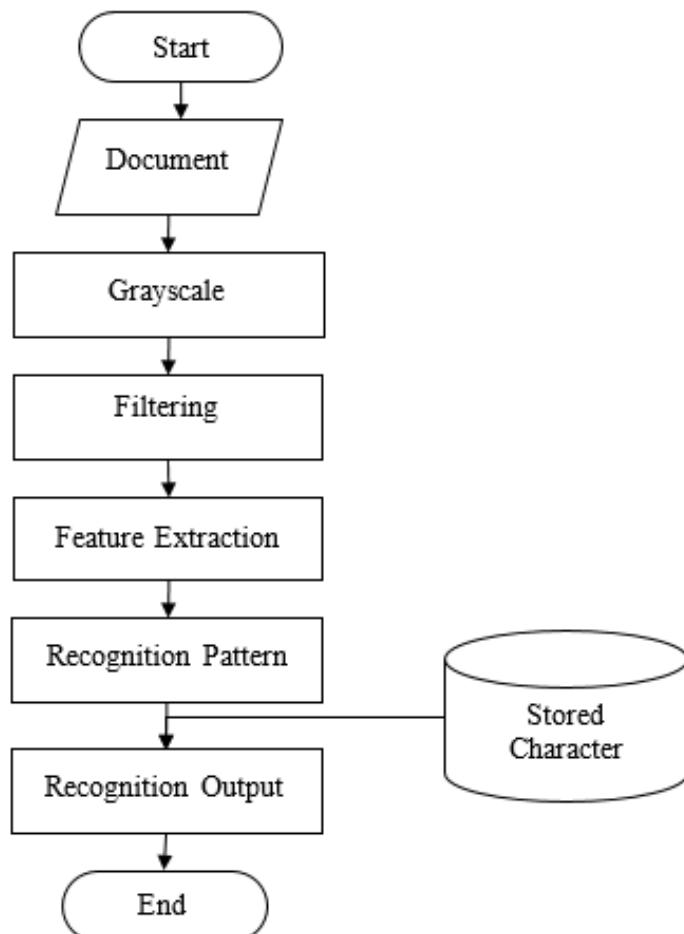
Selain itu, dalam e-KTP juga terdapat *chip* yang berisi sidik jari dari jempol dan telunjuk kanan yang direkam ketika pembuatan e-KTP. Perekaman sidik jari ini berfungsi sebagai verifikasi dan validasi karakter fisik penduduk. Sidik jari dipilih karena setiap manusia pasti memiliki gurat sidik jari yang berbeda serta biayanya yang murah (Dispendukcapil Kota Semarang, 2015).

2.3 Optical Character Recognition (OCR)

Character recognition dapat diklasifikasikan menjadi dua yaitu *Handwritten Character Recognition* (HCR) dan *Optical Character Recognition* (OCR). HCR cenderung memiliki akurasi yang lebih rendah dibandingkan OCR. Hal ini karena pada HCR terdapat perbedaan bentuk dan tipe tulisan (Phangriastu, 2017 *cit.* Rao dkk., 2016).

Optical Character Recognition (OCR) adalah proses untuk mendekripsi dan mengenali huruf atau karakter dari gambar dan mengkonversinya menjadi

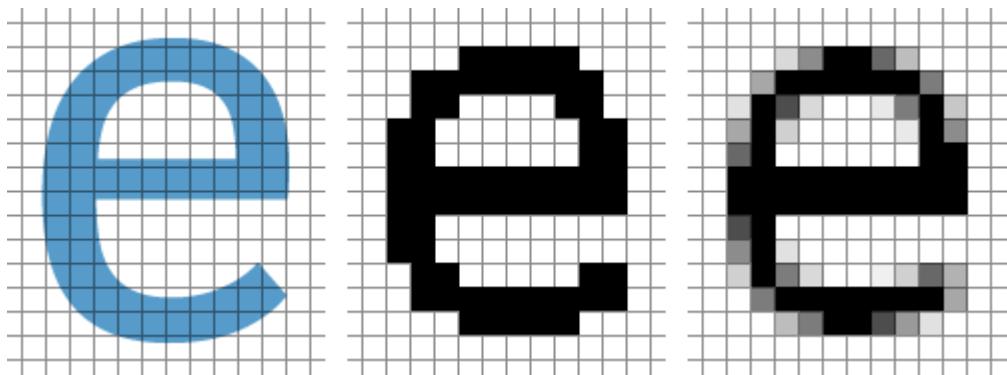
American Standard Code for Information Interchange (ASCII) atau karakter yang dikenali komputer dan dapat diedit (Phangriastu., 2017 *cit.* Rao dkk., 2016). Gambar yang akan diproses menggunakan OCR dapat diperoleh dari teks yang difoto atau dipindai menjadi gambar digital, maupun dari gambar digital langsung.



Gambar 2.1 Sistem Blok Diagram OCR (Phangriastu, 2017 *cit.* Mohammad dkk., 2014)

Dalam mendekripsi teks menggunakan OCR dapat dilakukan melalui beberapa langkah seperti pada Gambar 2.1. Pertama gambar yang diinputkan akan diubah menjadi *grayscale* untuk meningkatkan keakuratan data. Proses *grayscale* ini dilakukan menggunakan persamaan 2-1 yang akan menghasilkan citra digital seperti pada Gambar 2.2 (Phangriastu, 2017 *cit.* Mohammad dkk., 2014).

$$Y = 0.2126R + 0.7152G + 0.0722B \quad (2-1)$$



Gambar 2.2 Contoh hasil *grayscale* (Phangriastu, 2017 *cit. Mohammad dkk., 2014*)

Hasil dari *grayscale* kemudian dilakukan *filtering* dan *feature extraction*. *Feature extraction* adalah proses yang dilakukan untuk mendapatkan informasi dari gambar yang nantinya akan digunakan untuk proses klasifikasi. Pada proses *feature extraction* ini dilakukan seperti memotong data gambar untuk tiap huruf atau karakter, menormalisasi ukuran pixel, atau mengubah data pixel menjadi binary (Phangriastu, 2017 *cit. Mohammad dkk., 2014*).

2.4 Pytesseract

Python-tesseract atau yang sering disebut dengan *Pytesseract* adalah *tools* Python yang digunakan untuk *Optical Character Recognition* (OCR). *Pytesseract* adalah *tools* pembungkus untuk mesin Tesseract-OCR Google. *Pytesseract* dapat membaca berbagai format gambar yang didukung oleh *library* pencitraan, seperti format jpeg, png, gif, dan lain-lain. *Pytesseract* biasanya digunakan untuk membaca teks digital (Indrakusuma dkk., 2021).

Pytesseract harus diinstall terlebih dahulu sebelum digunakan. Setelah diinstall, kemudian *Pytesseract* dapat digunakan dengan diimpor. Dalam penggunaanya biasanya *Pytesseract* digunakan bersamaan dengan *library* OpenCV. Dalam OCR, OpenCV biasanya digunakan untuk mengolah gambar sebelum dilakukan *character recognition* (Hoffstaetter dkk., 2022).

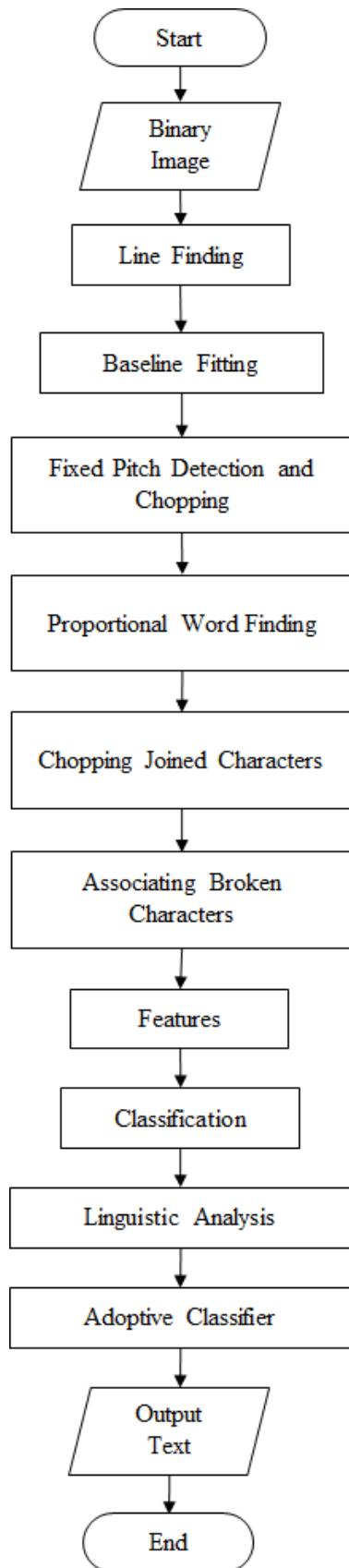
Pytesseract memiliki beberapa fungsi dari *library Tesseract* yang dapat membantu memudahkan dalam melakukan OCR seperti pada tabel 2.1 (Hoffstaetter dkk., 2022).

Tabel 2.1 Contoh fungsi dalam *library Pytesseract*

Fungsi	Keterangan
image_to_string	Memberikan output dalam bentuk string dari proses <i>Tesseract</i> OCR
image_to_boxes	Memberikan <i>output</i> huruf atau karakter yang sudah dikenali dan batas kotaknya
image_to_data	Memberikan hasil yang terdiri dari batas kotak dari huruf atau karakter yang dikenali, <i>confidences</i> , dan lain-lain.
run_and_get_output	Memberikan hasil kasar dari <i>Tesseract</i> OCR

Pada *Tesseract* 4 ditambahkan algoritma *long short-term memory* (LSTM) yang berfokus pada pengenalan baris. Meskipun begitu, penambahan ini masih mendukung OCR *Tesseract* 3 yang mengenali pola karakter. Kompabilitas *Tesseract* 3 menggunakan mesin OCR yang membutuhkan *trained data* yang terdapat pada repository tessdata.

Secara keseluruhan, algoritma *Pytesseract* dapat dilihat pada Gambar 2.3. Algoritma pembacaan text pada gambar oleh metode *Pytesseract* dibagi menjadi lima tahapan besar yaitu *line and word finding*, *word recognition*, *static character classifier*, *linguistic analysis*, dan *adaptive classifier*. Tahapan *line and word finding*, tahapan *word recognition*, dan tahapan *static character classifier* dibagi menjadi beberapa tahapan lagi. Tahapan *line and word finding* dibagi menjadi empat tahapan yaitu *line finding*, *baseline fitting*, *fixed pitch detection and chopping*, dan *proportional word finding*. Tahapan *word recognition* dibagi menjadi dua tahapan yaitu *chopping joined characters* dan *associating broken characters*. Tahapan *static character classifier* dibagi menjadi dua tahapan yaitu *features* dan *classification*.



Gambar 2.3 *Flowchart Pytesseract*

2.4.1 Line and Word Finding

Tahapan *line and word finding* adalah tahapan untuk menentukan baris dan karakter yang akan dibaca. Tahapan *line and word finding* dibagi menjadi empat tahapan yaitu *line finding*, *baseline fitting*, *fixed pitch detection and chopping*, dan *proportional word finding*.

Tahapan *line finding* adalah tahapan untuk menentukan baris dari gambar yang akan dibaca. Diasumsikan bahwa karakter yang akan dibaca memiliki ukuran yang sama. Filter akan menghilangkan huruf kapital. Kemudian akan dicari tinggi rata-rata dari setiap karakter. Filter kemudian akan mengurutkan dan memproses karakter berdasarkan koordinat x dan mengurnya menjadi satu baris (Ferdaush & Islam, 2021).

Tahapan *baseline fitting* adalah tahapan untuk menentukan garis dasar pada suatu baris. Tahapan ini membantu *Tesseract* untuk membaca karakter pada halaman yang melengkung. Garis dasar dipasang menggunakan *quadratic spline*. Garis dasar ditentukan dengan mempartisi karakter menjadi kelompok. *Quadratic spline* kemudian dipasang pada partisi terpadat dengan kecocokan kuadrat terkecil (Ferdaush & Islam, 2021).

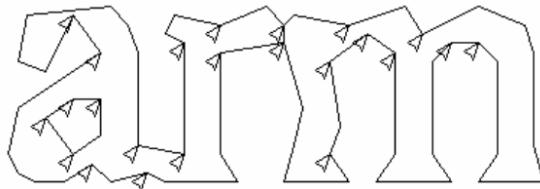
Tahapan *pitch detection and chopping* adalah tahapan untuk memotong setiap karakter. Pada tahapan ini, *Tesseract* menguji baris teks untuk menentukan apakah pola karakter. *Tesseract* kemudian memotong setiap kata menjadi karakter berdasarkan pola tersebut (Ferdaush & Islam, 2021).

Tahapan *proportional word finding* adalah tahapan untuk menentukan spasi pada teks dengan pola yang tidak tetap. *Tesseract* mengukur celah dalam rentang vertikal antara garis dasar dan garis rata-rata. Spasi yang mendekati ambang batas dibuat kabur sehingga penentuan spasi akan dibuat setelah pengenalan kata (Ferdaush & Islam, 2021).

2.4.2 Word Recognition

Tahapan *word recognition* adalah tahapan untuk mengidentifikasi segmentasi kata menjadi karakter. Tahapan *word recognition* dibagi menjadi dua tahapan yaitu *chopping joined characters* dan *associating broken characters*.

Tahapan *chopping joined characters* adalah tahapan untuk memisahkan karakter yang masih saling bergabung. Pada tahapan ini, *Tesseract* memotong gumpalan dengan tingkat keyakinan terburuk dari proses klasifikasi karakter. Titik potong ditentukan berdasarkan simpul cekung dari pendekatan poligonal pada garis, contohnya seperti pada Gambar 2.4. Setidaknya diperlukan tiga titik potong untuk memisahkan karakter. Proses pemisahan karakter dilakukan secara bertahap. Jika pemotongan tidak dapat meningkatkan tingkat keyakinan, pemotongan akan dibatalkan (Ferdaush & Islam, 2021).



Gambar 2.4 Contoh *candidate chop points* (Smith, 2007)

Tahapan *associating broken characters* adalah tahapan untuk menggabungkan kembali karakter agar dapat dibaca menjadi huruf. Ketika hasil pemotongan karakter kurang baik, proses akan dialihkan pada asosiator. Asosiator kemudian menggunakan algoritma pencarian A*. Algoritma pencarian A* akan menarik bagian dari urutan dan mengevaluasinya dengan mengklasifikasikan bagian yang tidak terklarifikasi (Ferdaush & Islam, 2021).

2.4.3 *Static Character Classifier*

Tahapan *static character classifier* adalah tahapan untuk mengklasifikasikan atau mengenali karakter. Tahapan *static character classifier* dibagi menjadi dua tahapan yaitu *features* dan *classification*.

Tahapan *features* adalah tahapan untuk mencocokkan pola karakter. Segmen dari pendekatan polygonal digunakan untuk fitur. Dalam proses pengenalan, karakter yang kecil dan memiliki panjang yang tetap diekstraksi dari garis luar dan dicocokkan dengan prototipe dari data training. Fitur yang diekstraksi dari karakter yang akan dibaca terdapat dalam bentuk 3 dimensi (x, y, sudut) dengan biasanya 50-100 fitur dalam satu karakter, sedangkan fitur prototipe terdapat dalam

bentuk 4 dimensi (x, y, sudut, panjang) dengan biasanya 10-20 fitur dalam satu konfigurasi prototipe (Ferdaush & Islam, 2021).

Tahapan *classification* dibagi menjadi dua tahap. Pada tahap pertama, *class purner* membuat daftar pendek kelas karakter yang mungkin sesuai. Setiap karakter mengambil 3 dimensi yang dikuantisasi kasar, vektor bit kelas yang mungkin sesuai, dan vektor bit dijumlahkan di semua fitur. Kelas dengan jumlah tertinggi menjadi daftar pendek untuk tahap berikutnya. Setiap karakter mencari vektor prototipe dari kelas yang cocok, kemudian kesamaan aktualnya dihitung. Setiap kelas karakter prototipe diwakili oleh konfigurasi sehingga proses perhitungan jarak menyimpan catatan kesamaan total setiap karakter di setiap konfigurasi. Jarak gabungan terbaik dihitung berdasarkan karakter yang dijumlahkan dan bukti prototipe (Ferdaush & Islam, 2021).

2.4.4 Linguistic Analysis

Tahapan *linguistic analysis* adalah tahapan untuk menganalisis kata berdasarkan modul linguistik. Setiap ada segmentasi baru, modul linguistik memilih kata terbaik yang tersedia berdasarkan kategori kata yang sering digunakan, kata dalam kamus, kata numerik, huruf besar, huruf kecil, dan kata pilihan pengklasifikasi teratas. Kata dari segmentasi berbeda dapat menghasilkan jumlah karakter yang berbeda sehingga *Tesseract* menggunakan dua angka untuk setiap klasifikasi. Angka pertama adalah tingkat kepercayaan dikurangi jarak yang dinormalisasi dari prototipe. Angka kedua adalah peringkat yang mengalihkan jarak normalisasi dari prototipe dengan total panjang garis luar pada karakter yang tidak diketahui. Peringkat untuk karakter dapat dijumlahkan karena total panjang garis luar untuk semua karakter dalam sebuah kata selalu sama (Ferdaush & Islam, 2021).

2.4.5 Adaptive Classifier

Tahapan *static character classifier* adalah tahap yang lebih cenderung menggeneralisasi semua jenis font sehingga kurang dapat membedakan karakter. Oleh karena itu diadakan tahapan *adaptive classifier* dengan pengklasifikasian yang lebih sensitif terhadap font. Perbedaannya adalah pada *adaptive classifier* menggunakan normalisasi garis dasar atau *x-height normalization* sedangkan pada

static character classifier menggunakan sentroid dan normalisasi ukuran anisotropik. Normalisasi garis dasar atau *x-height normalization* mempermudah dalam membedakan karakter huruf besar dan kecil, serta meningkatkan kekebalan terhadap *noise*. Normalisasi juga menghapus rasio aspek font dan sedikit goresan font sehingga pengenalan subskrip lebih sederhana (Ferdaush & Islam, 2021).

2.5 Template Matching

Template Matching adalah salah satu teknik yang dapat digunakan untuk *Optical Character Recognition* (OCR). *Template Matching* adalah teknik yang mencocokkan setiap bagian dalam suatu gambar dengan gambar yang dijadikan acuan menggunakan aturan tertentu. Penentuan kecocokan gambar pada *Template Matching* menggunakan nilai kesamaan yang dapat dihitung menggunakan rumus 2-2 (Maskuri, 2017).

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

atau

$$d(p, q) = d(q, p) = \sqrt{\sum_{n=1}^n (q_1 - p_1)^2 \dots \dots} \quad (2-2)$$

Keterangan :

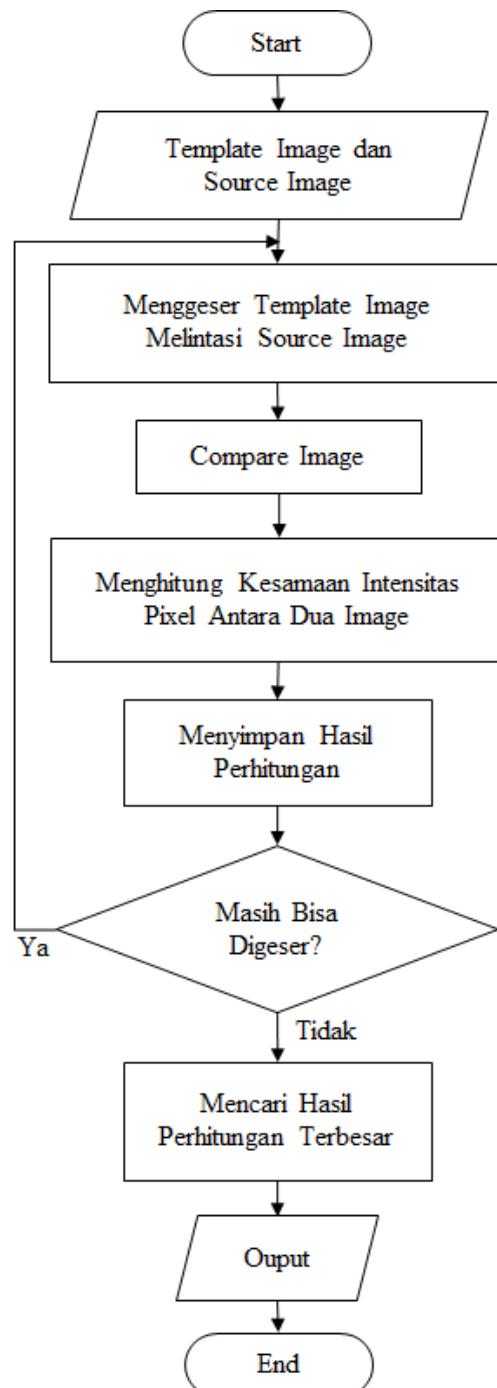
d = nilai terkecil dari dua gambar

q = nilai biner dari gambar baru

Template Matching memiliki beberapa kelebihan dan kekurangan. Kelebihan template matching adalah menggunakan data biner sehingga komputasinya tidak terlalu besar dan mudah menyiapkan data referensinya sehingga mudah untuk dituliskan dalam bahasa pemrograman. Kekurangan dari *Template Matching* adalah membutuhkan banyak referensi untuk mendapatkan hasil yang optimal (Maskuri, 2017).

Pada metode *Template Matching*, sebelumnya disiapkan karakter *template* yang sesuai terlebih dahulu. Kemudian dilakukan *image processing* seperti

pengaturan kecerahan, *grayscale*, segmentasi baris dan karakter, serta *resize template* agar karakter dapat semakin menyerupai *template*. Proses pencocokan karakter dengan *template* dilakukan dengan membandingkan setiap *pixel*. Kemudian diambil jumlah selisih nilai *pixel* terkecil (Widodo dan Gunawan, 2015). Diagram flowchart untuk *Template Matching* dapat dilihat pada Gambar 2.5.

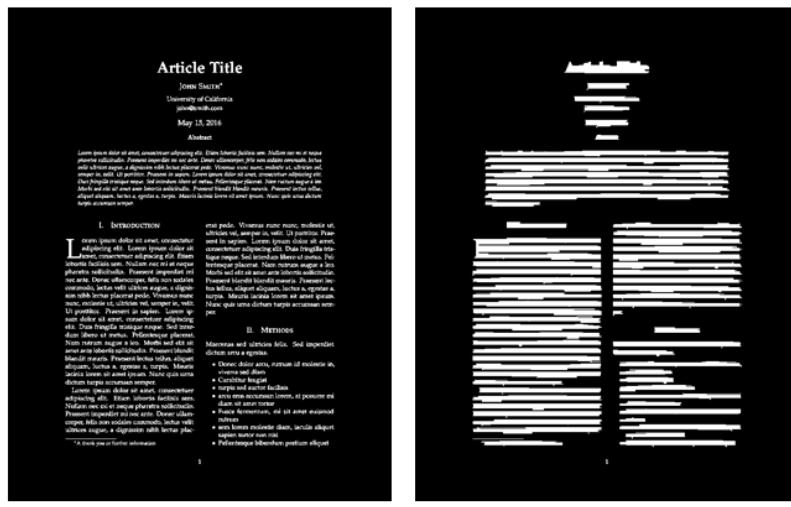


Gambar 2.5 Flowchart *Template Matching*

2.6 Run-Length Smearing Algorithm (RLSA)

Run-Length Smearing Algorithm (RLSA) adalah metode untuk membantu proses pencarian lokasi teks dalam suatu gambar. RLSA dapat membantu menemukan lokasi teks secara vertikal maupun horizontal (Putra dkk., 2021). RLSA membantu mempermudah dalam proses segmentasi baris dan kata.

Pada metode RLSA dilakukan proses *smearing* yaitu proses *scanning* baris pada gambar secara horizontal dan vertikal sebanyak tiga kali. Pada proses *smearing* pertama dilakukan pencarian pixel berwarna hitam. Pixel hitam ini adalah jarak dari setiap karakter. Kemudian pada proses *smearing* kedua dicari pixel berwarna putih. Pixel putih ini adalah lebar dari karakter. Berikutnya, kedua hasil ini digabungkan dan dilakukan proses *smearing* ketiga. Proses *smearing* ketiga ini digunakan untuk menemukan pixel berwarna hitam yang adalah besar dari lokasi teks secara utuh. Selanjutnya pixel-pixel berwarna putih dihubungkan menjadi suatu area yang mencakup dua kata atau lebih. Contoh hasil dari RLSA yang digunakan dapat dilihat pada Gambar 2.6 (Putra dkk., 2021).



Gambar 2.6 Contoh before after penggunaan metode RLSA (Valle, 2019)

2.7 Levenshtein Distance

Levenshtein distance adalah algoritma untuk menghitung perbedaan atau selisih dari dua string. Algoritma ini ditemukan pertama kali pada tahun 1965 oleh Vladimir Levenshtein. Selisih dihitung dengan menentukan angka minimum untuk mengubah string yang satu menjadi string yang lainnya. Dalam algoritma ini

terdapat tiga operasi utama yaitu operasi penyisipan karakter (*insertion*), penghapusan karakter (*deletion*), dan penukaran karakter (*substitution*) (Adawiyah & Saragih, 2022).

Algoritma *levenshtein distance* dimulai dengan menghitung panjang kedua string yang ingin diuji terlebih dahulu. Kemudian akan dicek panjang keduanya. Jika ada salah satunya nol, algoritma akan berhenti dan hasil jaraknya adalah nol. Jika kedua string bukan string kosong, perhitungan jarak dilakukan dengan menjumlahkan dua karakter terakhir dari string pertama dan mentransformasikannya menjadi jumlah dari dua karakter terakhir string kedua. Jika karakter dengan index kedua dari terakhir dari string pertama sama dengan karakter dengan index kedua dari terakhir dari string kedua, nilai *cost* yang didapat adalah 0 dan nilai jaraknya adalah dengan mentransformasikan karakter terakhir dari string pertama menjadi karakter terakhir dari string kedua. Jika tidak, terdapat tiga kemungkinan lainnya. Kemungkinan pertama yaitu karakter dengan index kedua dari terakhir dari string pertama akan ditransformasikan menjadi karakter dengan index kedua dari terakhir dari string kedua dengan nilai *cost* 1 sehingga nilai jaraknya didapat dengan mentransformasikan karakter terakhir dari string pertama menjadi karakter terakhir dari string kedua ditambah 1. Kemungkinan kedua dengan menghapus karakter dengan index kedua dari terakhir dari string pertama dan mengubah karakter terakhir dari string pertama menjadi jumlah dari dua karakter dari string kedua sehingga karakter terakhir dari string pertama diubah menjadi jumlah dari dua karakter terakhir dari string kedua ditambah 1. Kemungkinan ketiga adalah dengan menghapus karakter dengan index kedua dari terakhir dari string kedua dan mengubah jumlah dua karakter terakhir dari string pertama menjadi karakter terakhir dari string kedua, kemudian akan dihitung nilai jaraknya (Trivusi, 2022).

2.8 Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) adalah teknologi untuk mengidentifikasi dan mengambil data menggunakan *magnetic card* atau *barcode*. Proses identifikasi RFID menggunakan gelombang elektromagnetik sehingga

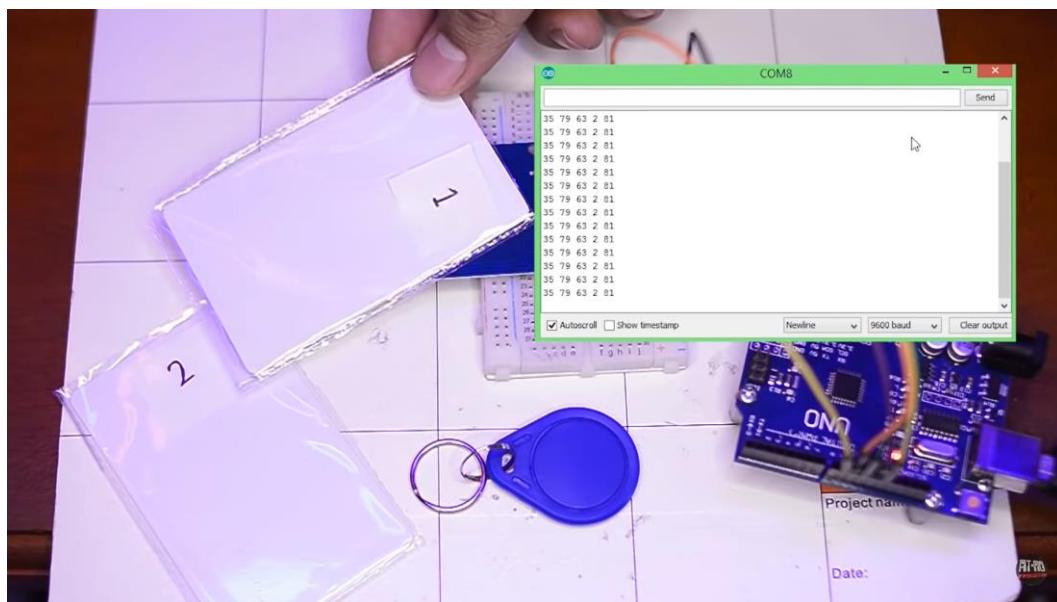
RFID membutuhkan *tag* dan *reader* agar dapat berfungsi dengan baik (Bakti, 2019). Contoh RFID dapat dilihat pada Gambar 2.7.



Gambar 2.7 Contoh alat RFID

RFID *tag* adalah alat yang akan ditempelkan pada objek yang akan diidentifikasi oleh RFID *reader*, sedangkan RFID *reader* adalah alat yang dapat membaca RFID *tag*. Baik RFID *tag* maupun RFID *reader* dapat dibagi menjadi dua, yaitu aktif dan pasif. RFID *tag* aktif memerlukan baterai, sedangkan RFID *tag* pasif tidak. Setiap *tag* pada RFID *tag* berbeda-beda. Informasi yang tersimpan pada objek yang terhubung pada *tag* hanya terdapat pada *database* yang terhubung dengan RFID *reader*. RFID *reader* aktif dapat memancarkan sinyal interrogator ke RFID *tag* dan menerima balasan autentikasinya, sedangkan RFID *reader* pasif hanya dapat menjangkau hingga 600 meter dan hanya dapat menerima sinyal dari *tag* aktif (Bakti, 2019).

Pada RFID, *tag* dapat dibaca meskipun melewati RFID *reader* ataupun dalam kondisi ditutupi oleh objek lain. Selain itu, RFID *reader* dapat membaca banyak *tag* dalam satu waktu. Oleh karena itu, RFID banyak digunakan di perusahaan, supermarket, atau tempat lainnya (Bakti, 2019). Gambar 2.8 adalah contoh hasil pembacaan RFID *tag* oleh RFID *reader*.



Gambar 2.8 Contoh hasil pembacaan RFID (PRODUCTION, 2020)

2.9 Penelitian Terdahulu

Optical Character Recognition (OCR) menggunakan metode *Pytesseract* dan *Template Matching* ini bukanlah yang pertama. Sebelumnya sudah pernah ada penelitian OCR menggunakan metode *Pytesseract* dan *Template Matching* untuk berbagai objek.

Pytesseract pernah digunakan sebagai metode untuk melakukan OCR terhadap plat nomor kendaraan. Pada penelitian ini, Sawalkar *et al.*, (2022) mendapatkan tingkat keberhasilan sebesar 60%. Diwanti dkk. (2019) juga pernah melakukan penelitian dengan judul “*Real Time Smart CCTV Untuk Mendeteksi Plat Nomor Kendaraan Menggunakan Optical Character Recognition*”. Pada penelitian tersebut, Diwanti dkk. (2019) menggunakan metode Pytesseract dengan tingkat keberhasilan 80%. Pada penelitian Salar *et al.*, (2021), metode Pytesseract digunakan untuk mengidentifikasi tulisan tangan dari gambar dengan tingkat keberhasilan 90%. Uddin (2021) pernah melakukan penelitian dengan menggunakan metode Pytesseract dengan judul “*Features Extraction of Tax Card by Using OCR Based Deep Learning Techniques*”. Pada penelitian tersebut, tingkat keberhasilan yang dicapai adalah 80%.

Template Matching pernah digunakan sebagai metode untuk melakukan OCR sebelumnya. Asih (2017) pernah melakukan penelitian pengenalan huruf

dengan OCR menggunakan metode *Template Matching*. Berdasarkan penelitian tersebut didapat hasil bahwa komputer dapat mengenali huruf dalam citra digital dengan baik menggunakan algoritma *Template Matching*. Selain penelitian tersebut, Abdullah dan Muhammad (2020) juga pernah menggunakan metode *Template Matching* untuk pengenalan E-KTP. Pada penelitian Abdullah dan Muhammad (2020) didapat tingkat keberhasilan sebesar 85%. Metode *Template Matching* juga pernah digunakan untuk mengenali rambu lalu lintas oleh Chrisdwianto dkk. (2018). Pada penelitian Chrisdwianto dkk. (2018), tingkat keberhasilannya mencapai 88%. Muharom (2019) pernah menggunakan metode *Template Matching* untuk mengenali nomor ruangan dengan tingkat keberhasilan 81%. Rahman dkk. (2018) pernah melakukan penelitian pengenalan iris mata menggunakan metode *Template Matching* dengan keberhasilan 86.36%. Nugroho dan Prabandaru (2022) juga pernah menggunakan metode *Template Matching* untuk mendeteksi gambar nomor panggil buku di perpustakaan ITATS dengan tingkat keberhasilan 87%.

Penelitian menggunakan RFID juga sudah banyak dilakukan. Sopa dkk. (2019) dalam penelitiannya yang berjudul “Sistem Kehadiran Menggunakan RFID pada E-KTP Berbasis *Internet of Things*” berhasil melakukan pembacaan kode E-KTP menggunakan RFID *reader*. Selain itu, ada penelitian dari Hani dkk. (2019) tentang “Perancangan Sistem Akses Kunci Elektronik Pada Kotak Penyimpanan Memanfaatkan E-KTP dan Teknologi RFID” yang juga berhasil membaca kode KTP menggunakan RFID.

2.10 Hypertext Markup Language (HTML)

Hypertext Markup Language atau yang dikenal dengan HTML adalah bahasa pemrograman yang menggunakan *tag* atau simbol-simbol tertentu untuk membuat halaman *website*. HTML adalah dasar dari semua bahasa pemrograman yang diaplikasikan ke *website*, seperti PHP dan CSS. *Tag* atau symbol-simbol HTML digunakan untuk menyatakan kode-kode yang akan dibaca dan ditampilkan oleh *browser*. *Tag* pada HTML dapat digunakan untuk membuat tulisan di dalam *website* menjadi bervariasi. Kode-kode dalam HTML disimpan ke dalam file yang diberikan imbuhan .html dibelakang nama filenya. Kode-kode HTML dapat

dituliskan dengan berbagai *text editor* seperti Notepad++, Sublime, Visual Code, dan lain-lain.

HTML dikembangkan pertama kali pada tahun 1991 oleh Tim Bernes-Lee. Tim Bernes-Lee adalah seorang fisikawan di lembaga penelitian CERN (Organisasi Eropa untuk Riset Nuklir) yang terletak di Swiss. Pada awal dikembangkannya, HTML hanya berisi sekitar 18 *tag*. Meskipun sudah dikembangkan sejak tahun 1991, HTML baru dirilis tahun 1995, yaitu HTML 2.0. HTML kemudian terus mengalami perkembangan. HTML 4.01 yang dirilis pada tahun 1999 telah digunakan untuk mengembangkan berbagai web. HTML terbaru adalah HTML 5 (Aprilia, 2021).

Pada HTML terdapat tiga komponen utama yang terdiri dari tag, elemen, dan atribut. Elemen tersusun dari *tag* yang mengapit objek. *Tag* adalah kode yang akan dibaca oleh *browser*. *Tag* terdiri dari *tag* pembuka (`<...>`) dan *tag* penutup (`</...>`) yang mengapit objek. Elemen adalah gabungan dari *tag* dan objek. Atribut adalah perintah tambahan yang ada di dalam sebuah elemen. Atribut selalu ditulis berpasangan dengan nilainya (Hermawan, 2021).

2.11 Bootstrap

Bootstrap adalah salah satu *framework* yang dapat membantu mempermudah dalam merancang tampilan sebuah website. Bootstrap membantu *developer* dalam merancang website menjadi lebih menarik dan responsif. Bootstrap bersifat *open-source* dan mudah diaplikasikan sehingga banyak digunakan oleh *developer* (Adani, 2021).

Bootstrap pertama kali dikembangkan oleh programmer Twitter bernama Mark Otto dan Jacob Thornton. Pada awalnya, Bootstrap dikenal sebagai Twitter Blueprint. Bootstrap pertama kali dirilis menjadi proyek *open-source* pada Agustus 2011. Bootstrap 2.0 dirilis pada Januari 2012 dengan menambahkan fitur *layouting* yang responsif. Sejak dirilisnya Bootstrap 2.0, Bootstrap sudah banyak digunakan dalam proyek-proyek di GitHub. Bootstrap 3.0 yang dirilis pada Agustus 2013 sudah dapat mengakomodasi tampilan *mobile*. Bootstrap 4.0 dirilis pada Januari 2018 dan sudah mengalami banyak perubahan dibandingkan sebelumnya. Bootstrap terbaru adalah Bootstrap 5.0 (Adani, 2021).

Bootstrap adalah *class* yang tersusun dari kumpulan file CSS dan JavaScript. Untuk menggunakan Bootstrap, *developer* hanya perlu memanggil *class* tersebut. Bootstrap memiliki banyak jenis *class* yang memiliki fungsinya masing-masing, seperti *class* table untuk membuat tabel, *class* img untuk mengatur tampilan gambar, *class* button untuk membuat tombol, dan lain-lain (Haekal, 2021).

2.12 Angular

Angular adalah framework JavaScript yang dapat digunakan untuk membuat *Single-Page-Applications* yang reaktif. Angular ditulis dalam bentuk TypeScript. Angular dikelola oleh Google. Angular menyediakan struktur standar yang membantu *developer* sehingga dapat mudah dikelola (Wibowo, 2021).

Misko Hevery dan Adam Abrons pertama kali mengembangkan Angular pada tahun 2009. Angular 1 ini disebut juga AngularJS. Pada saat itu, Angular yang dikembangkan memiliki ukuran kode yang besar dan tingkat kesulitan yang tinggi sehingga developernya memutuskan untuk memodifikasi kembali kodennya. Setelah kode tersebut selesai dimodifikasi muncullah Angular 2. Angular 3 kemudian dilewati untuk mencegah terjadinya kebingungan karena sebelumnya telah merilis paket router versi 3.3.0, kemudian Angular 4 dirilis pada Maret 2017. Pada tahun 2017 – 2019, Angular 5 – 8 dirilis. Kemudian pada tahun 2020 dirilis Angular 9 dan 10 (Inggihpangestu, 2022).

Arsitektur dasar Angular terdiri dari modul, *component*, dan *service*. Modul Angular disebut NgModule. Dalam Angular dapat dipecah menjadi beberapa modul dengan minimal satu module, yaitu *root module* yang disebut AppModule. Setiap modul memiliki *routing* yang akan mengarahkan ke url halaman yang dituju. *Component* pada Angular adalah *view* untuk menampilkan HTML. *Component* pada Angular memiliki fungsi *directive* yang dapat menampilkan elemen pada DOM HTML. *Service* pada Angular digunakan untuk melakukan HTTP request. *Service* digunakan untuk menyalurkan data antar *component*, seperti mengambil atau menyimpan data (Angular, 2022).

Angular memiliki beberapa fitur yang membuatnya menarik di kalangan *developer*, seperti *Model-View-Controller* (MVC), *pipes*, *data model binding*, *directive*, dan lain-lain. *Pipes* adalah fitur Angular yang memungkinkan *developer*

untuk mendeklarasikan tampilan nilai yang berada pada template HTML. *Data binding* adalah konsep yang dimiliki Angular untuk menyatakan komunikasi antar *component* dan DOM yang memudahkan dalam mengambil atau menyimpan data. Secara teknis, *directive* adalah *component*. *Directive* muncul di dalam sebuah elemen sebagai atribut atau sebagai *target binding* (Angular, 2022).

2.13 Node.js

Node.js adalah *runtime environment* untuk menjalankan JavaScript dari sisi server. Node.js bersifat *open source* dan *cross platform*. Node.js dirancang sebagai aplikasi *data-intensive-real-time* yang dapat berjalan di beberapa platform. Node.js memiliki performa yang tinggi karena dikembangkan menggunakan Engine JavaScript V8 milik Google (Appkey, 2020).

Node.js ditemukan pada tahun 2009 oleh seorang *developer* Joyent bernama Ryan Dahl. Sebelum menemukan Node.js, Ryan Dahl sudah mengamati cara kerja JavaScript selama 13 tahun. Ryan Dahl menemukan kekurangan JavaScript yaitu hanya berjalan pada sisi browser saja. Ryan Dahl kemudian mengembangkan sebuah *engine* yang bernama Engine V8. *Engine* tersebut dapat menjalankan JavaScript di luar browser, yang sekarang dikenal dengan nama Node.js (Appkey, 2020).

Node.js menggunakan arsitektur “*Single Threaded Event Loop*”. Pemrosesan Node.js didasarkan pada *event-based* JavaScript dan mekanisme *callback* JavaScript. Arsitektur Node.js terbagi menjadi dua komponen utama yaitu Engine Javascript V8 dan Libuv *Library*, serta pola desain. Engine JavaScript V8 adalah *engine* milik Google yang dikembangkan menggunakan Bahasa pemrograman C++. *Engine* ini berfungsi untuk meng-*compile* JavaScript menjadi kode tingkat *assembly*. Libuv Library merupakan *multiplatform library* yang dilengkapi dengan *event loop* dan dirancang secara *asynchronous I/O*. Libuv Library memiliki *thread pool* untuk menangani *thread* setiap operasi I/O. Node.js memiliki dua *design pattern* yaitu *object pool* yang berisi kumpulan objek untuk *task* tertentu, dan *facade* yang memberikan tampilan *interface* untuk *body code*-nya (Rabbani, 2022).

2.14 TypeScript

TypeScript adalah salah satu bahasa pemrograman yang digunakan untuk membuat website. TypeScript adalah bahasa pemrograman yang dikembangkan berdasarkan JavaScript. TypeScript nantinya akan dikonversi menjadi JavaScript agar dapat dimengerti browser. File TypeScript disimpan menggunakan ekstensi .ts (Oktriwina, 2020).

TypeScript pertama kali dikembangkan oleh Anders Hejlsberg dari Microsoft. TypeScript versi pertama dirilis pada tahun 2012. Pada awalnya, TypeScript dikembangkan untuk membantu memecahkan permasalahan penggunaan JavaScript pada proyek berskala besar. Saat ini, TypeScript sudah banyak digunakan dalam berbagai *software* seperti Slack, Asana, dan Visual Studio Code. Banyak juga *framework* yang ditulis dalam TypeScript seperti Angular, Aurelia, dan Ionic (Oktriwina, 2020).

TypeScript memiliki tiga komponen utama yaitu bahasa pemrograman, *compiler*, dan *TypeScript Language Service*. Bahasa pemrograman TypeScript terdiri dari sintaksis, *keyword*, dan anotasi. *Compiler* TypeScript berfungsi untuk mengkonversi kode-kode TypeScript menjadi JavaScript untuk dibaca oleh browser. *TypeScript Language Service* serupa dengan editor yang berfungsi untuk menampilkan tambahan di sekitar inti *pipeline* (Oktriwina, 2020).

2.15 Python

Python adalah bahasa pemrograman interpretatif. Python berfokus pada keterbacaan kode sehingga memiliki kode pemrograman yang jelas, lengkap, dan mudah dipelajari. Bahasa pemrograman Python cukup sederhana sehingga sangat cocok untuk digunakan dalam pembuatan aplikasi berbasis *artificial intelligence* (AI). Python dapat digunakan untuk mengembangkan *software* dan dapat berjalan di berbagai platform sistem operasi, seperti Linux, Windows, Mac OS X, dan lain-lain (Enterprise, 2019).

Python pertama kali dikembangkan oleh programmer Belanda bernama Guido van Rossum di Centrum Wiskunde & Informatika (CWI) Amsterdam pada tahun 1991. Python adalah kelanjutan dari bahasa pemrograman ABC. Nama “Python” sendiri dipilih oleh Guido van Rossum dari acara televisi “Monty

Python's Flying Circus". Pengembangan Python saat ini dilakukan oleh *programmer* yang dikoordinasikan oleh Guido dan Python Software Foundation (Enterprise, 2019).

Python banyak digemari di kalangan developer karena fitur dan kelebihannya. Dalam Python sudah disediakan banyak *library* yang siap dipakai oleh para *programmer* dan developer. Struktur bahasa Python jelas, sederhana, dan mudah dipelajari sehingga sangat cocok untuk pemula. Python juga berorientasi multi-paradigma, yaitu procedural dan objek. Python bersifat modular sehingga mudah dikembangkan melalui modul-modul baru. Selain itu, Python memiliki sistem pengolahan memori otomatis atau yang disebut *garbage collection* (Enterprise, 2019).

2.16 Django

Django adalah salah satu *framework* yang menggunakan bahasa pemrograman Python. Pada Django terdapat berbagai modul, *library*, dan API yang dapat digunakan oleh developer. Hal ini dapat mempermudah dan mempercepat kinerja developer. Django juga merupakan *fullstack framework* karena dapat berperan sebagai *frontend framework* sekaligus *backend framework* (Ariffudin, 2022).

Django menerapkan arsitektur Model Template View atau yang disingkat MTV. Model mengurus seluruh aktivitas *backend* pada Django, sedangkan *Template* dan *View* menangani seluruh aktivitas *frontend*. Model pada arsitektur MTV Django berfungsi untuk memproses dan mengolah data yang didapat dari *database*. *Template* pada arsitektur MTV Django berperan sebagai *View*. *Template* ini berfungsi untuk menampilkan data sesuai perintah. *View* pada arsitektur MTV Django berperan sebagai *Controller*. *View* ini berfungsi untuk mengirimkan data dari model menuju *Template* dan menerima data dari *user* (Ariffudin, 2022).

Django memiliki banyak kelebihan sehingga banyak digunakan oleh perusahaan terkemuka seperti *Instagram*, *Spotify*, dan *Pinterest*. Django sebagai *framework* Python dapat menjalankan berbagai fitur Python. Django juga memiliki berbagai modul, *library*, dan API sehingga dapat membantu mempermudah pembuatan *website*. Fitur MTV Django memiliki *template* yang dapat membantu

mempercepat proses pengembangan *website*. Selain itu, Django memiliki fungsi *Cryptographic Hash* yang berfungsi untuk melakukan enkripsi dan dekripsi pada *password*. Django juga memiliki prinsip *Don't Repeat Yourself* (DRY) yang membuat developer tidak perlu menuliskan kode yang sama berulang kali. Hal ini dapat meringkas kode. Django memiliki prinsip modular sehingga setiap komponen bersifat independen. (Ariffudin, 2022).

2.17 MariaDB

MariaDB adalah salah satu *software relational database management system* (DBMS). MariaDB disebut sebagai *fork* karena dapat dianggap sebagai versi lain dari MySQL. MariaDB memiliki fitur yang kompatibel dengan MySQL. MariaDB banyak digunakan oleh perusahaan besar seperti Accenture, Grooveshark, Docplanner, dan lain-lain (Huda, 2022).

MariaDB dikembangkan oleh tim yang dipimpin oleh Michael Widenius pada tahun 2010. Nama MariaDB sendiri berasal dari nama anak perempuannya. MariaDB pertama kali dikembangkan karena pada tahun 2009 MySQL diakuisisi oleh Oracle Corporation sehingga menjadi produk yang memiliki lisensi. Hal ini menciptakan kekhawatiran karena pengembangan MySQL menjadi tidak leluasa (Risyan, 2020).

MariaDB memiliki berbagai fungsi sebagai DBMS. MariaDB dapat membaca *query* dengan cepat, serta memiliki kemampuan *processing* dan fleksibilitas yang tinggi. MariaDB memiliki kapasitas menampung hingga 5 miliar baris dan 60 ribu tabel. MariaDB juga dapat mendeteksi pesan error sehingga memudahkan developer dalam mengembangkan aplikasi menggunakan MariaDB. (Huda, 2022).

2.18 Visual Studio Code

Visual Studio Code atau yang disingkat VS Code adalah sebuah aplikasi yang digunakan untuk mengedit kode. Visual Studio Code adalah aplikasi ringan dan kuat yang tersedia di Windows, macOS, dan Linux. VS Code dapat digunakan untuk mengembangkan aplikasi menggunakan bahasa pemrograman JavaScript,

TypeScript, C++, C#, Java, Python, PHP, dan bahasa pemrograman lainnya (Ariffudin, 2022).

Visual Studio Code adalah salah satu aplikasi yang paling banyak digunakan. Visual Studio Code memiliki fitur *basic editing* yang membuat kegiatan mengedit kode menjadi lebih mudah karena dilengkapi dengan berbagai *keyboard shortcuts*, *multiple selections*, dan *column selections*. Visual Studio Code juga memiliki fitur Intellisense yang memudahkan *developer* dalam menuliskan kode. Fitur ini mirip seperti *auto complete* yang menyarankan keseluruhan kata berdasarkan apa yang sudah diketik. Selain itu, dalam Visual Studio Code terdapat fitur *debugging* yang memudahkan *developer* mengedit, meng-*compile*, dan menjalankan kode berulang kali. Visual Studio Code memiliki *extension marketplace* untuk mengunduh *tools*, *debuggers*, ataupun bahasa pemrograman tambahan. Visual Studio Code berintegrasi dengan GitHub sehingga dapat membantu *developer* untuk dapat saling berbagi kode dan berkolaborasi dengan *developer* lain tanpa menggunakan *software* lainnya (Ariffudin, 2022).

2.19 PyCharm

PyCharm adalah salah satu *Integrated Development Environment* (IDE) terbaik untuk mengembangkan aplikasi atau program menggunakan bahasa pemrograman Python. PyCharm dapat digunakan untuk mengedit atau meng-*compile* program yang menggunakan bahasa pemrograman Python versi 2.7 dan Python versi 3.5 ke atas. Meskipun dikembangkan untuk bahasa pemrograman Python, PyCharm dikembangkan oleh JetBrains yang merupakan pengembang dari IntelliJ IDEA IDE. PyCharm tersedia di Linux, macOS, dan Windows (Appkey, 2021).

PyCharm cukup populer karena dapat digunakan di berbagai platform dan memiliki banyak fitur yang mempermudah proses pengembangan aplikasi. PyCharm memiliki tingkat pemahaman kode yang tinggi dan dapat mempermudah proses pembacaan kode dengan pemberian warna berbeda pada kata kunci, kelas, fungsi, sintaksis, dan kesalahan kode. Selain itu, PyCharm dapat membantu memberikan instruksi dan saran untuk kesalahan kode atau proses penyelesaian kode. PyCharm juga dapat mengintegrasikan berbagai alat lain seperti Anaconda,

IPython, Kite, Pylint, pytest, dan WakaTime. Pada edisi Profesional, PyCharm dapat digunakan untuk mengembangkan bahasa pemrograman untuk web seperti HTML, CSS, dan Javascript dan dapat berintegrasi dengan Google App Engine untuk mengembangkan dan menghosting aplikasi berbasis web (Appkey, 2021).

Meskipun memiliki berbagai kelebihan, PyCharm juga memiliki kekurangan. Versi berbayar PyCharm cukup mahal dan sulit untuk memperbaiki *tools* seperti *virtual environment* atau *venv* karena dapat menimbulkan masalah. Hal ini membuat PyCharm kurang cocok untuk pemula. Selain itu, PyCharm membutuhkan banyak memori dan ruang penyimpanan (Appkey, 2021).

2.20 XAMPP

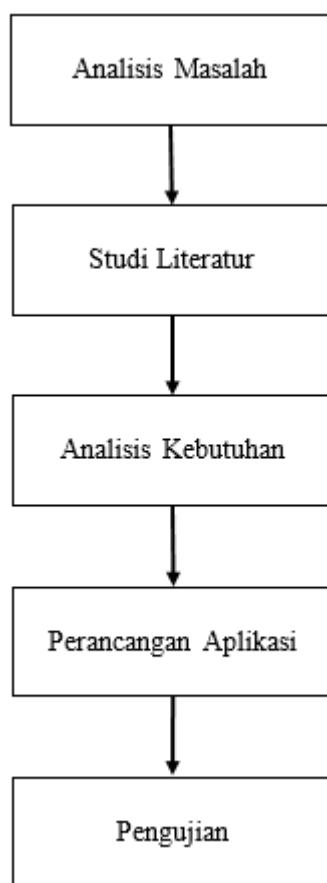
XAMPP adalah *software open source* untuk membangun *website local*. Setiap huruf dalam “XAMPP” memiliki arti masing-masing. Huruf X mewakili kata *Cross* yang berarti bahwa XAMPP dapat digunakan *cross platform*. Huruf A mewakili kata *Apache* yang merupakan web server *default* XAMPP. Huruf M adalah MariaDB, yaitu *software database management system* (DBMS) *default* dari XAMPP. Huruf P setelah huruf M mewakili PHP, yaitu bahasa pemrograman yang digunakan untuk mengembangkan *website*. Huruf P paling akhir adalah Perl, yaitu bahasa pemrograman yang digunakan untuk membangun *website* yang kompleks (Ariffudin, 2022).

XAMPP dikembangkan oleh Kai Oswald Seidler dan Kay Vogelgesang pada tahun 2002. Latar belakang dari dikembangkannya XAMPP adalah karena pada saat itu, Kai dan Kay ingin mempromosikan *Apache Web Server*. Pada saat itu, *Apache* dianggap susah untuk diisntal di komputer dan susah untuk dihubungkan dengan modul lain seperti MariaDB, PHP, dan Perl. Oleh karena itu, Kai dan Kay mengembangkan proyek bernama *Apache Friends* dan mengembangkan XAMPP (Ariffudin, 2022).

BAB III

METODE PENELITIAN

Proyek Tugas Akhir ini bertujuan untuk mengembangkan aplikasi untuk melakukan otomatisasi input data kartu tanda penduduk (KTP) dengan metode *Optical Character Recognition* (OCR) serta mengautentikasi KTP. Proses pengerjaan proyek Tugas Akhir ini dibagi menjadi beberapa tahapan. Berikut adalah diagram alur dari proses pengerjaan proyek Tugas Akhir.



Gambar 3.1 Alur penelitian

Seperti pada Gambar 3.1, tahapan penelitian dimulai dengan tahapan analisis masalah. Pada tahapan ini dilakukan identifikasi masalah dan batasan masalah yang menjadi pendahuluan pada proses pengerjaan proyek Tugas Akhir. Setelah masalah diidentifikasi, berikutnya adalah studi literatur untuk mempelajari terlebih dahulu mengenai literatur yang diperlukan dalam proses pengerjaan proyek Tugas Akhir. Selanjutnya adalah menganalisis kebutuhan dan menentukan apa saja

yang akan diperlukan dalam proses penggerjaan proyek Tugas Akhir. Berikutnya adalah merancang aplikasi yang nantinya akan dikembangkan. Semua yang sudah dirancang kemudian dikembangkan menjadi sebuah aplikasi pada tahap pelaksanaan penelitian. Tahapan terakhir adalah merancang pengujian yang nantinya akan dilakukan pada program atau aplikasi yang telah dikembangkan.

3.1 Analisis Masalah

Masalah yang terjadi adalah karena banyaknya data kartu tanda penduduk (KTP) yang harus dimasukkan. Proses penginputan secara manual menyebabkan sering terjadinya kesalahan pengetikan, serta tidak hemat waktu dan tenaga. Oleh karena itu, nantinya akan dilakukan otomatisasi proses penginputan data KTP menggunakan metode *Optical Character Recognition* (OCR).

Keaslian KTP juga sulit untuk diidentifikasi sehingga banyak terjadi kasus pemalsuan KTP. Untuk mengatasi hal ini, nantinya akan dikembangkan fitur untuk memverifikasi uid KTP menggunakan *Radio Frequency Identification* (RFID). Verifikasi akan dilakukan berdasarkan data yang ada pada *database* lokal karena tidak memiliki akses data ke Direktorat Jenderal Kependudukan dan Pencatatan Sipil (Ditjen Dukcapil).

3.2 Studi Literatur

Studi literatur dilakukan untuk memahami terlebih dahulu mengenai materi dari metode, *framework*, dan *tools* yang akan digunakan. Dengan adanya studi literatur akan mempermudah dalam mengembangkan program atau aplikasi nantinya. Studi literatur yang dilakukan adalah dengan mencari dan memahami teori melalui jurnal artikel maupun artikel dari website.

3.3 Analisis Kebutuhan

Perancangan penelitian adalah tahapan untuk menganalisa mengenai apa saja yang dibutuhkan dalam penggerjaan proyek Tugas Akhir. Berdasarkan hasil dari studi literatur, aplikasi akan dikembangkan berbasis *web* menggunakan *framework* Angular dan Bootstrap untuk *frontend* dan Django untuk *backend*. Untuk mengembangkan *Optical Character Recognition* (OCR) akan menggunakan

metode *Pytesseract* dan *Template Matching* yang membutuhkan *library* seperti OpenCV, rlsa-phython, numpy, imutils, base64, Pytesseract. Aplikasi akan dikembangkan menggunakan laptop dengan spesifikasi processor Intel® Core™ i7-8565U CPU, 8 GB RAM, dan 64 bit operating system, serta memiliki kamera dengan spesifikasi resolusi 1280x720, 720p HD perekam audio/video, dan *super high dynamic range imaging* (SHDR) (Acer, 2019). Untuk pengujian metode akan diperlukan 10 KTP dan satu KTP tambahan sebagai uji coba hasil yang akan ditampilkan pada laporan, kamera belakang *smartphone* 13 MP, kamera laptop, dan LUX meter. Untuk keperluan data mengenai kelurahan, kecamatan, kota, dan provinsi di Indonesia diambil dari github (Pasaribu, 2018).

3.4 Perancangan Aplikasi

Secara garis besar, proses perancangan aplikasi dibagi menjadi dua yaitu perancangan metode *Optical Character Recognition* (OCR) dan perancangan aplikasi web. Perancangan ini digunakan untuk mempermudah proses pengembangan program nantinya.

1. Perancangan Metode OCR

Sistem yang digunakan untuk pembacaan data pribadi yang berasal dari kartu tanda penduduk (KTP) akan menggunakan metode *Optical Character Recognition* (OCR). Metode OCR yang akan digunakan akan membandingkan antara dua metode yaitu *Pytesseract* dan *Template Matching*. Cara untuk membandingkan antara metode *Pytesseract* dan *Template Matching* nantinya akan menggunakan 10 KTP. KTP tersebut akan dibaca menggunakan metode *Pytesseract* dan *Template Matching*.

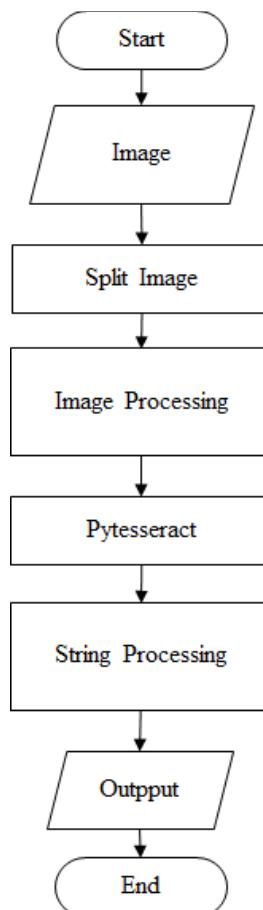
Perancangan metode OCR dibagi menjadi perancangan metode *Pytesseract*, perancangan metode *Template Matching*, dan perancangan metode OCR untuk digunakan pada aplikasi web.

a. Perancangan Metode *Pytesseract*

Diagram *flowchart* untuk proses perancangan metode *Pytesseract* dapat dilihat pada Gambar 3.2. Pada tahap pertama, gambar akan dibagi menjadi dua yaitu gambar NIK dan gambar data selain NIK. Pada kedua gambar tersebut kemudian

dilakukan *image processing* seperti *grayscale* dan *threshold*. Hasil dari *image processing* itu yang kemudian akan dikirim untuk dibaca oleh *Pytesseract*.

Hasil pembacaan *Pytesseract* akan berbentuk string. Untuk hasil pembacaan NIK akan langsung dimasukkan dalam array string, sedangkan untuk hasil pembacaan data selain NIK harus dipecah terlebih dahulu karena hanya dalam satu string kemudian akan dimasukkan ke dalam array yang sama dengan yang digunakan NIK sebelumnya. Array tersebut kemudian dipecah lagi berdasarkan index untuk menentukan data NIK, nama, tempat lahir, tanggal lahir, jenis kelamin, alamat, RT, RW, kelurahan, desa, kecamatan, agama, status perkawinan, pekerjaan, kewarganegaraan, dan waktu berlakunya KTP. Kemudian dilakukan *string procession* untuk menghapus karakter kotor seperti simbol-simbol dan kelebihan spasi. *String procession* ini digunakan untuk meningkatkan keakuratan hasil pembacaan yang sudah dilakukan. Hasil dari *string procession* ini yang akan menjadi *output* atau hasil akhir dari pembacaan KTP oleh *Pytesseract*.

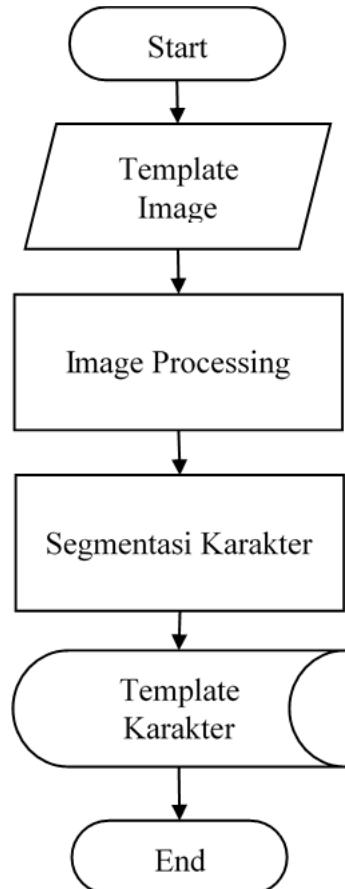


Gambar 3.2 Flowchart perancangan metode *Pytesseract*

b. Perancangan Metode *Template Matching*

Proses dalam metode *Template Matching* dimulai dengan menyiapkan *template* terlebih dahulu. Untuk KTP akan disiapkan dua *template* dengan jenis tulisan yang berbeda, yaitu OCR A Extended untuk NIK dan Arial untuk data selain NIK. *Template* dengan jenis tulisan OCR A Extended terdiri dari angka 0 – 9, sedangkan *template* dengan jenis tulisan Arial terdiri dari angka 0 – 9 dan huruf kapital A – Z (Asia, 2022).

Template yang sudah disiapkan akan diproses terlebih dahulu, seperti pada Gambar 3.3. Pada *template* yang sudah disiapkan pertama dilakukan *image processing* seperti *grayscale* dan *threshold*. Hasil dari *image processing* tersebut kemudian dilakukan segmentasi karakter untuk memisahkan antara karakter yang satu dengan yang lain. Setiap hasil dari segmentasi karakter kemudian dimasukkan ke dalam array untuk dijadikan *template* pada proses pembacaan *Template Matching*.

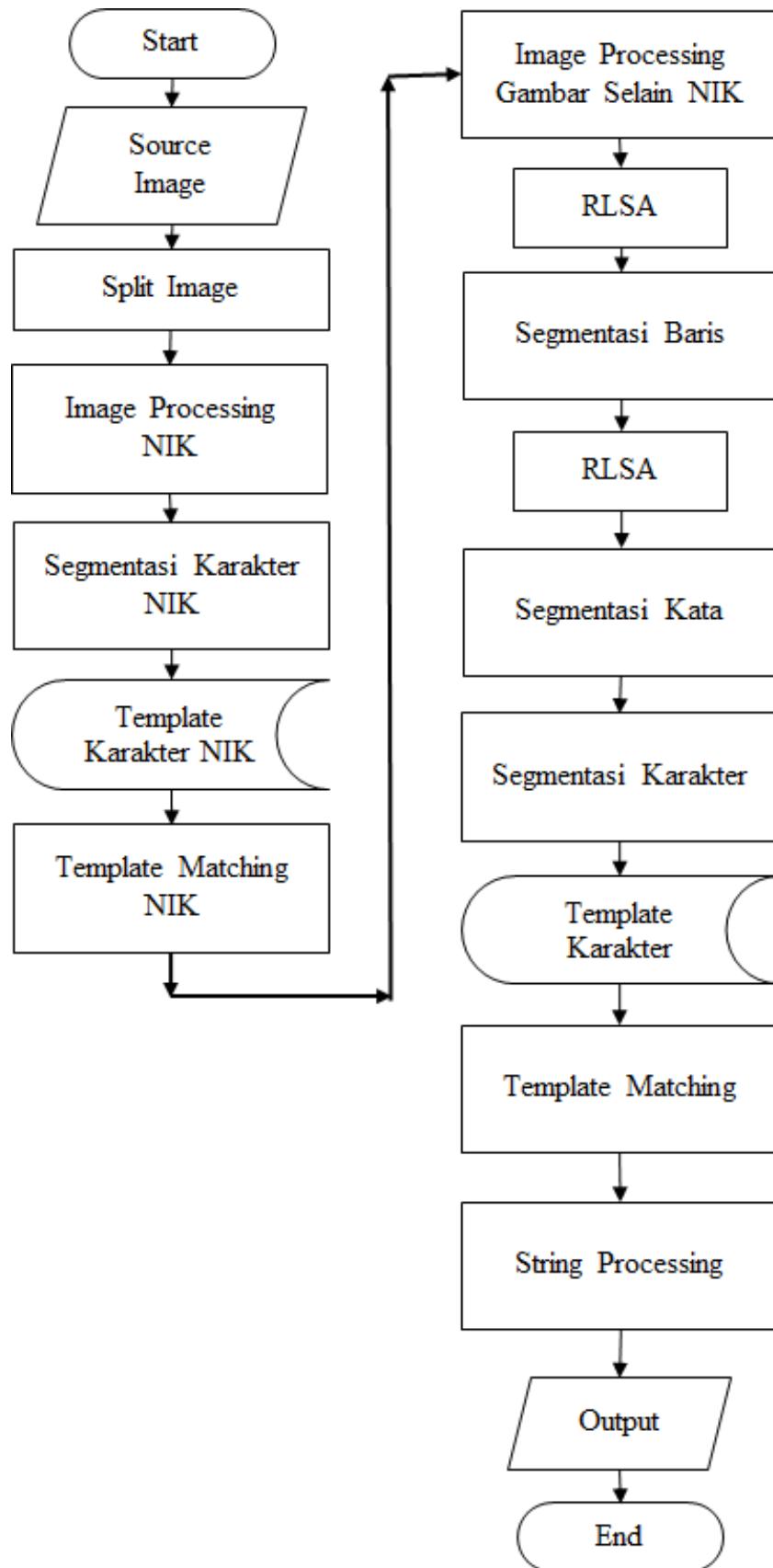


Gambar 3.3 *Flowchart* proses pengolahan *template*

Setelah mengolah *template*, berikutnya adalah memulai proses *Template Matching* pada KTP, seperti pada Gambar 3.4. Tahap selanjutnya adalah membagi KTP menjadi dua yaitu gambar NIK dan gambar data selain NIK. Kemudian pada kedua gambar dilakukan *image processing* seperti *grayscale* dan *threshold*. Setelah *image processing*, untuk gambar NIK KTP langsung dilakukan segmentasi karakter untuk memisahkan setiap karakternya. Hasil dari segmentasi karakter itu yang akan dibaca atau dicocokkan dengan *template* yang sudah diolah sebelumnya menggunakan metode *Template Matching*. Selanjutnya hasil pembacaan NIK tersebut langsung dimasukkan ke dalam array.

Untuk data selain NIK, setelah *image processing* dilakukan metode *run-length smearing algorithm* (RLSA) untuk mengidentifikasi letak setiap barisnya. Dari hasil RLSA tersebut diambil ukurannya untuk selanjutnya dilakukan segmentasi baris. Kemudian di setiap barisnya dilakukan lagi metode RLSA untuk mengidentifikasi setiap katanya. Berikutnya, dari hasil RLSA kedua diambil ukurannya dan dilakukan segmentasi kata dari hasil identifikasi tersebut. Dari kata-kata yang sudah dipisahkan akan dilakukan segmentasi karakter. Setiap karakternya kemudian dilakukan pembacaan atau pencocokan dengan *template* menggunakan metode *Template Matching*. Kata-kata yang dipisahkan akan disatukan dengan tambahan spasi. Dan hasil pembacaan untuk setiap barisnya kemudian akan dimasukkan ke array yang sudah ada NIK sebelumnya.

Pada array yang sudah terdapat semua hasil pembacaan dari metode *Template Matching* akan dibagi menjadi masing-masing komponen data KTP, seperti NIK, nama, tempat lahir, tanggal lahir, jenis kelamin atau gender, alamat, RT, RW, kelurahan, kecamatan, agama, status perkawinan, pekerjaan, kewarganegaraan, dan waktu berlakunya KTP. Pembagian ini dilakukan sesuai urutan pada array. Kemudian akan dilakukan *string processing*, seperti perubahan suatu string menjadi string tertentu dan penghapusan spasi yang berlebih. Hasil dari *string processing* ini yang akan menjadi hasil akhir dari pembacaan KTP oleh *Template Matching*.



Gambar 3.4 *Flowchart* perancangan metode *Template Matching*

c. Perancangan OCR Yang Digunakan

Setelah program pembacaan menggunakan metode *Pytesseract* dan *Template Matching* selesai dikembangkan, nantinya akan dicari metode yang memiliki hasil akurasi terbaik. Hasil akurasi akan dihitung pada pengujian metode menggunakan 10 KTP. Metode dengan hasil terbaik inilah yang akan dijadikan metode pembacaan OCR. Setelah ditemukan metode terbaik, nantinya pada tahap *string processing* akan dimasukkan algoritma *levenshtein distance* untuk komponen KTP gender, agama, status perkawinan, dan kewarganegaraan. Algoritma ini berfungsi untuk membantu proses *autocorrect* ketika terjadi kesalahan pembacaan.

2. Perancangan Aplikasi Web

Selain merancang metode OCR yang akan digunakan, tahap perencanaan ini juga merancang berbagai diagram. Diagram pertama yang dirancang adalah *use case diagram*. *Use case diagram* ini berfungsi untuk menentukan apa saja yang dapat dilakukan oleh *user*, Melalui *use case diagram* ini nantinya akan ditentukan fitur-fitur yang akan dikembangkan pada aplikasi.

Setelah merancang *use case diagram*, berikutnya adalah merancang diagram *flowchart*. Diagram *flowchart* ini digunakan untuk merancang alur. Alur ini berfungsi untuk menentukan jalannya proses input data pribadi yang berasal dari KTP, serta proses autentikasi data KTP.

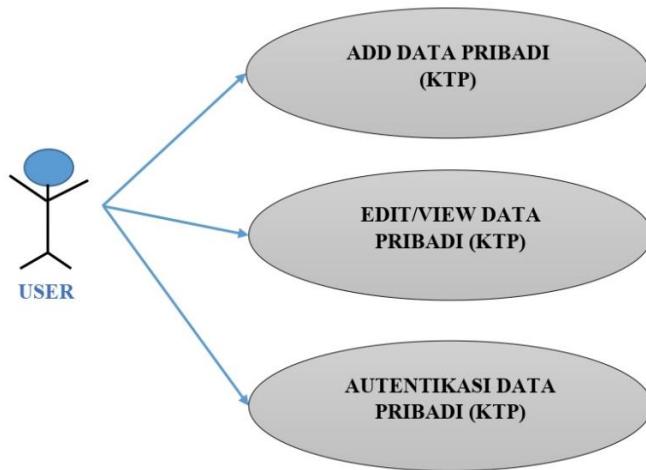
Tahap perancangan selanjutnya adalah tahap perancangan *database*. Dalam tahap perancangan *database* ini dilakukan perancangan *Entity Relationship Diagram* (ERD). Tahap perancangan ERD dibagi menjadi dua, yaitu tahap perancangan *Conceptual Data Model* (CDM) dan tahap perancangan *Physical Data Model* (PDM). Dengan adanya tahap perancangan *database* ini dapat membantu menentukan tabel dan attributnya. Selain itu, diagram ERD ini dapat membantu menentukan relasi antar tabel yang diperlukan oleh sistem.

Setelah merancang *database* yang diperlukan, berikutnya adalah merancang *mock up*. *Mock up* ini akan menjadi gambaran dari tampilan aplikasi yang akan dibuat. *Mock up* dapat membantu menentukan komponen-komponen yang akan ditampilkan dalam aplikasi. *Mock up* juga dapat membantu merancang *interface* aplikasi.

Hasil dari tahap perancangan sistem ini adalah berbagai diagram yang nantinya dapat membantu mempermudah proses pengembangan aplikasi. Diagram-diagram yang dihasilkan pada tahapan ini adalah *use case diagram* untuk membantu merancang fitur-fitur pada aplikasi nantinya, diagram *flowchart* untuk membantu menentukan alur jalannya aplikasi, *Entity Relationship Diagram* (ERD) yang terdiri dari *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) dari tahap perancangan *database*, dan *mock up* untuk merancang *interface* aplikasi.

a. *Use Case Diagram*

Diagram pertama yang dirancang adalah *use case diagram*. *Use case diagram* untuk proyek Tugas Akhir ini dapat dilihat pada Gambar 3.5. Dalam *use case diagram* ini hanya terdapat satu *entity* yaitu *user*. *Use case diagram* ini berfungsi untuk menentukan fitur-fitur apa saja yang dapat dilakukan oleh *user*. Fitur yang dapat dilakukan oleh *user* ada tiga. Pertama, *user* dapat menambah dan melakukan otomatisasi penginputan data peribadi yang diambil dari kartu tanda penduduk (KTP). Berikutnya, *user* dapat melihat (*view*) data yang sudah diinputkan sebelumnya dari KTP dan dapat mengubah data tersebut. *User* juga dapat melakukan autentikasi terhadap data KTP jika data dari KTP sudah pernah diinputkan sebelumnya.



Gambar 3.5 *Use Case Diagram*

b. *Diagram Flowchart*

Setelah merancang *use case diagram*, diagram berikutnya yang dirancang adalah *diagram flowchart*. *Diagram flowchart* dibuat agar membantu mengetahui

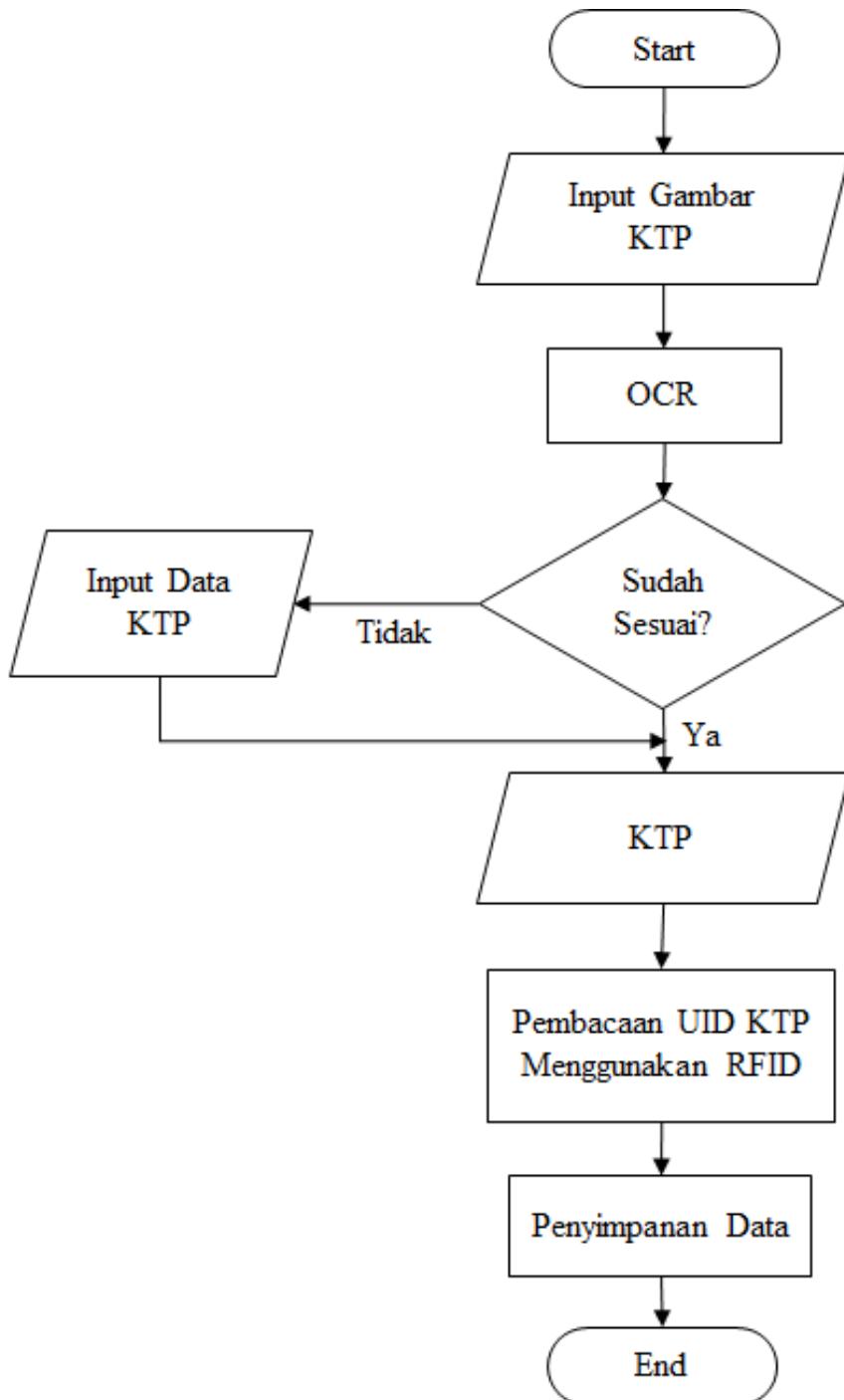
alur jalannya aplikasi yang akan dikembangkan nantinya. Diagram *flowchart* yang dibuat adalah *detail* alur dari fitur-fitur yang sudah ditentukan pada *use case diagram*.

Berdasarkan *use case diagram*, alur sistem dari proyek Tugas Akhir ini dibagi menjadi tiga, yaitu alur untuk menambahkan data baru, alur untuk menampilkan dan mengedit data yang sudah diinputkan sebelumnya, dan alur untuk mengautentikasi data. Oleh karena itu, dalam tahap perancangan ini terdapat tiga diagram *flowchart*, yaitu diagram *flowchart* untuk alur ketika ingin menambahkan data baru, diagram *flowchart* untuk alur ketika ingin menampilkan atau mengubah data yang sudah diinputkan sebelumnya, dan diagram *flowchart* untuk proses autentikasi data.

Diagram *flowchart* pertama yang dibuat adalah diagram *flowchart* untuk merancang alur ketika ingin menambahkan data baru. Diagram ini dapat dilihat pada Gambar 3.6. Alur pada diagram ini dimulai dengan menginputkan gambar kartu tanda penduduk (KTP). Kemudian gambar tersebut akan diproses dan dibaca oleh sistem menggunakan metode *Optical Character Recognition* (OCR). Setelah selesai, sistem akan mengeluarkan dan menampilkan data-data yang dibaca dari KTP.

User kemudian dapat memastikan terlebih dahulu kebenaran data-data yang sudah ditampilkan. *User* dapat mencocokkannya dengan gambar KTP yang sudah diinputkan sebelumnya. Jika data yang ditampilkan sudah benar, *user* dapat langsung menyimpan data dan melanjutkan ke tahap berikutnya, yaitu proses untuk membaca kode KTP menggunakan *Radio Frequency Identification* (RFID). Jika data yang ditampilkan tidak sesuai, *user* dapat langsung mengubah data, kemudian melanjutkan ke tahap penyimpanan kode KTP menggunakan RFID.

Setelah memastikan kebenaran data yang dibaca menggunakan metode OCR, *user* dapat lanjut ke proses pembacaan kode KTP menggunakan *Radio Frequency Identification* (RFID). Pada tahapan ini, KTP akan ditempelkan pada alat pembaca kode KTP menggunakan RFID. Jika alat sudah selesai membaca kode KTP, program akan menyimpan data yang dibaca dari KTP dan kode KTP yang dibaca menggunakan RFID ke *database*.

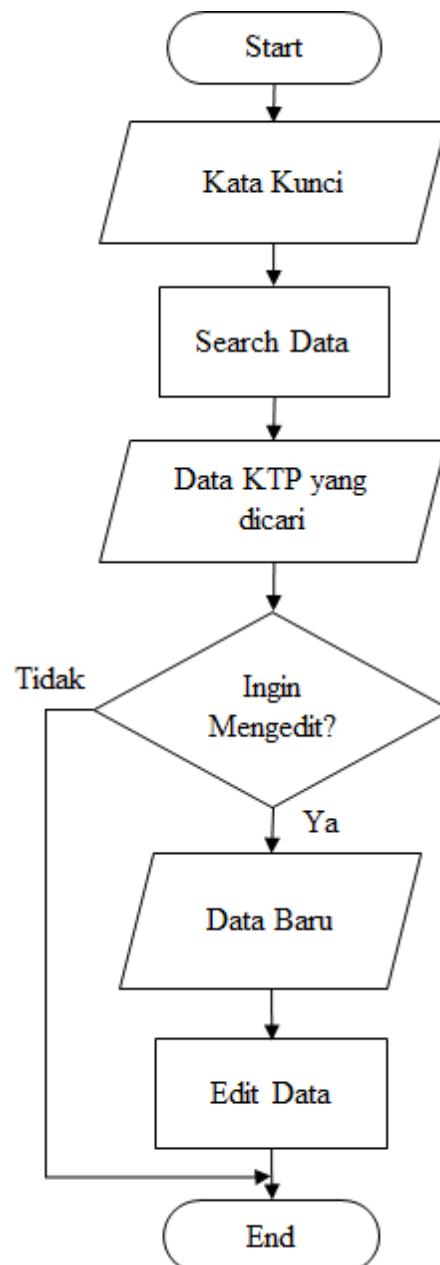


Gambar 3.6 *Flowchart* alur penambahan data baru

Berikutnya adalah merancang diagram *flowchart* untuk alur ketika ingin menampilkan atau mengubah data yang sudah diinputkan sebelumnya. Diagram *flowchart* untuk alur proses menampilkan atau mengubah data dapat dilihat pada Gambar 3.7. Alur untuk menampilkan dan mengubah data dijadikan satu karena sebelum mengubah data, data harus ditampilkan terlebih dahulu. Pada alur ini,

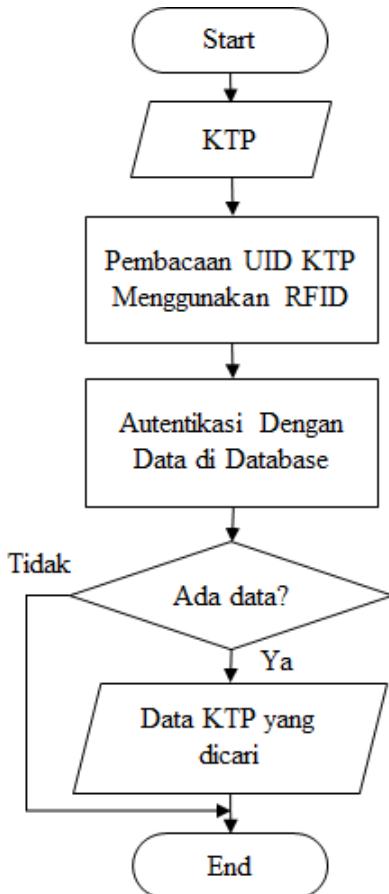
pertama *user* dapat memasukkan kata kunci. Kemudian sistem akan mencari data yang akan ditampilkan berdasarkan kata kunci tersebut. Setelah data ditemukan, sistem akan menampilkan data.

Setelah data ditampilkan, *user* memiliki pilihan untuk mengubah data atau tidak. Jika *user* ingin mengubah data, *user* dapat langsung mengubah data yang ditampilkan. Setelah *user* selesai mengubah data, sistem akan menyimpan data yang baru ke *database*. Jika *user* hanya ingin menampilkan data, setelah data ditampilkan maka alur akan berakhir.



Gambar 3.7 Flowchart alur view/edit data KTP

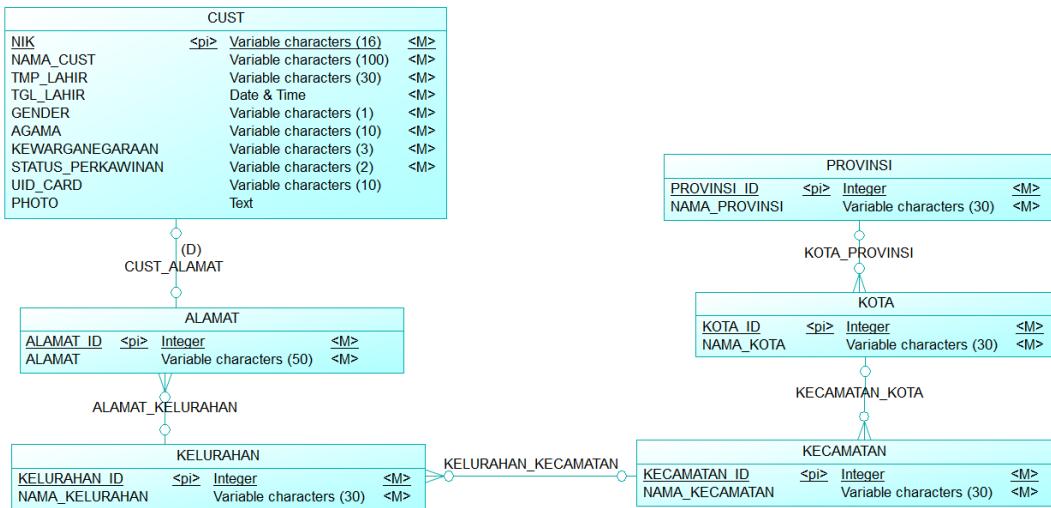
Diagram *flowchart* selanjutnya yang dirancang adalah diagram *flowchart* untuk mengautentikasi data KTP. Diagram *flowchart* ini dapat dilihat pada Gambar 3.8. Alur pada diagram ini dimulai dengan menempelkan KTP pada alat pembaca KTP menggunakan *Radio Frequency Identification* (RFID). Alat tersebut kemudian akan membaca kode dari KTP. Kode yang sudah dibaca akan dicocokkan dengan kode di *database* yang sudah dimasukkan sebelumnya.



Gambar 3.8 *Flowchart* alur autentikasi data KTP

c. Entity Relationship Diagram (ERD)

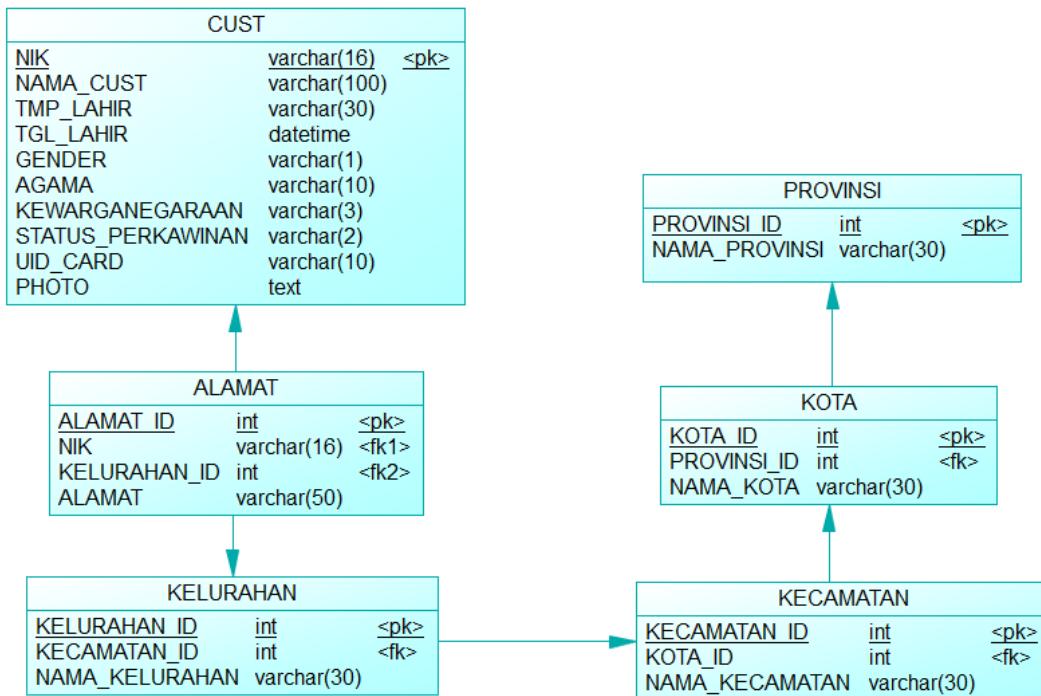
Setelah merancang alur melalui *flowchart* dan merancang *use case diagram*, tahapan berikutnya adalah merancang *Entity Relationship Diagram* (ERD). Dalam tahapan ini, ERD yang dirancang dibagi menjadi dua, yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). ERD berfungsi untuk membantu merancang dan menentukan tabel serta atributnya yang diperlukan untuk menyimpan data pribadi yang berasal dari kartu tanda penduduk (KTP). Selain itu, dengan adanya ERD ini dapat membantu merancang relasi antar tabel.



Gambar 3.9 Conceptual Data Model (CDM)

Rancangan diagram CDM dapat dilihat pada Gambar 3.9. Diagram CDM adalah diagram yang menceritakan mengenai relasi antar tabel yang akan dibuat nantinya. Relasi ini dapat berupa *one to one*, *one to many*, atau *many to one*.

Pada Gambar 3.9 dapat dilihat relasi antar tabel yang akan dibuat nantinya. Tabel CUST berelasi dengan tabel ALAMAT secara *one to one* yang berarti setiap pelanggan hanya memiliki satu alamat. Hal ini karena alamat yang akan disimpan di *database* hanyalah alamat yang tertera di KTP. Selain itu, tabel ALAMAT juga berelasi *many to one* dengan tabel KELURAHAN. Hal ini karena ada kemungkinan lebih dari satu orang yang memiliki alamat dengan kelurahan yang sama. Selain tabel CUST dan ALAMAT, ada juga tabel KELURAHAN yang berelasi *many to one* dengan tabel KECAMATAN. Relasi yang sama juga terjadi antara tabel KECAMATAN dengan tabel KOTA dan tabel KOTA dengan tabel PROVINSI. Hal ini sesuai dengan struktur administrasi kewilayahan di Indonesia. Dalam setiap provinsi terdapat banyak kota/kabupaten. Setiap kota/kabupaten memiliki lebih dari satu kecamatan. Setiap kecamatan memiliki lebih dari satu kelurahan. Tabel KELURAHAN, KECAMATAN, KOTA, dan PROVINSI hanya digunakan untuk menyimpan data-data yang akan digunakan untuk dropdown pada tampilan alamat, tabel ALAMAT akan langsung mengarahkan pada tabel KELURAHAN sehingga dari tabel KELURAHAN itu akan langsung diketahui KECAMATAN, KOTA, dan PROVINSI.



Gambar 3.10 Physical Data Model (PDM)

Rancangan diagram PDM dapat dilihat pada Gambar 3.10. Diagram PDM adalah diagram yang berfungsi untuk membantu menentukan atribut-atribut pada setiap tabel, serta tipe data dan *length*-nya. Selain itu, PDM juga membantu menentukan atribut *foreign key* (FK) yang menjadi penghubung antar tabel.

Pada Gambar 3.10 dapat dilihat terdapat enam tabel yaitu CUST, ALAMAT, KELURAHAN, KECAMATAN, KOTA, dan PROVINSI. Tabel CUST berfungsi untuk menyimpan data pribadi yang diambil dari kartu tanda penduduk (KTP). Tabel ALAMAT adalah tabel untuk menyimpan data alamat. Dalam tabel ALAMAT terdapat atribut NIK yang menjadi *foreign key* dari tabel CUST dan atribut KELURAHAN_ID yang menjadi *foreign key* dari tabel KELURAHAN. Tabel KELURAHAN memiliki atribut KECAMATAN_ID yang menjadi *foreign key* dari tabel KECAMATAN. Tabel KELURAHAN berfungsi untuk menyimpan nama-nama kelurahan, sedangkan tabel KECAMATAN berfungsi untuk menyimpan nama-nama kecamatan. Tabel KECAMATAN sendiri memiliki atribut KOTA_ID yang menjadi *foreign key* dari tabel KOTA. Tabel KOTA juga memiliki *foreign key* PROVINSI_ID yang berasal dari tabel PROVINSI. Tabel KOTA adalah tabel yang menyimpan nama-nama

kota/kabupaten, sedangkan tabel PROVINSI adalah tabel yang menyimpan nama-nama provinsi.

Setelah merancang *Physical Data Model* (PDM), rancangan tersebut digunakan untuk *generate script* SQL yang akan digunakan untuk membuat *database*. Rancangan dan keterangan mengenai kolom-kolom dari tabel yang akan dikembangkan nantinya dapat dilihat pada Tabel 3.1 hingga Tabel 3.6.

Tabel 3.1 Tabel Alamat

Nama Kolom	Tipe Data	Keterangan
ALAMAT_ID	int	not null, primary key
NIK	varchar(16)	not null, foreign key
ALAMAT	varchar(50)	not null
KELURAHAN_ID	int	not null

Tabel 3.2 Tabel Kelurahan

Nama Kolom	Tipe Data	Keterangan
KELURAHAN_ID	int	not null, primary key
KECAMATAN_ID	int(16)	not null, foreign key
NAMA_KELURAHAN	varchar(30)	not null

Tabel 3.3 Tabel Kecamatan

Nama Kolom	Tipe Data	Keterangan
KECAMATAN_ID	int	not null, primary key
KOTA_ID	int(16)	not null, foreign key
NAMA_KECAMATAN	varchar(30)	not null

Tabel 3.4 Tabel Kota

Nama Kolom	Tipe Data	Keterangan
KOTA_ID	int	not null, primary key
PROVINSI_ID	int(16)	not null, foreign key
NAMA_KOTA	varchar(30)	not null

Tabel 3.5 Tabel Provinsi

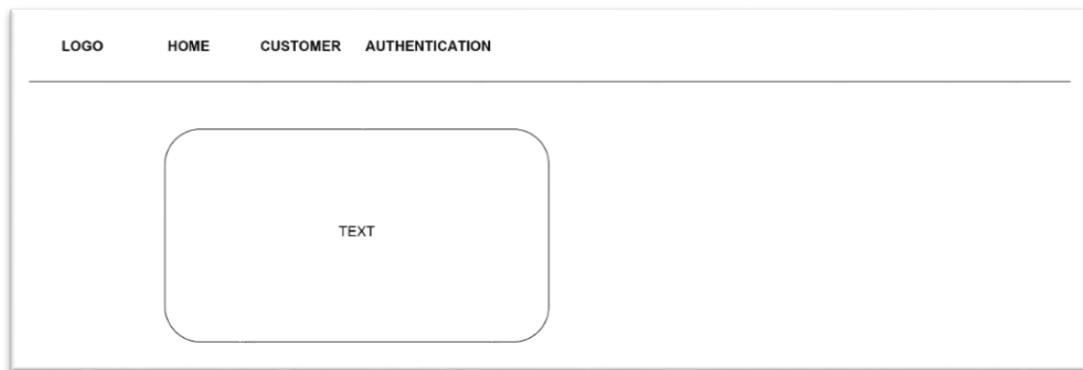
Nama Kolom	Tipe Data	Keterangan
PROVINSI_ID	int	not null, primary key
NAMA_PROVINSI	varchar(30)	not null

Tabel 3.6 Tabel Cust

Nama Kolom	Tipe Data	Keterangan
NIK	varchar(16)	not null, primary key
NAMA_CUST	varchar(100)	not null
TMP_LAHIR	varchar(50)	not null
TGL_LAHIR	Datetime	not null
GENDER	varchar(1)	not null
AGAMA	varchar(10)	not null
KEWARGANEGARAAN	varchar(3)	not null
STATUS_PERKAWINAN	varchar(2)	not null
UID_CARD	varchar(10)	null
PHOTO	text	null

d. *Mock Up*

Mock up adalah rancangan *interface* atau gambaran kasar dari aplikasi yang akan dibuat nantinya. Dengan adanya *mock up* dapat membantu merancang dan menentukan komponen-komponen yang perlu ditampilkan pada aplikasi. *Mock up* yang dirancang terdiri dari *mock up* menu Home, Customer, dan Authentication. Setiap *mock up* memiliki *navigation bar* yang berisi logo dan tiga menu yang akan dikembangkan yaitu *Home*, *Customer*, dan *Authentication*.



Gambar 3.11 *Mock up* menu *Home*

Mock up pertama yang dirancang adalah *mock up* untuk menu *Home* yang dapat dilihat pada Gambar 3.11. *Mock up* ini akan berperan seperti *cover*. *Mock up* untuk menu *home* hanya akan berisi tulisan pembuka atau pengenalan. *Mock up* ini akan menjadi tampilan pertama yang ditampilkan ketika memasuki *website*.

The mock-up shows a header with 'LOGO', 'HOME', 'CUSTOMER' (which is the active tab), and 'AUTHENTICATION'. Below this is a section titled 'CUSTOMER DATA' with an 'ADD' button. It includes input fields for 'ID NUMBER' and 'NAME', and a 'SEARCH' button. A table displays customer data with columns: NAME, NIK, GENDER, and ACTION. The first row contains 'TEXT' in all columns. The second row contains '...' in all columns. The 'ACTION' column for each row contains an 'ICON' button.

Gambar 3.12 *Mock up* menu *Customer*

Berikutnya adalah merancang *mock up* untuk menu *Customer*. *Mock up* ini dapat dilihat pada Gambar 3.12. *Mock up* ini adalah tampilan awal sebelum memasuki alur untuk menambah, menampilkan, atau mengedit data.

Pada *mock up* untuk menu *Customer* ini hanya terdapat satu *section* yaitu *section Customer Data*. Pada *mock up* ini terdapat tombol yang bertuliskan “ADD”. Tombol ini berfungsi untuk menambahkan data pribadi yang berasal dari kartu tanda penduduk (KTP). Tombol ini akan mengarahkan ke halaman *form customer*. Pada *mock up* ini juga terdapat tabel yang menampilkan data-data yang sudah diinputkan sebelumnya. Pada tabel ini akan tercantum beberapa informasi data pribadi yang diambil dari KTP, seperti nama, NIK, dan gender. Selain itu, pada tabel juga terdapat tombol pada kolom *action* yang berfungsi untuk mengarahkan ke halaman untuk menampilkan rincian data dan mengubah data. Di atas tabel terdapat komponen NIK dan Nama, serta tombol *search* sebagai tempat untuk menginputkan kata kunci ketika ingin mencari data tertentu. Tombol *search* berfungsi agar sistem mulai mencari data sesuai dengan kata kunci yang sudah

diinputkan. Jika sistem sudah menemukan data yang sesuai dengan kata kunci, data akan ditampilkan pada tabel.

The mock-up form for customer registration is structured as follows:

- Header:** LOGO, HOME, CUSTOMER, AUTHENTICATION.
- Photo Section:** Contains buttons for CAM, CHOOSE FILE, and UPLOAD, followed by a large rectangular area labeled "FOTO".
- Customer Main Information:** A section containing pairs of labels and input fields:
 - NAME (TEXT), GENDER (TEXT)
 - BIRTH PLACE (TEXT), BIRTH DATE (DATE PICKER)
 - NIK (TEXT), ID EXPIRED DATE (DATE PICKER)
 - RELIGION (TEXT), MARITAL STATUS (TEXT)
 - NATIONALITY (TEXT), PROFESSION NAME (TEXT)
- Address Information:** A section containing pairs of labels and input fields:
 - ADDRESS (TEXT AREA), KELURAHAN (DROPDOWN)
 - KECAMATAN (DROPDOWN), CITY (DROPDOWN)
 - RT / RW (TEXT) / (TEXT)
- Buttons:** CANCEL and NEXT.

Gambar 3.13 Mock up form customer

Pada Gambar 3.13 adalah *mock up form customer*. Pada *form customer* nantinya akan ada komponen untuk *upload* KTP atau mengaktifkan *webcam* untuk langsung memfoto KTP. Kemudian ada tombol *upload* untuk mulai memproses gambar KTP. KTP kemudian akan ditampilkan dan data yang sudah dibaca akan ditampilkan pada *section customer main information* dan *section address information*. *Section customer main information* berfungsi untuk menampilkan data pelanggan dari KTP selain alamat karena alamat akan ditampilkan pada *section address information*. KTP yang ditampilkan berfungsi untuk membantu *user* mengkonfirmasi kebenaran data yang sudah dibaca dan ditampilkan. Jika terdapat ketidaksesuaian, *user* dapat langsung mengedit data. Pada *mock up* ini juga terdapat dua tombol yaitu *cancel* dan *save*. Tombol *cancel* untuk mengarahkan ke halaman menu *customer* dan tombol *save* berfungsi untuk melanjutkan ke proses untuk membaca kode KTP menggunakan alat pembaca kode KTP dengan *Radio Frequency Identification (RFID)*.



Gambar 3.14 *Mock up add autentikasi KTP*

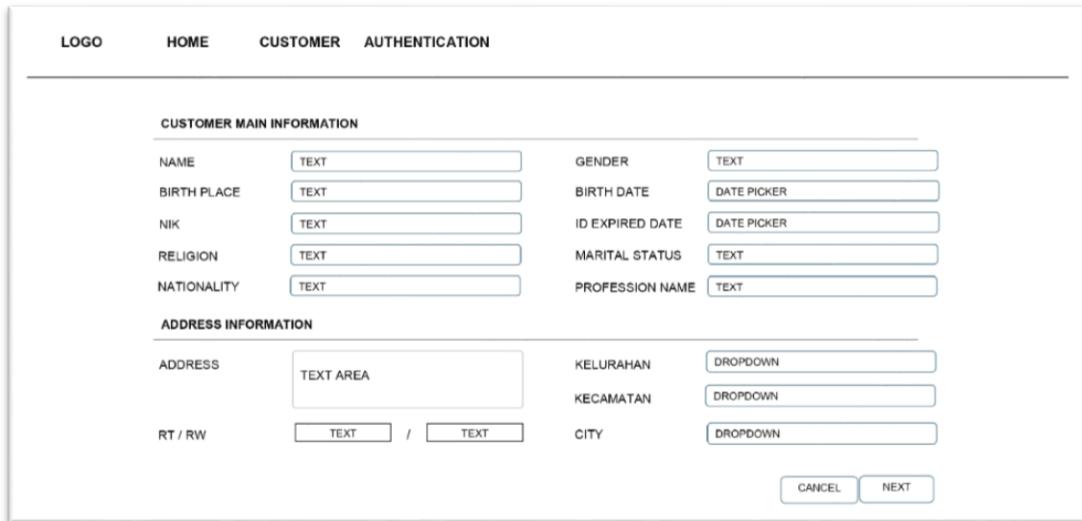
Mock up berikutnya adalah *mock up* untuk membaca kode KTP menggunakan alat pembaca dengan RFID seperti pada Gambar 3.14. *Mock up* ini dapat diakses setelah selesai proses pembacaan data KTP menggunakan OCR. Pada *mock up* ini terdapat tulisan konfirmasi untuk memulai proses pembacaan kode KTP. Selain itu, pada *mock up* ini terdapat dua tombol yaitu *start* dan *cancel*. Tombol *start* adalah tombol untuk memulai proses pembacaan kode KTP menggunakan alat pembaca kode dengan RFID. Tombol *cancel* akan mengarahkan kembali ke halaman *form customer*.

A screenshot of a web application interface titled 'CUSTOMER MAIN INFORMATION'. At the top, there are navigation links: LOGO, HOME, CUSTOMER, and AUTHENTICATION. Below the navigation, there is a section titled 'CUSTOMER MAIN INFORMATION' containing various input fields: NAME (TEXT), BIRTH PLACE (TEXT), NIK (TEXT), RELIGION (TEXT), NATIONALITY (TEXT), GENDER (TEXT), BIRTH DATE (DATE PICKER), ID EXPIRED DATE (DATE PICKER), MARITAL STATUS (TEXT), and PROFESSION NAME (TEXT). There is also an 'EDIT' button. Below this section is another titled 'ADDRESS INFORMATION' containing: ADDRESS (TEXT AREA), RT / RW (TEXT), KELURAHAN (DROPDOWN), KECAMATAN (DROPDOWN), and CITY (DROPDOWN).

Gambar 3.15 *Mock up view data pribadi customer*

Pada Gambar 3.15 adalah *mock up* untuk menampilkan data pribadi yang sudah diinputkan sebelumnya. *Mock up* ini mirip seperti *mock up form customer*. Bedanya adalah pada *mock up* untuk *view* ini tidak terdapat tombol untuk membaca data pribadi KTP menggunakan metode OCR. Selain itu, pada mode *view*, semua

kolom inputan *disable* dan tidak dapat diubah. Pada *mock up view* ini terdapat tombol *edit* untuk *enable* semua inputan sehingga *user* dapat mengubah data.



The image shows a wireframe mock-up of a web application interface for editing customer personal information. At the top, there is a navigation bar with tabs: LOGO, HOME, CUSTOMER (which is the active tab), and AUTHENTICATION. Below the navigation bar, the main content area is divided into two sections: 'CUSTOMER MAIN INFORMATION' and 'ADDRESS INFORMATION'. In the 'CUSTOMER MAIN INFORMATION' section, there are eight input fields arranged in two rows of four. The first row contains 'NAME' (text), 'GENDER' (text), 'BIRTH PLACE' (text), and 'BIRTH DATE' (date picker). The second row contains 'NIK' (text), 'ID EXPIRED DATE' (date picker), 'RELIGION' (text), 'MARITAL STATUS' (text), 'NATIONALITY' (text), and 'PROFESSION NAME' (text). In the 'ADDRESS INFORMATION' section, there are five input fields arranged in two rows of three. The first row contains 'ADDRESS' (text area) and 'KELURAHAN' (dropdown). The second row contains 'RT / RW' (text), 'KECAMATAN' (dropdown), and 'CITY' (dropdown). At the bottom right of the form, there are two buttons: 'CANCEL' and 'NEXT'.

Gambar 3.16 *Mock up edit data pribadi customer*

Mock up pada Gambar 3.16 adalah *mock up* ketika dalam mode edit. *Mock up* ini mirip seperti *mock up view*. Bedanya adalah pada *mock up* ini kolom inputan *enable* dan tidak ada tombol *edit*. Pada *mock up* ini terdapat dua tombol yaitu tombol *cancel* dan tombol *save*. Tombol *cancel* berfungsi untuk mengarahkan ke halaman menu *customer* tanpa menyimpan data. Tombol *save* berfungsi untuk menyimpan data yang diubah, kemudian mengarahkan ke halaman menu *customer*.

3.5 Pengujian

Pengujian yang akan dilakukan nantinya dibagi menjadi tiga, yaitu pengujian metode *Optical Character Recognition* (OCR), pengujian autentikasi, dan pengujian aplikasi *web*. Pengujian metode OCR terdiri dari pengujian metode *Pytesseract* dan pengujian metode *Template Matching*. Pengujian *web* terdiri dari pengujian pada fitur-fitur yang akan dikembangkan nantinya.

1. Pengujian Metode

Pengujian metode akan dilakukan menggunakan dua kamera, yaitu kamera *smartphone* dan kamera laptop, serta tiga jenis pencahayaan, yaitu gelap (kurang dari 100 LUX), terang (100 – 200 LUX), dan terang sekali (lebih dari 200 LUX). Pengujian menggunakan kamera *smartphone* dilakukan pada masing-masing

pencahayaan yaitu dengan pencahayaan gelap pada 10 LUX, pencahayaan terang pada 122 LUX, dan pencahayaan terang sekali pada 390 LUX. Pengujian menggunakan kamera laptop dilakukan ada pencahayaan gelap pada 59 LUX.

Akurasi dari hasil pembacaan menggunakan metode *Pytesseract* dan *Template Matching* akan dihitung menggunakan rumus 3-1. 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 (Aprillian, 2019).

$$\text{Keakuratan}(\%) = \frac{\text{jumlah karakter terbaca benar}}{\text{jumlah karakter pada KTP}} \times 100\% \quad (3-1)$$

Hasil akurasi akan dihitung untuk masing-masing komponen KTP yaitu NIK, nama, tempat lahir, tanggal lahir, gender, alamat, RT, RW, kelurahan, kecamatan, agama, status perkawinan, pekerjaan, kewarganegaraan, dan waktu berlakunya KTP. Kemudian dicari rata-rata dari hasil pembacaan setiap KTP dengan rumus 3-2, yaitu semua hasil akurasi dijumlahkan dan dibagi jumlah komponen pada KTP yang dilakukan pengujian (Aprillian, 2019).

$$\text{Rata - rata}(\%) = \frac{\text{total keakuratan dari masing - masing komponen}}{\text{jumlah komponen pada KTP}} \quad (3-2)$$

2. Pengujian Autentikasi

Pengujian autentikasi dilakukan dengan membaca uid KTP menggunakan *Radio Frequency Identification* (RFID). Hasil dari pengujian autentikasi adalah apakah uid KTP akan dapat dibaca oleh RFID atau tidak. Pengujian dilakukan menggunakan 10 KTP yang sama dengan yang digunakan untuk pengujian metode OCR.

3. Pengujian Aplikasi Web

Pengujian aplikasi adalah fitur yang dikembangkan. Pengujian dilakukan dengan menjalankan semua fitur dan mengecek apakah fitur yang dikembangkan

sudah berjalan dengan lancar. Fitur yang dikembangkan antara lain seperti pada Tabel 3.7.

Tabel 3.7 Tabel Pengujian Aplikasi *Web*

No.	Aspek Pengujian
1.	Pembacaan KTP (upload file) menggunakan OCR
2.	Pembacaan KTP (foto langsung) menggunakan OCR
3.	Penambahan data
4.	<i>View</i> data
5.	Pengubahan data
6.	Pencarian data
7.	Autentikasi data

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Uji Coba

Pengujian dilakukan sesuai dengan skenario yang sudah ditentukan sebelumnya. Pengujian yang dilakukan dibagi menjadi pengujian metode, pengujian autentikasi, dan pengujian *web*. Pengujian metode terdiri dari pengujian metode *Pytesseract* dan *Template Matching*. Pengujian *web* dibagi menjadi pengujian penambahan data, pengujian pengubahan data, dan pengujian pencarian data.

1. Pengujian Metode

Pengujian metode dilakukan dengan menguji metode *Pytesseract* dan *Template Matching*. Pengujian dilakukan menggunakan 10 KTP berbeda dengan empat situasi berbeda yaitu pencahayaan gelap (kurang dari 100 LUX), pencahayaan terang (100 – 200 LUX), pencahayaan terang sekali (> 200 LUX), dan menggunakan kamera laptop. KTP yang ditampilkan pada laporan ini diambil dari gurusiana (Shaleng, 2021).

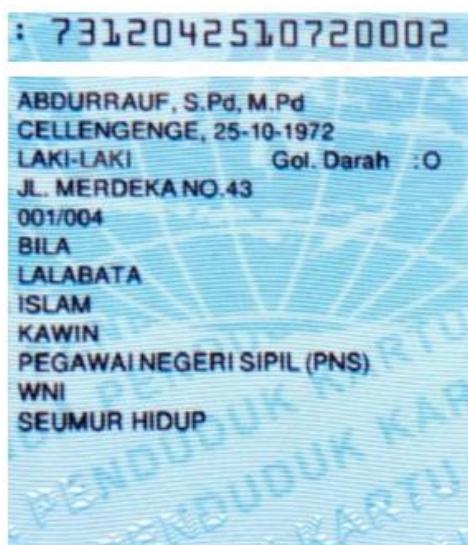
a. Pengujian Metode *Pytesseract*

Pada metode *Pytesseract*, pertama akan dilakukan pemotongan gambar KTP menjadi gambar NIK dan gambar data selain NIK. Ukuran pemotongan gambar ini berbeda antara gambar yang diambil menggunakan kamera *smartphone* dan kamera laptop. Hal ini karena pada gambar yang diambil menggunakan kamera *smartphone*, gambar sudah dilakukan pemotongan sebelumnya sehingga hanya gambar KTP saja. Selain itu, ukuran gambar dibedakan menjadi dua yaitu ukuran gambar NIK untuk mengambil 16 digit NIK dan data selain NIK untuk mengambil data selain NIK dari nama hingga kewarganegaraan. Contoh *source code* pemotongan gambar KTP dapat dilihat pada Gambar 4.1. *Source code* untuk pemotongan gambar yang diambil menggunakan kamera *smartphone* dapat dilihat pada Gambar 4.1 baris 2–7. Ukuran untuk pemotongan gambar NIK adalah 14% hingga 23% secara vertikal dan 20% hingga 70% secara horizontal, sedangkan ukuran pemotongan untuk data selain NIK adalah 22% hingga 100% secara vertikal dan 23% hingga 70% secara horizontal. *Source code* untuk pemotongan gambar yang diambil menggunakan kamera laptop

dapat dilihat pada Gambar 4.1 baris 9–13. Ukuran pemotongan untuk gambar NIK adalah 21% hingga 30% secara vertikal dan 20% hingga 70% secara horizontal, sedangkan untuk gambar selain NIK adalah 29% hingga 100% secara vertikal dan 22% hingga 70% secara horizontal. Contoh hasil pemotongan dapat dilihat pada Gambar 4.2. Kemudian pada kedua gambar dilakukan *image processing* seperti *grayscale* dan *threshold*, seperti pada Gambar 4.3 dan Gambar 4.4.

```
1.     rows, cols, _ = img.shape
2.     x = int(20/100*cols)
3.     y = int(14/100*rows)
4.     x1 = int(70/100*cols)
5.     y1_nik = int(23/100*rows)
6.     y_data = int(22/100*rows)
7.     x_data = int(23/100*cols)
8.
9.     if rows/cols == 0.75:
10.         x_data = int(22 / 100 * cols)
11.         y = int(21/100*rows)
12.         y_data = int(29 / 100 * rows)
13.         y1_nik = int(30 / 100 * rows)
14.
15.     img_nik = img[y:y1_nik, x:x1]
16.     img = img[y_data:rows, x_data:x1]
```

Gambar 4.1 *Source code* pemotongan gambar KTP



Gambar 4.2 Contoh hasil pemotongan gambar KTP

: 7312042510720002

Gambar Asli



: 7312042510720002

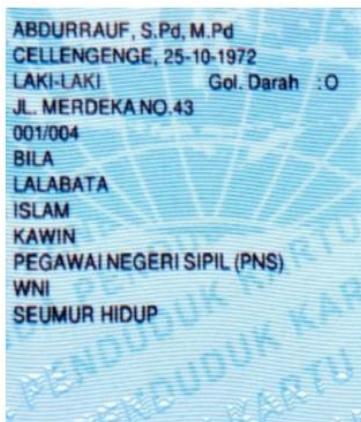
Grayscale



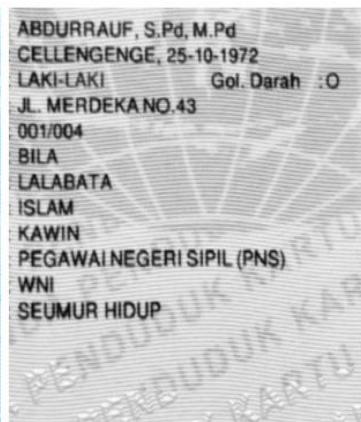
: 7312042510720002

Threshold

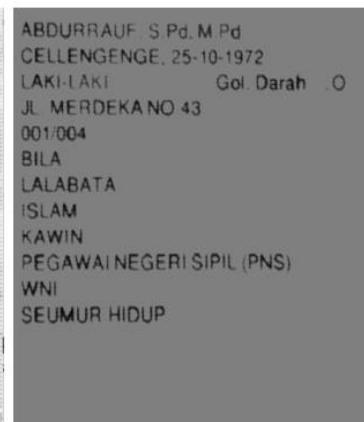
Gambar 4.3 Contoh hasil *image processing* NIK KTP



Gambar Asli



Grayscale



Threshold

Gambar 4.4 Contoh hasil *image processing* data KTP selain NIK

Hasil dari *image processing* tersebut yang akan dibaca oleh *Pytesseract*.

Contoh *source code* yang digunakan untuk proses pembacaan *Pytesseract* terdapat pada Gambar 4.5. Pada Gambar 4.5 baris satu dapat dilihat aplikasi *tesseract* dipanggil terlebih dahulu. Hal ini dilakukan karena *tesseract* belum didaftarkan pada *environment* laptop. Gambar 4.5 pada baris dua adalah pembacaan *Pytesseract* menggunakan fungsi `image_to_string`. Hasil pembacaan NIK akan langsung

dimasukkan dalam suatu array, kemudian hasil pembacaan data selain NIK dipisah-pisah terlebih dahulu dan dimasukkan ke dalam array yang sama. Hal ini bertujuan agar mempermudah menentukan masing-masing data dalam KTP berdasarkan urutannya. Selanjutnya dilakukan *string processing*, seperti membagi array menjadi data KTP seperti NIK, nama, dan lain-lain dan mengolah string agar hasil akhir lebih akurat.

```

1.         pytesseract.pytesseract.tesseract_cmd =
r'C:\Program Files\Tesseract-OCR\tesseract.exe'
2.         result_nik =
pytesseract.image_to_string(threshed_nik, lang="ind")

```

Gambar 4.5 Contoh *source code* pembacaan *Pytesseract*

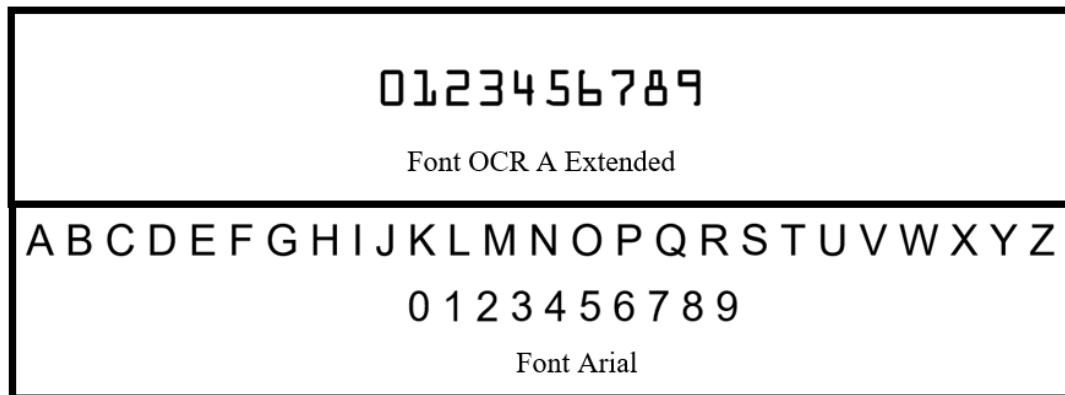
Hasil pengujian untuk metode *Pytesseract* dapat dilihat pada Tabel 4.1. Pada Tabel 4.1 dapat dilihat terdapat empat kondisi, yaitu gelap, terang, terang sekali, dan menggunakan kamera laptop. Untuk kondisi gelap, terang, dan terang sekali, gambar diambil menggunakan kamera *smartphone*. Untuk kondisi menggunakan kamera laptop, gambar diambil pada tingkat kecerahan 59 LUX. Hasil pada Tabel 4.1 ini adalah rata-rata presentase akurasi dari setiap KTP pada setiap kondisinya.

Tabel 4.1 Hasil Pengujian Metode *Pytesseract*

	Gelap (<100 LUX) (%)	Terang (100 – 200 LUX) (%)	Terang Sekali (200> LUX) (%)	Kamera Laptop (59 LUX) (%)
KTP 1	99,58	97,36	98,92	97,70
KTP 2	96,11	99,58	100	99,70
KTP 3	99,17	97,74	98,92	97,19
KTP 4	98,75	98,92	98,33	99,78
KTP 5	99,17	99,58	100	100
KTP 6	93,95	98,29	98,45	92,86
KTP 7	96,11	97,47	99,49	98,65
KTP 8	99,17	99,58	99,17	97,45
KTP 9	99,17	95,86	97,08	97,73
KTP 10	99,05	99,58	99,17	98,37
Rata-Rata	98,02	98,40	98,95	97,94

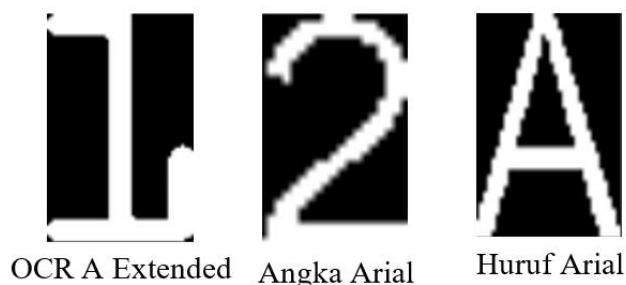
b. Pengujian Metode *Template Matching*

Pada metode *Template Matching*, pertama akan disiapkan *template* untuk pembacaan KTP. *Template* yang disiapkan terdiri dari dibagi menjadi 2 yaitu *template* menggunakan jenis tulisan OCR A Extended yang terdiri dari angka 0 – 9 dan jenis tulisan Arial yang terdiri dari angka 0 – 9 dan huruf kapital A – Z. Font yang akan digunakan dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Template* dan jenis tulisan yang digunakan

Pada *template* kemudian dilakukan segmentasi karakter untuk memisahkan setiap karakter pada *template*. Hasil dari segmentasi karakter ini selanjutnya dimasukkan ke dalam array untuk menjadi *template* dalam proses metode *Template Matching*. Contoh hasil segmentasi karakter *template* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Contoh hasil segmentasi karakter *template*

Setelah selesai menyiapkan *template*, berikutnya adalah proses metode *Template Matching*. Sama seperti pada metode *Pytesseract*, gambar KTP pertama

dibagi menjadi dua yaitu gambar NIK dan gambar data selain NIK. Ukuran pemotongan gambar untuk metode *Template Matching* sama seperti yang digunakan untuk metode *Pytesseract*. Kemudian pada kedua gambar tersebut dilakukan *image processing* seperti *grayscale* dan *threshold*. Selanjutnya pada gambar NIK hasil *image processing* langsung dilakukan segmentasi karakter. Contoh hasil segmentasi karakter NIK dapat dilihat pada Gambar 4.8. Hasil dari segmentasi karakter ini kemudian dilakukan metode *Template Matching* yaitu mencocokkan dengan *template* yang sudah disiapkan sebelumnya. Contoh *source code* untuk proses *Template Matching* dapat dilihat pada Gambar 4.9. Pada Gambar 4.9 baris 3 adalah proses pemanggilan fungsi *Template Matching*. Kemudian pada Gambar 4.9 baris 4 akan dicari nilai min dan max dari hasilnya. Selanjutnya pada Gambar 4.9 baris 6 akan dicari nilai max dari semua kemungkinan pembacaannya. Selanjutnya hasil pencocokannya akan dimasukkan ke dalam suatu array.



Gambar 4.8 Contoh hasil segmentasi karakter NIK KTP

```
1.         scores = []
2.         for (digit, digitROI) in digitsNik.items():
3.             result = cv2.matchTemplate(roi, digitROI,
cv2.TM_CCOEFF)
4.             (_, score, _, _) = cv2.minMaxLoc(result)
5.             scores.append(score)
6.         groupOutput.append((np.argmax(scores)))
```

Gambar 4.9 Contoh *source code* untuk proses *Template Matching*

Untuk gambar data selain NIK, setelah *image processing* dilakukan metode *run-length smearing algorithm* (RLSA) untuk mengidentifikasi baris-baris yang ada, seperti pada Gambar 4.10. Kemudian dilakukan segmentasi baris untuk memisahkan setiap barisnya. Contoh hasil segmentasi baris dapat dilihat pada Gambar 4.11.



Gambar 4.10 Contoh hasil RLSA untuk mengidentifikasi baris



Gambar 4.11 Contoh hasil segmentasi baris

Pada setiap hasil dari segmentasi baris kemudian dilakukan metode RLSA lagi untuk mengidentifikasi setiap kata pada baris tersebut dilanjutkan dengan segmentasi kata. Contoh hasil RLSA untuk mengidentifikasi kata dapat dilihat pada Gambar 4.12, sedangkan contoh hasil segmentasi kata dapat dilihat pada Gambar 4.13. Selanjutnya dilakukan segmentasi karakter pada setiap kata. Hasil dari segmentasi karakter ini berikutnya dilakukan metode *Template Matching* yaitu mencocokkan dengan *template*. Hasil dari pencocokan karakter ini akan digabungkan menjadi kata dengan dipisahkan oleh spasi sesuai dengan barisnya. Hasil penggabungan ini kemudian dimasukkan ke dalam array yang sebelumnya sudah terdapat hasil pembacaan dari NIK. Selanjutnya setiap bagian array akan dipisahkan menjadi data-data sesuai urutannya. Berikutnya dilakukan *string processing* untuk meningkatkan keakuratan hasil.



Gambar 4.12 Contoh hasil RLSA untuk mengidentifikasi kata



Gambar 4.13 Contoh hasil segmentasi kata

Hasil pengujian untuk metode *Template Matching* dapat dilihat pada Tabel 4.2. Untuk kondisi gelap, terang, dan terang sekali diambil menggunakan kamera *smartphone*, dan kondisi menggunakan kamera laptop diambil pada tingkat kecerahan 59 LUX. Hasil pada Tabel 4.2 ini adalah rata-rata persentase akurasi dari setiap KTP pada setiap kondisinya.

Tabel 4.2 Hasil Pengujian Metode *Template Matching*

	Gelap (<100 LUX) (%)	Terang (100 – 200 LUX) (%)	Terang Sekali (200> LUX) (%)	Kamera Laptop (59 LUX) (%)
KTP 1	54,12	80,41	70,20	44,57
KTP 2	73,92	82,86	71,52	56,66
KTP 3	89,87	87,56	88,30	75,83
KTP 4	65,13	68,26	82,31	56,54
KTP 5	90,67	88,72	86,67	64,15
KTP 6	29,32	43,06	54,04	42,99
KTP 7	55,18	64,36	74,20	50,81
KTP 8	64,05	78,26	69,13	59,08
KTP 9	51,61	66,40	75,24	58,09
KTP 10	55,19	76,83	80,59	66,54
Rata-Rata	62,91	73,67	75,22	57,53

c. Metode OCR yang Digunakan

Berdasarkan hasil akurasi pada metode *Pytesseract* di Tabel 4.1 dan hasil akurasi pada metode *Template Matching* di Tabel 4.2 dapat dilihat bahwa akurasi metode *Pytesseract* lebih baik dibandingkan akurasi metode *Template Matching*. Oleh karena itu, metode yang digunakan untuk OCR pada aplikasi adalah metode *Pytesseract*. Pada proses *string processing* metode *Pytesseract* juga akan ditambahkan algoritma *levenshtein distance* pada *string processing* untuk komponen KTP gender, agama, status perkawinan, dan kewarganegaraan. Hasil dari algoritma *levenshtein distance* berupa jarak beda antara string hasil pembacaan dan string *template* pada array yang sudah disiapkan. Gambar 4.14 baris 1–29 adalah *source code* dari fungsi algoritma *levenshtein distance*. Fungsi ini akan memberikan balikan berupa jarak beda seperti

pada Gambar 4.15. Hasil ini yang kemudian digunakan untuk fitur *autocorrect* pada gender, agama, status perkawinan, dan kewarganegaraan.

```
1.  def levenshteinDistance(self, str1, str2):
2.      distances = np.zeros((len(str1) + 1, len(str2) + 1))
3.      for t1 in range(len(str1) + 1):
4.          distances[t1][0] = t1
5.
6.          for t2 in range(len(str2) + 1):
7.              distances[0][t2] = t2
8.
9.      a = 0
10.     b = 0
11.     c = 0
12.
13.     for t1 in range(1, len(str1) + 1):
14.         for t2 in range(1, len(str2) + 1):
15.             if (str1[t1-1] == str2[t2-1]):
16.                 distances[t1][t2] = distances[t1 - 1][t2
- 1]
17.             else:
18.                 a = distances[t1][t2 - 1]
19.                 b = distances[t1 - 1][t2]
20.                 c = distances[t1 - 1][t2 - 1]
21.
22.                 if (a <= b and a <= c):
23.                     distances[t1][t2] = a + 1
24.                 elif (b <= a and b <= c):
25.                     distances[t1][t2] = b + 1
26.                 else:
27.                     distances[t1][t2] = c + 1
28.
29.     return distances[len(str1)][len(str2)]
```

Gambar 4.14 Source code algoritma *levenshtein distance*

[3.0, 9.0, 9.0, 8.0]

Gambar 4.15 Contoh hasil balikan dari fungsi algoritma *levenshtein distance* yang sudah dijadikan satu array

2. Pengujian Autentikasi

Pengujian autentikasi dilakukan dengan membaca uid dari 10 KTP menggunakan *Radio Frequency Identification* (RFID). KTP cukup ditempelkan pada RFID yang sudah disambungkan pada laptop menggunakan USB. Hasil dari pengujian autentikasi dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Autentikasi

Pengujian	Hasil
KTP 1	Berhasil
KTP 2	Berhasil
KTP 3	Berhasil
KTP 4	Berhasil
KTP 5	Berhasil
KTP 6	Berhasil
KTP 7	Berhasil
KTP 8	Berhasil
KTP 9	Berhasil
KTP 10	Berhasil

3. Pengujian Web

Pengujian *web* dilakukan setelah pengujian metode selesai dilakukan. Hal ini karena pada pengujian *web* juga terdapat pengujian metode *Optical Character Recognition* (OCR) untuk pembacaan kartu tanda penduduk (KTP) pada tahap penambahan data pelanggan.

Pengujian *web* dilakukan dengan menguji alur jalannya aplikasi *web* yang sudah dikembangkan. Selain itu, pengujian ini dilakukan untuk menguji fitur-fitur yang sudah dikembangkan. Fitur-fitur yang sudah dikembangkan yaitu pembacaan KTP menggunakan RFID baik secara upload maupun difoto secara langsung menggunakan kamera laptop, penambahan data, pengubahan data, pencarian data, *view* data, dan autentikasi data.

Pada *web* terdapat tiga menu yang terletak pada *navigation bar* di atas, yaitu *Home*, *Customer*, dan *Authentication*. *Navigation bar* ini akan selalu ditampilkan untuk mempermudah *user* ketika ingin berpindah menu. Ketika membuka web pertama kali, yang ditampilkan adalah tampilan pada menu *Home* seperti pada Gambar 4.16.



Gambar 4.16 Menu *Home*

Pengujian yang pertama dilakukan adalah pengujian penambahan data. Pengujian ini dilakukan pada menu *Customer*. Gambar 4.17 adalah tampilan saat pertama kali memasuki menu *Customer*. Pada tampilan awal menu *Customer* dapat dilihat terdapat tombol “Add” untuk masuk ke halaman *form* data, dua *field* yaitu NIK dan Nama untuk membantu memfilter data berdasarkan NIK dan/atau nama pelanggan, tombol “Reset” untuk menghapus field NIK dan Nama agar kembali kosong, dan tombol “Search” untuk mencari data berdasarkan field NIK dan Nama.

Dari menu *Customer* klik tombol “Add” di kanan atas untuk masuk ke *form* penambahan data seperti pada Gambar 4.18. Pada halaman *form* penambahan data, di paling atas terdapat tiga tombol yaitu “Camera” untuk menyalakan *webcam*, “Choose File” untuk memilih file gambar/KTP yang akan dilakukan OCR, dan “Upload” untuk mulai menjalankan OCR. Di bawahnya terdapat *subsection* Customer Main Information yang berisi NIK, nama, tempat lahir, tanggal lahir, gender, agama, status

perkawinan, pekerjaan, kewarganegaraan, tanggal berakhirnya masa berlaku KTP, dan *subsection* Address Information yang berisi alamat, RT, RW, kelurahan, kecamatan, kota, dan provinsi.

The screenshot shows a web-based application interface for managing customer data. At the top, there is a navigation bar with links for HOME, CUSTOMER, and AUTHENTICATION. Below the navigation bar, the title "CUSTOMER DATA" is displayed, followed by a green "ADD" button. The main area contains two input fields: "NIK" and "NAME". Below these fields are two buttons: "RESET" (yellow) and "SEARCH" (green). The entire interface is enclosed in a light gray border.

Gambar 4.17 Tampilan awal menu *Customer*

The screenshot shows a detailed form for adding customer information. At the top, there are three buttons: "CAMERA" (yellow), "CHOOSE FILE" (yellow), and "UPLOAD" (green). Below these buttons, the title "CUSTOMER MAIN INFORMATION" is shown. The form is divided into several sections: "NAME" (text input), "BIRTH PLACE" (text input), "NIK" (text input), "RELIGION" (text input), "NATIONALITY" (text input), "GENDER" (dropdown menu), "BIRTH DATE" (text input), "ID EXPIRED DATE" (text input), "MARITAL STATUS" (dropdown menu), and "PROFESSION NAME" (text input). The next section is "ADDRESS INFORMATION", which includes "ADDRESS" (text input), "PROVINSI" (dropdown menu), "KOTA" (dropdown menu), "KECAMATAN" (dropdown menu), "RT / RW" (text input), "KELURAHAN" (dropdown menu), and "NEXT" (green) and "CANCEL" (yellow) buttons at the bottom right.

Gambar 4.18 Form penambahan data

Penambahan data dapat dilakukan dengan mengisi *form* atau dengan melakukan pembacaan KTP oleh OCR. Pada pengujian ini, penambahan data dilakukan dengan pembacaan KTP oleh OCR dan pembacaan uid KTP dengan RFID.

Pembacaan KTP oleh OCR dapat dilakukan dengan memfoto KTP secara langsung dengan mengklik tombol “Camera” atau mengupload pada tombol “Choose File”. Setelah gambar siap dapat langsung melakukan proses OCR dengan mengklik tombol “Upload”. Ketika proses OCR selesai akan ada proses ekstraksi foto KTP. Hasil dari pembacaan OCR dan ekstraksi foto akan dimunculkan pada tampilan seperti dapat dilihat pada Gambar 4.19. Sebelum masuk ke tahap pembacaan uid KTP oleh RFID dengan mengklik tombol “Next”, *user* dapat memastikan terlebih dahulu mengecek kebenaran hasil yang ditampilkan pada *form*.

HOME CUSTOMER AUTHENTICATION

CAMERA CHOOSE FILE UPLOAD

PROVINSI SULAWESI SELATAN
KABUPATEN SOPPENG

NIK : 7312042510720002

Name : ABDURRAUF, S.Pd.M.Pd
Tempat Lahir : CELLENGENGE, 01/01/1972
Jenis Kelamin : Laki-laki
Alamat : J. MERDEKANO 43, Kot. Derah - O
RT/RW : 001/004
Kota : Soppeng
Kecamatan : LALABATA
Agama : ISLAM
Status Pendidikan : YAYASAN
Pekerjaan : PEGAWAI NEGERI SIPIL (PNS)
Tempat Tinggal : Soppeng
Berikut Hingga : SEUMUR HIDUP

KTP5.JPG(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)

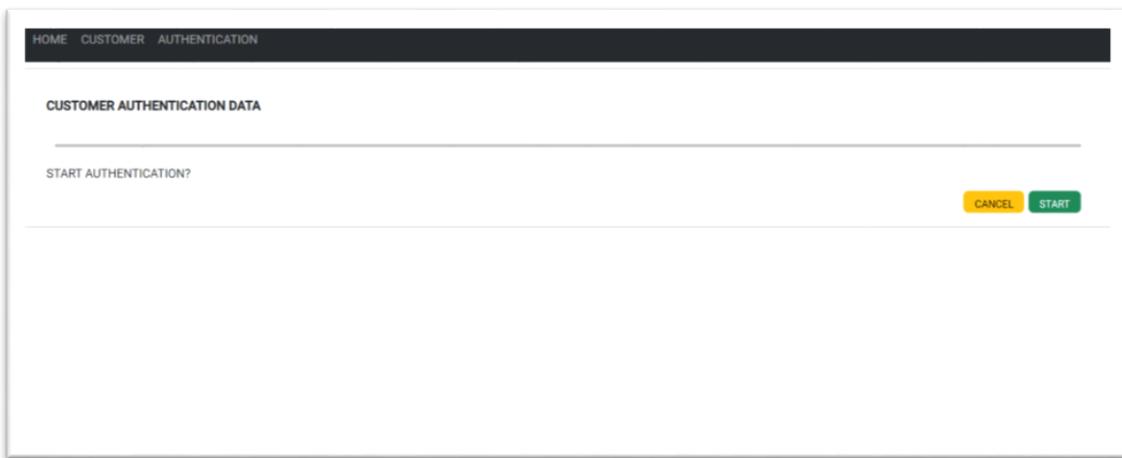
ADDRESS INFORMATION

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

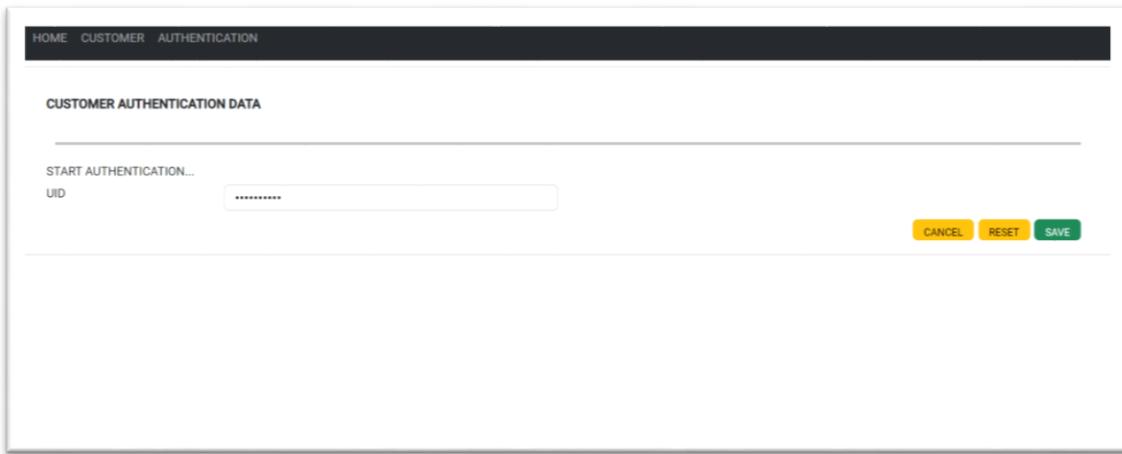
CANCEL NEXT

Gamabr 4.19 Contoh hasil pembacaan OCR

Pembacaan uid KTP oleh RFID dilakukan langsung setelah *user* memastikan kecocokan data dan menekan tombol “Next”. Proses pembacaan uid dimulai dengan menekan tombol “Start” di halaman *Customer Authentication Data* seperti pada Gambar 4.20. Kemudian akan muncul *field* UID. *User* dapat mengarahkan kursor ke *field* UID dan menempelkan KTP ke RFID. RFID akan secara otomatis membaca uid KTP dan menampilkannya dalam simbol seperti pada Gambar 4.21. Selanjutnya *user* dapat menekan tombol “Save” untuk menyimpan data ke *database*. Data gender dan status perkawinan akan tersimpan dalam bentuk kode di *database*, contohnya kode P untuk gender perempuan.



Gambar 4.20 Tampilan awal *Customer Authentication Data*



4.21 Contoh tampilan setelah pembacaan uid KTP oleh RFID

Setelah data tersimpan di *database*, *user* akan langsung kembali ke tampilan awal menu *Customer* seperti pada Gambar 4.17. Jika dilakukan pencarian, data yang sudah tersimpan di *database* akan ditampilkan dalam tabel yang berisi nama, NIK, gender, dan *icon* untuk edit. *User* juga dapat memfilter data berdasarkan nik dan/atau nama seperti pada Gambar 4.22. Untuk mengubah data dapat langsung mengklik *icon* di kolom *Action* dan kembali masuk ke halaman *form*.

The screenshot shows a web-based application interface for managing customer data. At the top, there is a navigation bar with links for HOME, CUSTOMER, and AUTHENTICATION. Below this is a section titled "CUSTOMER DATA" with an "ADD" button. A search form is present, with fields for NIK (containing "7312042510720002") and NAME, and buttons for "RESET" and "SEARCH". Below the search form is a table displaying customer information:

NAME	NIK	GENDER	ACTION
ABDURRAUF, S.PD, M.PD	7312042510720002	LAKI-LAKI	

Gambar 4.22 Contoh tampilan menu Customer setelah klik “Search”

Pengujian pengubahan data dilakukan dengan *view* data terlebih dahulu seperti pada Gambar 4.23. Pada *view* ini, semua input dinonaktifkan sehingga *user* tidak dapat mengubah data. Selain itu, terdapat dua tombol yaitu tombol “Cancel” untuk mengarahkan kembali ke tampilan awal menu *Customer* dan tombol “Edit” untuk masuk ke mode edit. Jika *user* ingin mengubah data dapat langsung mengklik tombol “Edit” dan inputan akan menjadi aktif (kecuali inputan NIK) seperti pada Gambar 4.24. Kemudian *user* dapat langsung mengubah data dan mengklik tombol “Save” untuk menyimpan perubahan. Setelah perubahan tersimpan, *user* akan langsung kembali ke tampilan awal pada menu *Customer*.

HOME CUSTOMER AUTHENTICATION



CUSTOMER MAIN INFORMATION

NAME	ABDURRAUF, S.PD, M.PD	GENDER	MALE
BIRTH PLACE	CELLENGENCE	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	JL. MERDEKA NO 43	PROVINSI	SULAWESI SELATAN
		KOTA	KAB. SOPPENG
		KECAMATAN	LALABATA
RT / RW	001	KELURAHAN	BILA

CANCEL **EDIT**

Gambar 4.23 Contoh tampilan *view* data

HOME CUSTOMER AUTHENTICATION



CUSTOMER MAIN INFORMATION

NAME	ABDURRAUF, S.PD, M.PD	GENDER	MALE
BIRTH PLACE	CELLENGENCE	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)

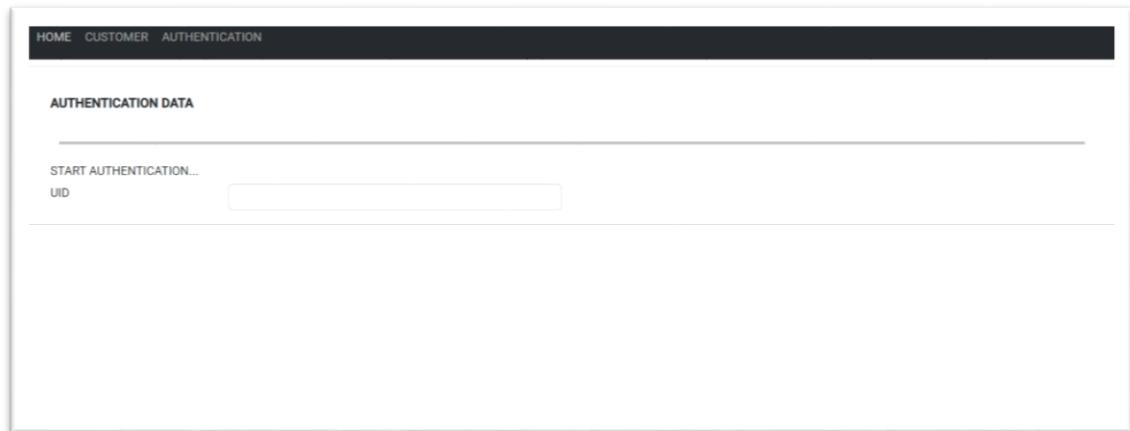
ADDRESS INFORMATION

ADDRESS	JL. MERDEKA NO 43	PROVINSI	SULAWESI SELATAN
		KOTA	KAB. SOPPENG
		KECAMATAN	LALABATA
RT / RW	001	KELURAHAN	BILA

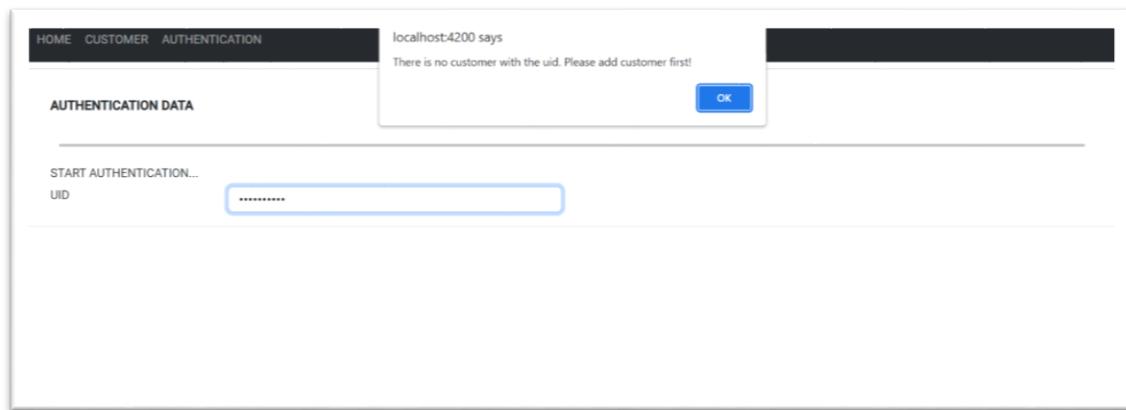
CANCEL **SAVE**

Gambar 4.24 Contoh tampilan *edit* data

User juga dapat melakukan pencarian data menggunakan pembacaan uid KTP dengan masuk ke menu *Authentication*. Gambar 4.25 adalah tampilan pertama kali ketika masuk ke manu *Authentication*. Pada tampilan ini terdapat *field* UID untuk pembacaan uid KTP. Untuk memulai autentikasi, *user* harus menyambungkan laptop dengan RFID menggunakan USB terlebih dahulu. Kemudian, *user* dapat mengarahkan kursor ke *field* UID dan menempelkan KTP ke RFID. Ketika RFID sudah membaca uid KTP, jika uid sudah pernah disimpan sebelumnya, secara otomatis akan ditampilkan data customer pada mode *view* seperti pada Gambar 4.23. Jika uid yang dibaca belum terdaftar, akan ada pemberitahuan seperti pada Gambar 4.26 dan *user* dapat mendaftarkan data terlebih dahulu.



Gambar 4.25 Tampilan pertama ketika masuk menu *Authentication*



Gambar 4.26 Contoh jika uid yang dicari belum terdaftar

4.2 Evaluasi Pengujian

Evaluasi pengujian yang dilakukan terdiri dari evaluasi untuk metode *Optical Character Recognition* (OCR) yang diuji dan evaluasi aplikasi *web*. Evaluasi metode OCR dibagi menjadi evaluasi metode *Pytesseract* dan evaluasi metode *Template Matching*, sedangkan evaluasi aplikasi *web* adalah evaluasi jalannya aplikasi.

1. Evaluasi OCR

Evaluasi OCR dibagi menjadi dua yaitu evaluasi terhadap metode *Pytesseract* dan metode *Template Matching*. Evaluasi yang dilakukan terdiri dari evaluasi dari hasil akurasi kedua metode.

a. Evaluasi Metode *Pytesseract* Untuk OCR KTP

Pengujian metode *Pytesseract* dilakukan dalam kondisi gelap dengan menggunakan penerangan 10 LUX, kondisi terang dengan menggunakan penerangan 122 LUX, kondisi terang sekali dengan menggunakan penerangan 390 LUX, dan menggunakan kamera laptop pada penerangan 59 LUX. Hasil pengujian metode *Pytesseract* dapat dilihat pada tabel 4.1. Hasil pembacaan pada kondisi kondisi gelap memiliki rata-rata tingkat akurasi 98,02%, pada kondisi terang yaitu memiliki rata-rata tingkat akurasi 98,4%, pada kondisi terang sekali memiliki tingkat rata-rata akurasi 98,95%, dan pada kondisi menggunakan kamera laptop memiliki rata-rata tingkat akurasi 97,94%.

Berdasarkan hasil pengujian akurasi metode *Pytesseract* pada Tabel 4.1 dilakukan uji normalitas menggunakan uji *Shapiro-Wilk*. Hasil uji normalitas ini dapat dilihat pada Gambar 4.27. Berdasarkan Gambar 4.27, *p-value* kurang dari 0,05 sehingga dapat disimpulkan bahwa data hasil pengujian *Pytesseract* tidak berdistribusi normal. Kemudian dilakukan uji *Kruskal-Wallis* untuk menentukan apakah setiap kondisi memiliki perbedaan yang signifikan. Hasil uji *Kruskal-Wallis* ini dapat dilihat pada Gambar 4.28. Hasil pada Gambar 4.28 menunjukkan bahwa *p-value* lebih dari 0,05 sehingga dapat disimpulkan bahwa keempat kondisi tidak memiliki beda yang signifikan.

```
ShapiroResult(statistic=0.8779436349868774, pvalue=0.0004634644719772041)
```

Gambar 4.27 Hasil uji *Shapiro-Wilk* untuk uji normalitas *Pytesseract*

```
KruskalResult(statistic=1.3559456398641103, pvalue=0.7158930126052936)
```

Gambar 4.28 Hasil uji *Kruskal-Wallis* untuk uji nonparametrik *Pytesseract*

b. Evaluasi Metode *Template Matching* Untuk OCR KTP

Sama seperti pada pengujian metode *Pytesseract*, pengujian metode *Template Matching* dilakukan dalam kondisi gelap dengan menggunakan penerangan 10 LUX, kondisi terang dengan menggunakan penerangan 122 LUX, kondisi terang sekali dengan menggunakan penerangan 390 LUX, dan menggunakan kamera laptop pada pencahayaan 59 LUX. Hasil dari pengujian metode *Template Matching* terdapat pada Tabel 4.2. Pada tabel tersebut dapat dilihat bahwa hasil pembacaan pada kondisi kondisi gelap memiliki rata-rata tingkat akurasi 62,91%, pada kondisi terang yaitu memiliki rata-rata tingkat akurasi 73,67%, pada kondisi terang sekali memiliki tingkat rata-rata akurasi 75,22%, dan pada kondisi menggunakan kamera laptop memiliki rata-rata tingkat akurasi 57,53%.

Berdasarkan hasil pengujian akurasi metode *Pytesseract* pada Tabel 4.1 dilakukan uji normalitas menggunakan uji *Shapiro-Wilk*. Hasil uji normalitas ini dapat dilihat pada Gambar 4.29. Berdasarkan Gambar 4.29, *p-value* lebih dari 0,05 sehingga dapat disimpulkan bahwa data hasil pengujian *Pytesseract* berdistribusi normal. Selanjutnya dilakukan uji homogenitas menggunakan uji *Bartlett*. Hasil uji *Bartlett* dapat dilihat pada Gambar 4.30. Berdasarkan Gambar 4.30, *p-value* lebih dari 0,05 sehingga dapat disimpulkan bahwa data hasil akurasi metode *Template Matching* memiliki variasi yang homogen.

```
ShapiroResult(statistic=0.9751186370849609, pvalue=0.5139623880386353)
```

Gambar 4.29 Hasil uji *Shapiro-Wilk* untuk uji normalitas *Template Matching*

```
BartlettResult(statistic=4.708487408276337, pvalue=0.19443066966271166)
```

Gambar 4.30 Hasil uji *Bartlett* untuk uji homogenitas *Template Matching*

Berdasarkan hasil uji normalitas dan uji homogenitas, berikutnya akan dilakukan uji *one way anova* untuk menguji signifikansi dari keempat pengkondisian tersebut. Hasil dari uji *one way anova* dapat dilihat pada Gambar 4.31. Berdasarkan Gambar 4.31, *p-value* yang didapat lebih kecil dari 0,05 sehingga dapat disimpulkan bahwa setidaknya ada satu pasangan yang memiliki beda rata-rata hasil akurasi yang signifikan.

	df	sum_sq	mean_sq	F	PR(>F)
group	3.0	2181.63252	727.21084	3.99114	0.014931
Residual	36.0	6559.42644	182.20629	NaN	NaN

Gambar 4.31 Hasil uji *one way anova* untuk uji parametrik *Template Matching*

Untuk mencari tahu pasangan pengkondisian yang memiliki beda signifikan dilakukan uji lebih lanjut. Hasil uji dapat dilihat pada Gambar 4.32. Berdasarkan Gambar 4.32 dapat dilihat pasangan kondisi dengan nilai *p-value* lebih kecil dari 0,05 adalah pasangan kondisi terang sekali dengan gelap, terang dengan kamera laptop, dan terang sekali dengan kamera laptop sehingga dapat disimpulkan bahwa ketiga pasangan tersebut memiliki beda yang signifikan.

	coef	std err	t	P> t	Conf. Int. Low	Conf. Int. Upp.	pvalue-hs	reject-hs
kamera_laptop-gelap	-5.380	6.03666	-0.891221	0.378728	-17.622913	6.862913	0.614022	False
terang-gelap	10.766	6.03666	1.783437	0.082948	-1.476913	23.008913	0.228774	False
terang_sekali-gelap	12.314	6.03666	2.039870	0.048753	0.071087	24.556913	0.181208	False
terang-kamera_laptop	16.146	6.03666	2.674658	0.011180	3.903087	28.388913	0.054665	False
terang_sekali-kamera_laptop	17.694	6.03666	2.931091	0.005836	5.451087	29.936913	0.034507	True
terang_sekali-terang	1.548	6.03666	0.256433	0.799076	-10.694913	13.790913	0.799076	False

Gambar 4.32 Hasil uji untuk menentukan pasangan beda signifikan antara keempat pengkondisian pada metode *Template Matching*

2. Evaluasi Autentikasi

KTP yang digunakan uji coba dibuat pada tahun 2011 hingga 2019. Berdasarkan hasil autentikasi pada Tabel 4.3, *Radio Frequency Identification* (RFID) yang digunakan dapat membaca semua uid KTP uji coba. Dari hasil tersebut, RFID dapat membaca uid KTP yang dibuat mulai dari tahun 2011.

3. Evaluasi Aplikasi Web

Evaluasi aplikasi *web* yang dilakukan adalah evaluasi jalannya aplikasi dan fitur-fitur yang sudah dikembangkan. Evaluasi ini dilakukan untuk memastikan aplikasi dapat berjalan dengan lancar. Hasil evaluasi aplikasi *web* dapat dilihat pada tabel 4.4

Tabel 4.4 Hasil Evaluasi Pengujian Aplikasi *Web*

No.	Aspek Pengujian	Hasil
1.	Pembacaan KTP (upload file) menggunakan OCR	Berhasil
2.	Pembacaan KTP (foto langsung) menggunakan OCR	Berhasil
3.	Penambahan data	Berhasil
4.	<i>View</i> data	Berhasil
5.	Pengubahan data	Berhasil
6.	Pencarian data	Berhasil
7.	Autentikasi data	Berhasil

Aplikasi *web* yang dikembangkan sudah berjalan dengan lancar. Semua fitur aplikasi juga dapat berjalan dengan lancar. Fitur pembacaan OCR berhasil walaupun tidak selalu memberikan akurasi 100% karena metode *Pytesseract* dan *Template Matching* yang digunakan belum dapat memberikan hasil dengan tingkat akurasi 100%. Fitur penambahan dan pengubahan data berhasil menyimpan data ke *database*. Fitur pencarian dan *view* data berhasil menampilkan data yang sesuai. Fitur autentikasi juga dapat berjalan baik dengan berhasil membaca uid KTP dan menampilkan data yang sesuai.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Aplikasi berbasis *web* ini dapat melakukan otomatisasi input data pribadi yaitu kartu tanda penduduk (KTP) menggunakan *Optical Character Recognition* (OCR). Metode OCR yang digunakan adalah metode *Pytesseract*. Tingkat akurasi *Pytesseract* sudah mencapai 97,94% di ketika menggunakan kamera laptop, 98,02% di kondisi gelap, 98,4% di kondisi terang, dan 98,95% di kondisi terang sekali. Selain itu, pada aplikasi ini dapat membantu proses autentikasi KTP dengan membaca uid KTP menggunakan *Radio Frequency Identification* (RFID) dan menampilkan data KTP sesuai uid yang dibaca.

Secara keseluruhan, aplikasi yang dikembangkan terdiri dari tiga menu yaitu Home, Customer, dan Authentication. Aplikasi memiliki fitur pembacaan KTP menggunakan *Optical Character Recognition* (OCR), penambahan data, pengubahan data, *view* data, pencarian data, dan autentikasi data.

Proses pengujian fitur aplikasi berjalan dengan lancar. Selama proses pengujian, semua fitur dapat berjalan dengan lancar. Hanya saja pada fitur pembacaan KTP, hasil pembacaan OCR belum 100% akurat.

5.2 Saran

Saran yang diberikan untuk pengembangan aplikasi selanjutnya adalah untuk lebih meningkatkan akurasi pembacaan OCR, baik menggunakan *Pytesseract*, *Template Matching*, atau metode lainnya. Diharapkan juga untuk dapat menghubungkan aplikasi dengan *database* pemerintah agar fitur autentikasi dapat lebih efektif. Selain itu, diharapkan nantinya ditambahkan fitur untuk membaca data lain seperti kartu keluarga dan tanda pengenal lainnya.

DAFTAR PUSTAKA

- Abdullah, S. S. & Muhammad, F. D., 2021. Penggunaan e-KTP untuk Registrasi Otomatis Memanfaatkan Sistem OCR Dengan Metode Template Matching Correlation. *Media Jurnal Informatika*, 12(2), 57-60.
- Acer, 2019. Swift 3 Acer Day Edition, Si Tipis Responsif dengan Performa Tinggi. *Acer* , [Online] Available at: <<https://www.acerid.com/laptop-swift-3-acer-day-edition-sf314-56g/>> [Accessed 18 Juni 2023].
- Adani, M. R., 2021. Cara Menggunakan Framework Bootstrap dan Keunggulan yang Dimiliki. *Sekawan Media*, [Online] Available at: <<https://www.sekawanmedia.co.id/blog/apa-itu-bootstrap/>> [Accessed 16 November 2022].
- Adawiyah, R., & Saragih, N. E., 2022. Implementasi Algoritma Levenshtein Distance Dalam Mendeteksi Plagiarisme. *Journal Computer Science and Information Technology (JCoInT)*, 5(1), 54-63.
- Angular, 2022. What is Angular?. *Angular*, [Online] Available at: <<https://angular.io/>> [Accessed 4 September 2022].
- Appkey, 2020. Node.js adalah? Memahami Penggunaan dan Keunggulan Node.js. *Webapp*, [Online] Available at: <<https://appkey.id/pembuatan-website/web-programming/nodejs-adalah/>> [Accessed 16 November 2022].
- Appkey, 2021. Apa itu PyCharm? Mengenal PyCharm Python, Kekurangan, serta Kelebihannya. *Webapp*, [Online] Available at: <<https://appkey.id/pembuatan-aplikasi/mobile-programming/pycharm-python/>> [Accessed 16 November 2022].
- Aprillian, H. D., 2019. Pemanfaatan Teknologi Optical Character Recognition pada Pembacaan Kartu Tanda Penduduk. *Dissertation*, Program Stui Teknik Informatika FTI-UKSW.
- Aprilia, P., 2021. Apa itu HTML? Berikut Fungsi dan Cara Kerjanya!. *Niagahoster*, [Online] Available at: <<https://www.niagahoster.co.id/blog/html-adalah/>> [Accessed 16 November 2022].

- Ariffudin, M., 2022^a. Mengenal Visual Code Studio dan Fitur-Fitur Pentingnya. *Niagahoster*, [Online] Available at: <<https://www.niagahoster.co.id/blog/visual-code-studio/>> [Accessed 16 November 2022].
- Ariffudin, M., 2022^b. Apa itu XAMPP? Sejarah, Fungsionalitas dan Fitur-Fitur XAMPP. *Niagahoster*, [Online] Available at: <<https://www.niagahoster.co.id/blog/xampp-adalah/>> [Accessed 16 November 2022].
- Ariffudin, M., 2022^c. Belajar Django, Framework Python yang Kian Populer. *Niagahoster*, [Online] Available at: <<https://www.niagahoster.co.id/blog/django-framework/>> [Accessed 16 November 2022].
- Asia, M., 2022. 2 Font Yang Dipakai di KTP (NIK dan Identitas). *Mucho*, [Online] Available at: <<https://mucho.asia/font-yang-dipakai-di-ktp/>> [Accessed 4 Desember 2022].
- Asih, M. S., 2017. Pengenalan Huruf Pada Citra Digital Menggunakan Algoritma Template Matching. *Jurnal Tenik Informatika*, Vol. 2.
- Bakti Kominfo, 2019. Sekilas Tentang Teknologi RFID, Alat Identifikasi Yang Banyak Dipakai Oleh Perusahaan. *Bakti Kominfo*, [Online] Available at: <https://www.baktikominfo.id/id/informasi/pengetahuan/sekilas_tentang_teknologi_rfid_alat_identifikasi_yang_banyak_dipakai_oleh_perusahaan-792> [Accessed 4 Desember 2022].
- Chrisdwianto, T. O., Fitriyah, H., & Widasari, E. R., 2018. Perancangan sistem deteksi dan pengenalan rambu peringatan menggunakan metode template matching. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(3), 1265-1274.
- Damara, D., 2022. Kredit Kendaraan Bermotor Meningkat per Juli 2022, Ini Penyebabnya. *Bisnis.com*, [Online] Available at: <<https://finansial.bisnis.com/read/20220820/90/1568691/kredit-kendaraan-bermotor-meningkat-per-juli-2022-ini-penyebabnya>> [Accessed 12 Juni 2023].

- Dispendukcapil Kota Semarang, 2015. Apa dan Mengapa E-KTP. *Dinas Kependudukan dan Pencatatan Sipil Kota Semarang*, [Online] Available at: <<https://www.dispendukcapil.semarangkota.go.id/berita-Apa-dan-Mengapa-E-KTP>> [Accessed 4 Desember 2022].
- Diwanti, H., Sumaryo, S., Setianignsih, C., 2019. Real Time Smart CCTV To Detect Vehicle License Plate Using Optical Character Recognition. *E-Proceeding of Engineering*, vol. 6, No.2 pp 3045-3052.
- Enterprise, J., 2019. *Python untuk Programmer Pemula*. Jakarta: PT Elex Media Komputindo.
- Ferdaush, N., & Islam, R., 2021. ‘An Open Source Tesseract Based Optical Character Recognizer for Bengali Language’. *Tesis*, Stamford University, Bangladesh.
- Haekal, M. M., 2021. Bootstrap: Pengertian, Kegunaan, Kelebihan, dan Kekurangannya. *Niagahoster*, [Onliner] Available at: <<https://www.niagahoster.co.id/blog/bootstrap-adalah/>> [Accessed 16 November 2022].
- Hani, S., Santoso, G., & Hikam, F. B., 2019. Perancangan Sistem Akses Kunci Elektronis Pada Kotak Penyimpanan Memanfaatkan E-KTP Dan Teknologi RFID. *Jurnal Teknologi Technoscoentia*, 68-76.
- Hermawan, M., 2021. Tag, Atribut dan Elemen pada HTML (Kode HTML Lengkap). *Itkoding*, [Online] Available at: <<https://itkoding.com/tag-atribut-elemen-kode-html/>> [Accessed 16 November 2022].
- Hoffstaetter dkk., 2022. Pytesseract. *Pypi.org*, [Online] Available at: <<https://pypi.org/project/pytesseract/>> [Accessed 16 November 2022].
- Huda, N., 2022. Apa Itu MariaDB? Ini Fungsi dan Perbedaannya dengan MySQL. *dewaweb*, [Online] Available at: <<https://www.dewaweb.com/blog/apa-itu-mariadb/>> [Accessed 16 November 2022].
- Indrakusuma, R. M. I., Ahmadiyah, A. S., Ariyani, N. F. 2021. Pengenalan dan klasifikasi tulisan pada nota pembelian material (studi kasus proyek konstruksi). *Jurnal Teknik ITS*, vol. 10, No 2.

- Inggihpangestu, 2022. Mengenal Apa Itu Angular, Sejarah, Fitur Fitur, Arsitektur, Kelebihan dan Kekurangan. *IDMETAFORA*, [Online] Available at: <<https://idmetafora.com/news/read/1920/mengenal-apa-itu-angular-sejarah-fitur-fitur-arsitektur-kelebihan-dan-kekurangan.html>> [Accessed 16 November 2022].
- JDIH BPK RI., 2022. Undang-undang (UU) Nomor 27 Tahun 2022 Perlindungan Data Pribadi. *JDIH BPK RI Database Peraturan*, [Online] Available at: <<https://peraturan.bpk.go.id/Home/Details/229798/uu-no-27-tahun-2022>> [Accessed 4 Desember 2022].
- Maskuri, M., 2017. Implementasi Metode Template Matching Untuk Media Pembelajaran Pengenalan Karakter Huruf Alphabet. *Program Studi Teknik Informatika Fakultas Teknik Universitas Nusantara PGRI Kediri*.
- Muharom, S., 2019. Pengenalan Nomor Ruangan Menggunakan Kamera Berbasis OCR Dan Template Matching. *Jurnal Inform*, 4(1).
- Nugroho, H., & Prabandaru, A. A., 2022. Deteksi Citra Nomor Panggil Buku Menggunakan Metode Template Matching Studi Kasus Perpustakaan ITATS. *Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika (SNESTIK)*, Vol. 1, No. 1, pp. 461-465.
- Oktriwina, A. S., 2020. Pelajari Keunggulan dan Kekurangan Bahasa Pemrograman Populer, TypeScript. *Glints*, [Online] Available at: <<https://glints.com/id/lowongan/typescript-adalah/#.Y4MXbHZBw2x>> [Accessed 16 November 2022].
- Pasaribu, Edward Samuel, 2018. Wilayah-Administratif-Indonesia. *Github*, [Online] Available at: <<https://github.com/edwardsamuel/Wilayah-Administratif-Indonesia>> [Accessed 4 Desember 2022].
- Phangriastu, M. R., 2017. Optical Character Recognition (OCR). *Binus MTI*, [Online] Available at: <<https://mti.binus.ac.id/2017/07/03/optical-character-recognition-ocr/>> [Accessed 4 Desember 2022], cit. Mohammad dkk. 2014. Optical Character Recognition Implementation Using Pattern Matching. *International Journal of Computer Science and Information Technologies*, 2088-2090.

- Phangtriastu, M. R., 2017. Optical Character Recognition (OCR). *Binus MTI*, [Online] Available at: <<https://mti.binus.ac.id/2017/07/03/optical-character-recognition-ocr/>> [Accessed 4 Desember 2022], cit. Rao dkk.
2016. Optical Charater Recognition Technique Algorithms. *Journal of Theoretical and Applied Information Technology*, 83(2),275-282.
- PRODUCTION, A. M., 2020. Ini yang Kalian Cari, Cara Menggunakan Ektp Multi Card di RFID RC522. *Youtube*, [Online] Available at: <<https://www.youtube.com/watch?v=qm4ylZr9Vg>> [Accessed 4 Desember 2022].
- Putra, E. D., Utami, M., Rifqo, M. H., 2021. Identifikasi Text Meteran Air Menggunakan Metode Run-Length Smearing Algorithm (RIsa). *Jurnal Ilmu Komputer dan Informatika*, vol. 4 No 2 pp. 98-106.
- Rabbani, A., 2022. Node.js: Pengertian, Arsitektur, Cara Kerja, Kelebihan, Kekurangan, Contoh Penggunaan, dan Perbedaannya dengan Javascript. Social79, [Online] Available at: <<https://www.sosial79.com/2022/05/nodejs-pengertian-arsitektur-cara-kerja.html>> [Accessed 16 November 2022].
- Rahman, S., & Sahira, U., Shayputri, N., 2018. Pengenalan Iris Mata Menggunakan Metode Template Matchig Correlation. (*JurTI Jurnal Teknologi Informasi*, 2(2), 105-112.
- Risyan, R., 2020. Pengertian MariaDB Dan Cara Kerjanya. *Monitor Teknologi*, [Online] Available at: <<https://www.monitorteknologi.com/pengertian-mariadb/>> [Accessed 16 November 2022].
- Salar, S., Roy, S., Verma, S., 2021. Automate Identification and Recognition of Handwritte Text from an Image. *Turkish Journal of Computer and Mathematics Education*, vol. 12 No.3 pp 3800-3808.
- Sawalkar, A., Pathan, A., Kakade, A., Telvekar, P., Chandne, B., 2022. Number Plate Recognition System Using Pytesseract & Open CV. *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4 pp 5861-5863.

- Shaleng, Abdurrauf, 2021. Makna di Balik Nomor KTP atau NIK yang Tersimpan di Dompet Anda. *Gurusiana*, [Online] Available at: <<https://www.gurusiana.id/read/abdurraufshaleng/article/makna-di-balik-nomor-ktp-atau-nik-yang-tersimpan-di-dompet-anda-10009>> [Accessed 12 Juli 2023].
- Sopa, A., Saputra, H. M., & Nurhakim, A., 2020. Sistem Kehadiran Menggunakan RFID pada E-KTP Berbasis Internet Of Things. *Prosiding-Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung*, pp. 412-418.
- Smith, R., 2007. An overview of the Tesseract OCR engine. *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2 pp. 629-633.
- Trivusi, 2022. Apa itu Algoritma Levenshtein Distance? Berikut Definisinya. *Trivusi*, [Online] Available at: <<https://www.trivusi.web.id/2022/04/apa-itu-algoritma-levenshtein-distance.html>> [Accessed 12 Juli 2023].
- Uddin, Q., 2021. Features Extraction of Tax Card by Using OCR Based DeepLearning Techniques. *Tesis*, University of Eastern Finland, Finlandia.
- Valle, L. d., 2019. rlsa-python. *Github*, [Online] Available at: <https://github.com/luizvalle/rlsa_python/blob/master/rlsa_python/rlsa.py> [Accessed 11 Juni 2023].
- Wibowo, P. T., 2021. Apa Itu Angular?. *Warta Ekonomi*, [Online] Available at: <<https://wartaekonomi.co.id/read380412/apa-itu-angular>> [Accessed 16 November 2022].
- Widodo, S. dan Gunawan, 2015. Template Matching Pada Citra E-KTP Indonesia. *Prosiding SNATIKA*, 3, 30-35.