

SKRIPSI

**PEMBACAAN ANGKA METERAN LISTRIK PASCABAYAR DENGAN
MENGUNAKAN ALGORITMA *OPTICAL CHARACTER RECOGNITION*
(OCR)**

Disusun dan diajukan oleh

SUCIATI

D041 17 1007



DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2022

LEMBAR PENGESAHAN

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Suciati

NIM : D041 17 1007

Program Studi : Teknik Elektro

Jenjang : **S1** / S2 / S3

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**PEMBACAAN ANGKA METERAN LISTRIK PASCABAYAR DENGAN
MENGUNAKAN ALGORITMA *OPTICAL CHARACTER RECOGNITION*
(OCR)**

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi/tesis/disertasi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi/tesis/disertasi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, Januari 2022

Suciati

ABSTRAK

Energi listrik yang digunakan untuk keperluan rumah diukur dengan kilowatt hour meter, disingkat kWh meter. PT PLN (Perusahaan Listrik Negara) adalah perusahaan yang ditunjuk oleh negara untuk menyediakan jasa kelistrikan di Indonesia. Proses pembacaan dan pencatatan penggunaan kWh yang masih diketik secara manual membutuhkan waktu yang banyak dalam proses penginputan. Penelitian ini bertujuan untuk mengembangkan sebuah perangkat lunak yang dapat meningkatkan akurasi pembacaan meteran listrik pascabayar. Pada tahapan awal dilakukan akuisisi citra selanjutnya proses segmentasi menggunakan *bounding box* dan dilanjutkan pencocokan karakter dengan citra yang telah dilatih melalui fitur *ocrTrainer* pada matlab. Berdasarkan hasil pengujian, sistem pembacaan angka meteran listrik pascabayar dengan menggunakan algoritma *Optical Character Recognition* (OCR) memiliki persentase keberhasilan atau tingkat akurasi pengenalan baik dengan akurasi pengenalan 81,81%.

Kata Kunci: *Optical Character Recognition* (OCR), kWh Meter, Matlab, citra digital

ABSTRACT

Electrical energy used for household purposes is measured in kilowatt-hour meters, abbreviated as kWh meters. PT PLN (State Electricity Company) is a company appointed by the state to provide electricity services in Indonesia. The process of reading and recording the use of kWh which is still typed manually requires a lot of time in the input process. This study aims to develop a software that can improve the accuracy of postpaid electricity meter readings. At the beginning of the shooting, the segmentation process uses a bounding box and continues the characters that have been done through the ocrTrainer feature in Matlab. Based on the test, the postpaid electricity reading system using the Optical Character Recognition (OCR) algorithm has a good percentage of success or recognition rate with a recognition accuracy of 81.81%.

Keywords: *Optical Character Recognition (OCR), kWh Meter, Matlab, digital image*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat rahmat serta kasih-Nya sehingga penulis dapat menyelesaikan skripsi berjudul: “Pembacaan Angka Meteran Listrik Pascabayar dengan Menggunakan Algoritma *Optical Character Recognition* (OCR)”.

Penulis menyadari bahwa dalam penyelesaian tugas akhir ini banyak kendala yang dihadapi, baik itu kendala teknis maupun nonteknis. Pada kesempatan ini dengan segala kerendahan hati dan penuh rasa hormat penulis menghaturkan terima kasih yang sebesar-besarnya bagi semua pihak yang telah membeikan bantuan moril maupun material baik langsung maupun tidak langsung dalam penyusunan skripsi ini hingga selesai. Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tulus kepada:

1. Kedua orang tua penulis dan kakak penulis, yang selalu memberikan dukungan, do'a, dan semangat.
2. **Dr. Ir. Rhiza S. Sadjad, MSEE** selaku Pembimbing I dan **Dr. Ir. Ingrid Nurtanio, M.T** selaku Pembimbing II, terima kasih telah membimbing serta memberikan masukan terhadap penelitian yang dilakukan.
3. **Prof. Dr-Ing. Faizal Arya Samman, S.T., M.T** dan **Ida Rachmaniar Sahali, S.T., M.T** selaku dosen penguji yang memberikan saran, koreksi, dan arahan dalam menyelesaikan skripsi penulis.
4. **Dr. Eng. Ir. Dewiani, M.T** dan **Dr. Ikhlas Kitta, S.T., M.T.** selaku Ketua dan Sekretaris Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin.

5. Seluruh dosen pengajar serta pegawai Departemen Teknik Elektro Universitas Hasanuddin atas bimbingan, didikan, kemudahan dan bantuan yang telah diberikan kepada penulis selama menempuh perkuliahan.
6. Teman-teman seperjuangan di Laboratorium Sistem Kendali dan Instrumentasi. Alvanya Yostha Paramita, Arson Marianus, Ilham Ramli, Muhammad Alif Fikri dan Min Idznullah Said yang selalu membantu dan memberikan dorongan dalam penyelesaian tugas akhir ini.
7. Yulianti dan Nurul Fadillah, meski tak selalu ada tapi do'a dan dukungannya selalu menyertai.
8. Keluarga besar Teknik Elektro 2017 “**EQUAL17ER**” yang membersamai penulis dalam menuntut ilmu hingga banyak membantu dalam penyelesaian tugas kahir ini.

Akhirnya dengan segala kerendahan hati, penulis menyadari masih terdapat kekurangan dalam penyusunan tugas akhir ini baik isi maupun cara penyajian. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan. Akhir kata, semoga penelitian ini dapat bermanfaat bagi penulis dan pembaca sekalian. Aamiin.

Gowa, 28 Desember 2021

Suciati

DAFTAR ISI

SAMPUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN KEASLIAN	Error! Bookmark not defined.
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
1.6 Tahapan Penelitian	4
1.7 Sistematika Penulisan.....	5
BAB 2 KAJIAN PUSTAKA	6

2.1	Meteran Listrik Pascabayar	6
2.1.1	Perhitungan kWh Meter Pascabayar	7
2.2	Pengolahan Citra Digital	7
2.3	<i>Optical Character Recognition (OCR)</i>	10
2.3.1	Segmentasi Citra	11
2.4	Penelitian Terkait.....	12
BAB 3 METODOLOGI PENELITIAN		16
3.1	Lokasi dan Waktu Penelitian.....	16
3.2	Peralatan Penelitian	16
3.2.1	Kebutuhan Perangkat Keras (<i>Hardware</i>)	16
3.2.2	Kebutuhan Perangkat Lunak (<i>Software</i>)	16
3.3	Tahapan Penelitian	16
3.4	Proses Kerja Sistem.....	17
3.4.1	Jenis Meteran yang Digunakan	19
3.4.2	Proses Akuisisi Citra	19
3.4.3	Proses <i>Grayscale</i>	21
3.4.4	Proses Binerisasi	22
3.4.5	Proses Segmentasi Citra.....	23
3.4.6	Proses Training Angka.....	25
3.4.7	Proses Rekognisi Teks Karakter	27

BAB 4 HASIL DAN PEMBAHASAN	29
4.1 Perancangan Tampilan <i>Graphical User Interface</i> (GUI).....	29
4.2 Hasil Pengujian Sistem.....	30
BAB 5 PENUTUP.....	33
5.1 Kesimpulan.....	33
5.2 Saran.....	33
DAFTAR PUSTAKA.....	35

DAFTAR GAMBAR

Gambar 2.1. Meteran Listrik Pascabayar	6
Gambar 2.2 Contoh Representasi Piksel dengan Kombinasi warna R, G, dan B ...	8
Gambar 2.3 Citra biner dengan nilai piksel 0 atau 1 [3]	10
Gambar 3.1 Diagram Alir Penelitian	17
Gambar 3.2 Diagram alir proses pengenalan angka.....	18
Gambar 3.3 Jenis jenis meteran	19
Gambar 3.4 Contoh Hasil dari Akuisisi Citra	21
Gambar 3.5 Perbedaan Warna RGB (kiri) dan Gray (kanan)	21
Gambar 3.6 Hasil dari Proses Binerisasi.....	23
Gambar 3.7 Citra dengan Menggunakan Bounding Box	24
Gambar 3.8 Citra Setelah Penentuan Nilai aspectRatio.....	25
Gambar 3.9 Font yang digunakan dalam proses training angka	25
Gambar 3.10 Proses Training Angka	26
Gambar 3.11 Hasil dari Rekognisi Teks	27
Gambar 4.1 Tampilan GUI dari proses OCR.....	29
Gambar 4.2 Contoh meteran listrik yang gagal terbaca.....	32

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	12
Tabel 4.1 Hasil Pembacaan Meteran Listrik Pascabayar	30

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Energi listrik merupakan kebutuhan dasar dalam mendorong segala jenis aktivitas roda kehidupan manusia. Diantaranya dapat digunakan sebagai penerangan, fasilitas umum, keperluan rumah tangga, keperluan industri dan juga membantu peningkatan ekonomi negara. PT PLN (Perusahaan Listrik Negara) adalah perusahaan yang ditunjuk oleh negara untuk menyediakan jasa kelistrikan di Indonesia. Dalam mengukur konsumsi listrik pelanggannya, PT PLN menggunakan *kilowatt hour meter* (kWh). Perhitungan kWh setiap bulannya adalah perkalian harga satuan tarif dasar listrik ditambah dengan nilai abodemen dan pajak [1]. Jenis kWh yang digunakan oleh PLN dibagi menjadi dua macam, yaitu listrik pascabayar dan prabayar, keduanya memiliki layanan dan tarif yang sama tetapi tetap memiliki kelebihan dan kekurangan masing-masing.

Perkembangan teknologi dapat dengan cepat menyelesaikan masalah yang sering dijumpai dalam kehidupan sehari-hari. Pada penelitian ini peneliti membuat sistem pembacaan angka meteran listrik pascabayar dengan menggunakan *software* Matlab. Di Indonesia penginputan data penggunaan listrik per bulannya ke dalam sistem masih dilakukan secara manual. Pemrosesan suatu citra gambar meteran listrik pascabayar adalah inovasi terbaru untuk mempermudah pekerjaan penginputan data ke dalam sistem. Pengolahan citra digital adalah salah satu disiplin ilmu yang mempelajari tentang teknik maupun cara mengolah citra, dalam hal ini berupa gambar diam (foto) maupun gambar bergerak (video) yang berasal dari

perangkat atau alat akuisisi citra seperti kamera digital, webcam, maupun perangkat smartphone yang diolah secara digital menggunakan komputer [2]. Gambar huruf yang dimaksud dapat berupa hasil scan dokumen, hasil *print-screen* halaman web, hasil foto, dan lain-lain [3].

Dari masalah inilah sehingga peneliti memutuskan untuk melakukan sebuah penelitian dengan judul **“PEMBACAAN ANGKA METERAN LISTRIK PASCABAYAR DENGAN MENGGUNAKAN ALGORITMA *OPTICAL CHARACTER RECOGNITION* (OCR)”**. Dengan dilakukannya penelitian ini, peneliti berharap dapat membantu petugas PLN dalam penginputan data sehingga tidak lagi dilakukan secara manual yang nantinya dapat mengefisienkan waktu.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang maka rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana *Optical Character Recognition* dalam mengonversi sebuah angka pada gambar meteran listrik menjadi teks?
2. Bagaimana mengimplementasikan program dalam pengenalan angka pada meteran listrik?
3. Bagaimana akurasi yang didapatkan dari sistem pembacaan angka meteran listrik pascabayar?

1.3 Tujuan Penelitian

Tujuan yang hendak dicapai dari penelitian ini antara lain:

1. Untuk mengetahui proses OCR dalam mengonversi angka dari sebuah meteran listrik menjadi teks

2. Untuk mengetahui cara pengimplementasian program dalam pengenalan angka pada meteran listrik
3. Untuk menganalisis akurasi yang didapatkan dari sistem pembacaan meteran listrik pascabayar

1.4 Batasan Masalah

Untuk mendapatkan hasil akhir yang lebih terperinci dan terfokus, maka permasalahan yang akan dibahas dibatasi oleh beberapa ketentuan berikut:

1. *Software* dalam penelitian ini menggunakan dengan menggunakan Matlab R2016a
2. Dikhususkan dalam pembacaan meteran listrik pascabayar yang diambil oleh satu kamera ponsel dari jarak dekat
3. Dalam proses penelitian yang menjadi inputan adalah berupa gambar meteran listrik dengan angka 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9 sebagai pola yang harus dikenali.
4. Meteran yang menjadi sampel adalah meteran pascabayar model lama dan model baru
5. Gambar yang digunakan adalah gambar yang diambil sendiri oleh peneliti dengan resolusi 8 MP.

1.5 Manfaat Penelitian

Diharapkan dengan adanya penelitian ini sehingga dapat diambil manfaat sebagai berikut:

1. Bagi PT PLN sebagai masukan atau bahan pertimbangan dalam melakukan penginputan data meteran listrik pascabayar setiap bulannya.
2. Bagi institusi Universitas Hasanuddin, penelitian ini dapat berguna sebagai referensi ilmiah dalam pengembangan pengolahan citra.
3. Bagi peneliti, penelitian ini memiliki manfaat untuk menambah wawasan dan pengalaman yang nantinya diharapkan dapat bermanfaat di lapangan.

1.6 Tahapan Penelitian

Untuk menghasilkan tugas akhir yang komprehensif, maka dalam penelitian akan digunakan metode sebagai berikut:

1. Studi literatur
Studi literatur dilakukan untuk membentuk landasan teori yang konkrit berdasarkan literatur terkait, sebelum melakukan implementasi dan pengujian secara langsung
2. Pengujian dan analisis
Kegiatan pengujian dan analisis dimaksudkan untuk memperoleh data-data aktual yang merupakan hasil pengukuran dan observasi secara langsung
3. Diskusi dan konsultasi
Melakukan dialog secara langsung kepada pembimbing dan pihak-pihak yang berkompeten di bidang terkait untuk mendapatkan pengetahuan mengenai penelitian yang dilakukan.

1.7 Sistematika Penulisan

Tugas akhir ini terdiri atas lima bab, dengan sistematika penulisan sebagai berikut:

Bab 1 Pendahuluan

Bab ini menguraikan atas latar belakang, rumusan masalah, tujuan penulisan, batasan masalah, metode penelitian dan sistematika penulisan.

Bab 2 Tinjauan Pustaka

Bab ini berisi tentang teori-teori yang berkaitan dengan penelitian yang dilakukan, diantaranya Pengolahan Citra, Pengenalan Angka, *Optical Character Recognition*, dan beberapa penelitian yang terkait.

Bab 3 Metodologi Penelitian

Bab ini berisi tentang waktu dan tempat penelitian, mendeskripsikan metode yang digunakan dalam penyusunan tugas akhir ini serta bagan alir penelitian

Bab 4 Hasil dan Pembahasan

Bab ini berisi tentang pembahasan dari hasil pengolahan data yang disertai dengan tabel hasil penelitian

Bab 5 Penutup

Bab ini merupakan bab penutup yang berisi kesimpulan dan saran

BAB 2

KAJIAN PUSTAKA

2.1 Meteran Listrik Pascabayar



Gambar 2.1. Meteran Listrik Pascabayar

Meteran listrik pascabayar atau dalam dunia kelistrikan dikenal dengan kWh Meter Pascabayar adalah kWh Meter yang sistem/metoda pembayarannya setiap 1 bulan sekali, kWh Meter Pascabayar ini terdiri dari beberapa tipe, yaitu:

1. kWh Meter Pascabayar Analog/Konvensional
2. kWh Meter Pascabayar Digital

kWh Meter pascabayar analog/konvensional adalah meteran yang umum digunakan dan mudah ditemukan di masyarakat. PT. PLN (Persero) menggunakan kWh Meter untuk menghitung/ mengukur besar energi listrik yang digunakan pelanggan pada saat pelanggan menggunakan energi listrik [4]. Besar tagihan listrik dapat dihitung berdasarkan pada angka-angka yang tertera pada kWh Meter,

dan biasanya PT. PLN menghitung/mengukur Energi yang digunakan konsumen setiap bulan (Analog/Mekanik).

2.1.1 Perhitungan kWh Meter Pascabayar

Pada kWh Meter Pascabayar (Konvensional/Mekanik) besarnya penggunaan energi listrik dicatat setiap bulannya oleh petugas PLN (catat meter) dan kemudian dikalkulasikan dengan harga jual sesuai Tarif Dasar Listrik yang ditetapkan oleh PT. PLN (PERSERO).

Untuk mengetahui hitungan berapa putaran kWh Meter Pascabayar untuk mencapai 1 kWh, ketika membeli sebuah kWh Meter maka akan tercantum X putaran per kWh, artinya untuk mencapai 1 kWh dibutuhkan putaran sebanyak x kali putaran dalam setiap jamnya. Contohnya jika putaran 900 putaran per kWh maka harus ada 900 putaran setiap jamnya untuk dikatakan sebesar satu kWh. Jumlah kWh itu secara kumulatif dihitung dan pada akhir bulan dicatat oleh petugas besarnya pemakaian yang lalu dikalikan dengan tarif dasar listrik atau TDL ditambah dengan biaya abodemen dan pajak menghasilkan jumlah tagihan yang harus dibayar setiap bulannya.

2.2 Pengolahan Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek yang dapat diolah oleh komputer [5]. Pengolahan citra digital merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya

kapasitas dan kecepatan proses komputer serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra.

Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat (x,y) dan amplitudo $f(x,y)$. Citra dibagi menjadi tiga bagian, yaitu:

1. Citra RGB

Setiap piksel pada citra RGB, memiliki intensitas warna yang merupakan kombinasi dari tiga nilai intensitas pada kanal R (*Red*), G (*Green*), dan B (*Blue*). Banyaknya kombinasi warna piksel yang mungkin pada citra RGB *truecolor* 24-bit adalah sebanyak $256 \times 256 \times 256 = 16.777.216$.

HITAM R = 0 G = 0 B = 0	PUTIH R = 255 G = 255 B = 255	KUNING R = 255 G = 255 B = 255
---	---	--

Gambar 2.2 Contoh Representasi Piksel dengan Kombinasi warna R, G, dan B

2. Citra Grayscale

Citra *grayscale* merupakan citra yang nilai intensitas pikselnya didasarkan pada derajat keabuan. Pada citra *grayscale* 8-bit, derajat warna hitam sampai dengan putih dibagi kedalam 256 derajat keabuan dimana warna putih sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0. Citra RGB dapat dikonversi menjadi citra

grayscale. Persamaan yang umumnya digunakan untuk mengonversi citra RGB *truecolor* 24-bit menjadi citra *grayscale* 8-bit adalah:

$$Grayscale = 0,2989 * R + 0,5878 * G + 0,1140 * B.....(1)$$

Keterangan:

Grayscale adalah nilai intensitas citra *grayscale*, R adalah nilai intensitas piksel pada kanal merah, G adalah nilai intensitas piksel pada kanal hijau dan B adalah nilai intensitas piksel pada kanal biru.

Sehingga proses konversi menghasilkan citra *grayscale* yang hanya memiliki satu kanal warna [6]. Dalam matlab untuk mengubah citra RGB menjadi citra *grayscale* dapat digunakan perintah sederhana, yaitu

$$J = rgb2gray$$

Adapun perintah yang dapat digunakan dalam mengonversi citra RGB menjadi citra *grayscale* tanpa menggunakan perintah *rgb2gray* adalah sebagai berikut:

```
R = img(:, :, 1);
G = img(:, :, 2);
B = img(:, :, 3);

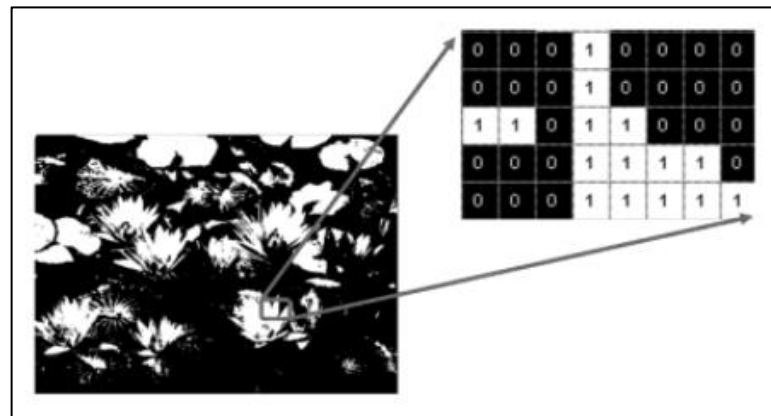
Img = zeros(size(img,1), size(img,2), 'uint8');
for x=1:size(img,1)
    for y=1:size(img,2)
        Img(x,y) = (R(x,y)*.2989)+(G(x,y)*.5870)+(B(x,y)*.1140);
    end
end
```

3. Citra Biner (*black and white*)

Citra biner adalah citra yang pikselnya memiliki kedalaman bit 1-bit sehingga hanya memiliki dua nilai intensitas warna yaitu 0 (hitam) dan 1

(putih). Dalam matlab nilai *threshold* diatur dalam kelas data double, sehingga untuk mengatur nilai *threshold* 128, nilai yang digunakan adalah $128/256=0,5$.

Berikut merupakan contoh gambar yang menunjukkan citra biner, perbedaan letak angka 0 dan 1 dari sebuah citra:



Gambar 2.3 Citra biner dengan nilai piksel 0 atau 1

2.3 Optical Character Recognition (OCR)

OCR adalah sistem yang sudah lama dikembangkan Tahun 1941, Emmanuel Goldberg telah mulai membuat sistem OCR yang berfungsi untuk telegram dan alat baca untuk orang tunanetra. Sistem OCR terus dikembangkan hingga kini dapat menghasilkan akurasi yang lebih baik bahkan dalam situasi-situasi yang dimana karakter sulit untuk dikenali. Sistem ini misalnya dipakai untuk mendeteksi sidik jari, tanda tangan, bahkan wajah seseorang. Ada banyak pendekatan yang dapat dipakai untuk mengembangkan pembuatan pengenalan pola otomatis antara lain memakai pendekatan numerik, statistic, neural dan aturan produksi (*rule-based*)[7].

Optical character recognition (OCR) juga merupakan sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak

(printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (*image character*) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam *database* aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar output scanner di mana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan [8].

Sistem OCR terus dikembangkan hingga kini sehingga dapat menghasilkan akurasi yang lebih baik bahkan dalam situasi-situasi yang dimana karakter sulit untuk dikenali. Pengaplikasian OCR sendiri memungkinkan komputer untuk melakukan proses lebih lanjut, contohnya translasi ke bahasa asing, pencarian, sistem baca otomatis untuk orang tunanetra, input data, pengenalan karakter seperti plat nomor, pengetesan CAPTCHA, atau masalah teks lainnya. Tesseract OCR adalah library yang digunakan untuk pemrosesan citra biner menjadi teks, metode yang digunakan adalah dengan Analisa distribusi piksel pada citra untuk mengenali karakter.

OCR dapat dipandang sebagai bagian dari pengenalan otomatis yang lebih luas yakni pengenalan pola otomatis (*automatic pattern recognition*). Dalam pengenalan pola otomatis, sistem pengenalan pola mencoba mengenali apakah citra masukan yang diterima cocok dengan salah satu citra yang telah ditentukan.

2.3.1 Segmentasi Citra

Segmentasi merupakan bagian penting dalam pengolahan citra digital. Proses segmentasi bertujuan untuk menemukan objek di dalam *frame* gambar. Analisis lebih lanjut akan dilakukan pada objek tersebut dengan tujuan pendeteksian atau

pengenalan objek. Warna yang tertangkap oleh kamera tidak saja berasal dari objek yang dikehendaki, tetapi dapat juga berasal dari objek lain yang merupakan *background*. Analisis pada citra digital tentu saja hanya akan berfokus pada *object of interest* sehingga diperlukan proses untuk mendapatkan informasi objek tersebut dan mengabaikan informasi *background* [6].

Pada umumnya *output* atau keluaran dari hasil segmentasi citra adalah berupa citra biner dimana objek (*foreground*) yang dikehendaki berwarna putih (1), sedangkan *background* yang ingin dihilangkan berwarna hitam (0). Segmentasi adalah proses membagi citra digital menjadi beberapa daerah atau kelompok, di mana masing masing daerah terdiri dari sekumpulan piksel. Segmentasi merupakan suatu bagian yang sangat penting dalam analisis citra secara otomatis, sebab pada prosedur ini obyek yang diinginkan akan disadap untuk proses selanjutnya, misalnya: pada pengenalan pola [9].

2.4 Penelitian Terkait

Tabel 2.1 Penelitian Terkait

No.	Nama Peneliti	Judul Penelitian	Metodologi Penelitian	Hasil Penelitian
1.	Didit Meidi (2018)	Implementasi OCR (<i>Optical Character Recognition</i>) Menggunakan Metode <i>Otsu Threshold</i> untuk mendeteksi Tajwid Al-Qur'an	Pengujian dari aplikasi pendeteksian tajwid menggunakan metode OCR dan algoritma <i>Otsu Threshold</i> yang dapat membaca tajwid secara visual dengan	Dari penelitian yang dilakukan aplikasi pendeteksi tajwid Al-Qur'an yang menggunakan OCR dan algoritma <i>otsu threshold</i> dengan

			memanfaatkan fitur kamera pada gadget	data uji sebanyak 200 data diperoleh persentase sebesar 66.5%
2.	Eka Ardianto (2011)	Rancang Bangun Aplikasi Pengolah Gambar Digital untuk Segmentasi Otomatis Lokasi Objek Angka pada Meter Listrik	Penelitian ini melakukan pensegmentasian lokasi angka meter listrik. Penggunaan algoritma dirancang untuk mendapatkan lokasi angka dengan menggunakan fungsi regionprops pada lingkungan MATLAB.	Hasil akhir yang diperoleh adalah sebesar 47% dari sample yang diambil ternyata dapat menemukan lokasi angka meter dengan baik.
3.	Andi Sudiarso dan Rierien J. Merischaputri (2014)	<i>Back Propagation Neural Network Approach for Electricity Usage Meter Numeral Recognition</i>	Pengimplementasian dan perhitungan akurasi dengan menggunakan metode jaringan syaraf tiruan menggunakan algoritma back propagation melalui pengenalan angka pembacaan meter pemakaian listrik.	Dari 33 sampel yang diuji, algoritma dapat bekerja dengan akurasi 100% setelah beberapa proses iterasi. Hasilnya menunjukkan bahwa dengan bertambahnya jumlah variasi sampel, semakin banyak akurasi yang dicapai oleh

				jaringan yang dilatih.
4.	Yusran Mansyur (2018)	<i>Optical Character Recognition</i> Untuk Deteksi Pelat Mobil dan Motor Kendaraan Pada Kampus Teknik Gowa	Metode penelitian yang digunakan adalah <i>Local Binary Pattern Cascade Classifier</i> untuk mendeteksi pelat dan <i>Optical Character Recognition (OCR)</i> untuk rekognisi teks pada pelat kendaraan.	Berdasarkan penelitian yang telah dilakukan, penelitian menunjukkan bahwa tingkat akurasi pendeteksian dan rekognisi pelat kendaraan area kampus Teknik Unhas mampu mencapai akurasi sebesar 80.9%.
5.	Awang Hendrianton Pratomo (2018)	Implementasi <i>Algoritma Region Of Interest (ROI)</i> untuk Meningkatkan Performa Algoritma Deteksi dan Klasifikasi Kendaraan	Metode yang digunakan dalam penelitian ini menggunakan <i>Region Of Interest (ROI)</i> merupakan cara yang tepat untuk mengurangi tingginya waktu pemrosesan. ROI mampu menandai area tertentu sehingga dapat digunakan untuk mrngoptimalisasikan	Hasil yang diperoleh adalah dengan menggunakan ROI waktu pemrosesan citra menggunakan metode segmenteasi MOG2 dan tracking dapat lebih cepat dibandingkan dengan waktu pemrosesan

			kinerja sistem untuk mendeteksi area.	ketika tidak menggunakan ROI.
--	--	--	---------------------------------------	-------------------------------

BAB 3

METODOLOGI PENELITIAN

3.1 Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Sistem Kendali dan Instrumentasi, Fakultas Teknik Universitas Hasanuddin, Gowa, Sulawesi Selatan. Waktu penelitian dimulai pada bulan Februari 2021 sampai dengan bulan Oktober 2021.

3.2 Peralatan Penelitian

3.2.1 Kebutuhan Perangkat Keras (*Hardware*)

Dalam membuat aplikasi pembacaan angka meteran listrik dibutuhkan beberapa peralatan yang akan menunjang dalam proses penelitian, yaitu:

1. Laptop
2. Kamera Ponsel

Peneliti menggunakan kamera ponsel Samsung dengan resolusi 8 MP.

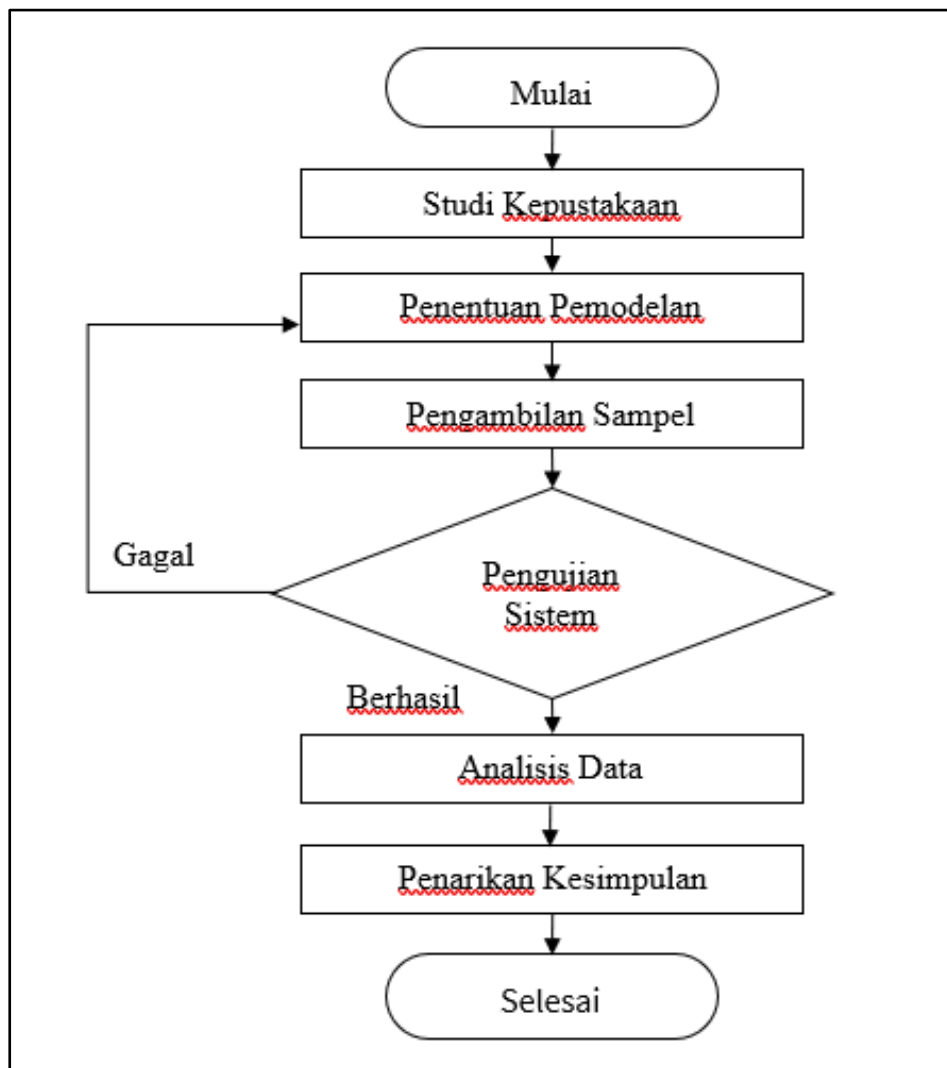
3.2.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian untuk memproses citra digital, yaitu:

1. Matlab R2016a
2. Microsoft Windows 10
3. CorelDRAW X8

3.3 Tahapan Penelitian

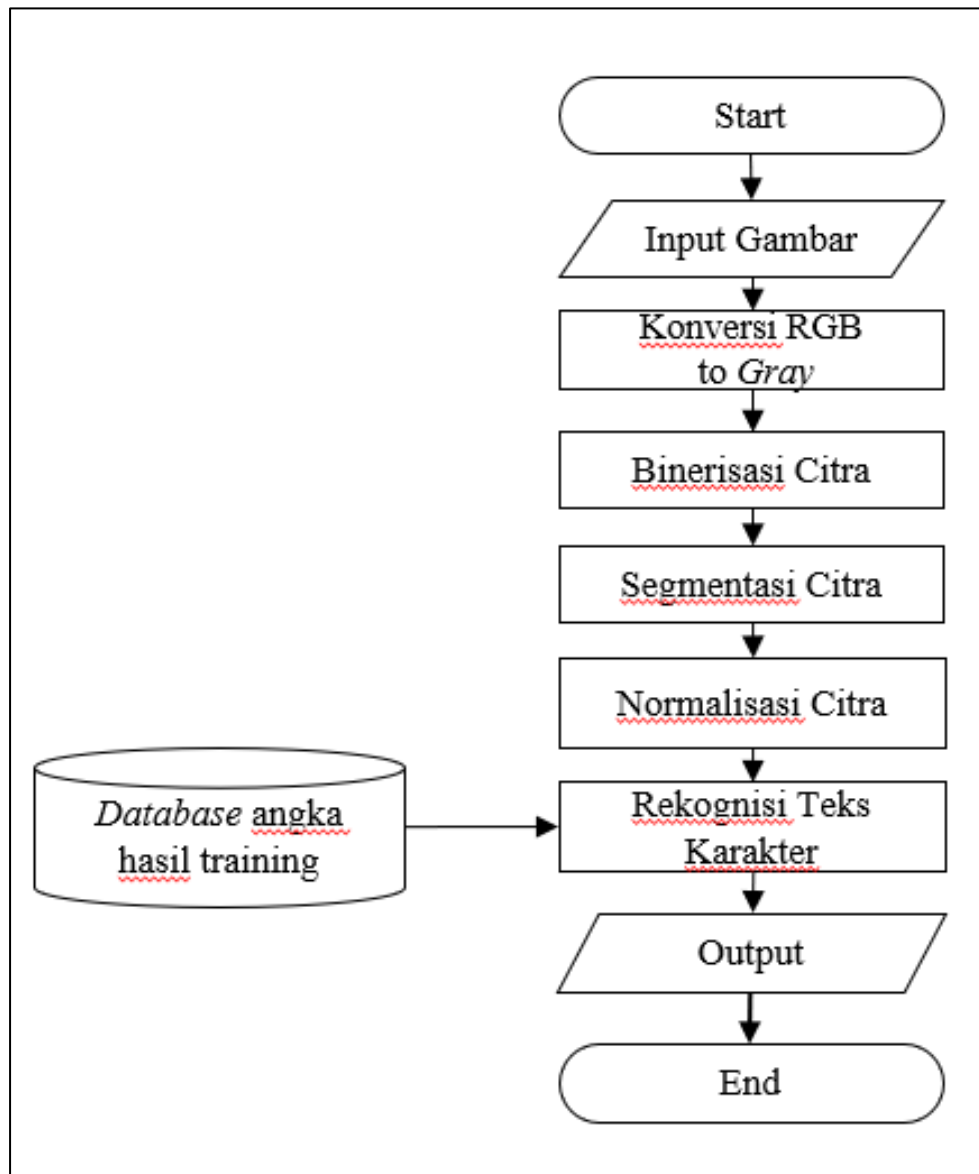
Adapun langkah-langkah dalam penelitian ini digambarkan dalam diagram alir (*flowchart*) dibawah ini:



Gambar 3.1 Diagram Alir Penelitian

3.4 Proses Kerja Sistem

Perancangan sistem pembacaan angka meteran listrik pascabayar ini terdiri dari tiga tahap yaitu masukan gambar (*input*), pemrosesan (*preprocessing*), dan hasil keluaran (*output*). Diagram alir mengenai proses pengenalan angka dapat dilihat pada gambar di bawah ini.



Gambar 3.2 Diagram alir proses pengenalan angka

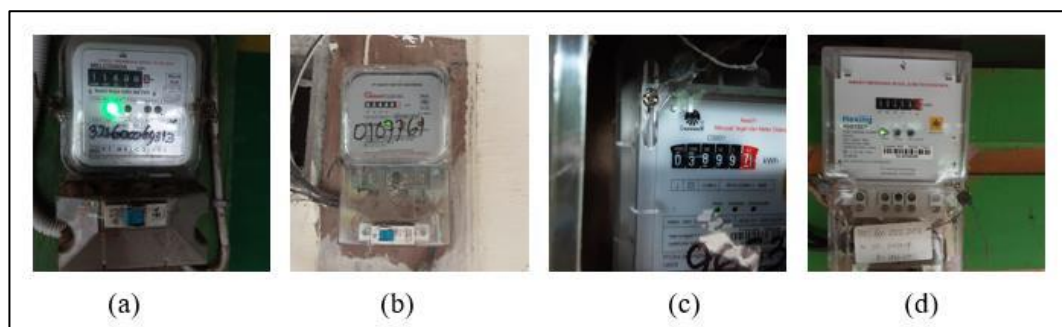
Pada gambar di atas dapat dilihat bahwa proses pengambilan citra dilakukan dengan menggunakan kamera ponsel, setelah itu citra diproses lebih lanjut di dalam sistem yang telah dibuat. Sistem pembacaan angka meteran listrik ini dikerjakan dengan menggunakan program Matlab versi R2016a.

Salah satu proses terpenting dari penelitian ini adalah membuat data referensi atau proses *training* angka dengan menggunakan fitur *ocrTrainer* pada matlab.

Penggabungan beberapa angka dengan *font* yang berbeda dilakukan menggunakan CorelDraw. Selanjutnya hasil training akan tersimpan di *database* yang nantinya akan dipanggil untuk dicocokkan dengan karakter yang telah tersegmentasi.

3.4.1 Jenis Meteran yang Digunakan

Pengujian terhadap sistem yang telah dikembangkan akan menggunakan 4 jenis meteran listrik yang berbeda dan umum digunakan oleh masyarakat. Dari setiap meteran listrik tersebut dilakukan beberapa kali pengambilan data sehingga total citra yang akan digunakan sebagai sampel pengujian adalah 22 citra.



Gambar 3.3 Jenis jenis meteran

Jenis meteran yang digunakan sebagai sampel penelitian diantaranya:

- a. Melcoinda
- b. SmartMeter
- c. Sanxing
- d. Hexing

3.4.2 Proses Akuisisi Citra

Akuisisi citra adalah proses pengubahan citra analog menjadi citra digital yang diambil dari lingkungan atau dunia nyata menggunakan beberapa alat berikut,

kamera digital, *webcam*, *smartphone*, *scanner*, dan lain-lain, agar bisa dilanjutkan ke tahap *preprocessing* [6].

Berikut merupakan langkah awal dari sistem yaitu proses memasukan gambar ke dalam program. Inputan yang digunakan dalam sistem ini adalah citra meteran listrik yang diambil dengan menggunakan kamera ponsel yang mempunyai resolusi 8 MP.

```
% --- Executes on button press in btnimage.
function btnimage_Callback(hObject, eventdata, handles)
[nama_file,nama_folder] = uigetfile({'*.jpg'; '*.png'; '*.bmp'}, 'Add Image');
% jika ada nama file yang dipilih maka akan mengeksekusi perintah di bawah
% ini
if ~isequal(nama_file,0)
    % membaca file citra rgb
    I = imread(fullfile(nama_folder,nama_file));
    % menampilkan citra rgb pada axes
    axes(handles.axes1)
    imshow(I)
    title('Original Image')

    % menampilkan nama file citra pada edit text
    set(handles.editid, 'string', nama_file)

    % menyimpan variabel I pada lokasi handles
    handles.I = I;
    guidata(hObject, handles)
else
    % jika tidak ada file yang dipilih maka akan kembali
    return
end
```

Dapat dilihat bahwa hanya file yang berformat JPG, PNG dan BMP yang dapat diproses oleh sistem. Langkah berikutnya adalah proses memotong bagian meteran yang akan diproses lebih lanjut dengan menggunakan perintah *crop* sehingga sistem lebih mudah mengubah gambar tersebut menjadi sebuah teks lebih cepat dan hasil akhirnya dapat dilihat pada gambar di bawah ini.

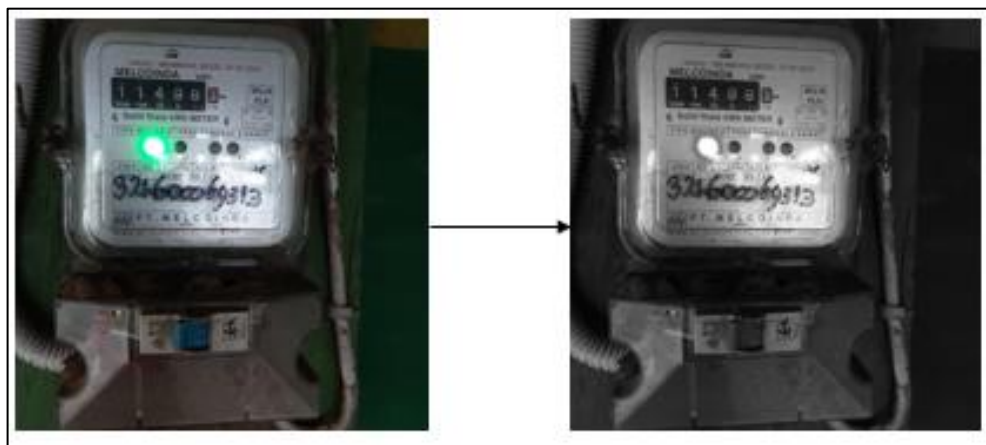


Gambar 3.4 Contoh Hasil dari Akuisisi Citra

Tujuan dari proses ini adalah untuk mendapatkan citra dari meteran listrik. Proses ini merupakan proses awal dari sistem pengenalan angka. Posisi kamera harus sejajar dengan bagian depan citra, pencahayaan yang cukup dan tidak ada bayangan (*noise*) yang dapat menghalangi angka sehingga dapat terbaca dengan baik oleh sistem. Dalam proses akuisisi citra peneliti menggunakan ponsel dengan resolusi 8 MP untuk keseluruhan sampel yang diambil.

3.4.3 Proses Grayscale

Tahap ini merupakan tahap konvergi citra RGB menjadi citra *grayscale*. Tujuannya yaitu untuk mereduksi ukuran dari citra. Berikut merupakan citra perbedaan warna dari citra RGB dan juga citra *ayscale*. Citra dari meteran diubah menjadi 200x600 pixel untuk mempermudah proses selanjutnya.



Gambar 3.5 Perbedaan Warna RGB (kiri) dan Gray (kanan)

Ada dua perintah yang dapat digunakan dalam mengonversi citra RGB menjadi citra *grayscale*. Pertama dengan menggunakan fungsi yang disediakan oleh matlab seperti berikut:

```
% konversi RGB to Gray
Img = rgb2gray(img);
Image = imresize(Img,[200 600]);
% figure, imshow(Image);
```

Perintah yang kedua tanpa menggunakan fungsi yang disediakan oleh matlab. Hasil keluaran yang didapatkan sama-sama dapat digunakan, perbedaannya hanya terdapat pada waktu yang dibutuhkan dalam memproses dari citra.

```
R = img(:, :, 1);
G = img(:, :, 2);
B = img(:, :, 3);

Img = zeros(size(img,1), size(img,2), 'uint8');
for x=1:size(img,1)
    for y=1:size(img,2)
        Img(x,y) = (R(x,y)*.2989)+(G(x,y)*.5870)+(B(x,y)*.1140);
    end
end
```

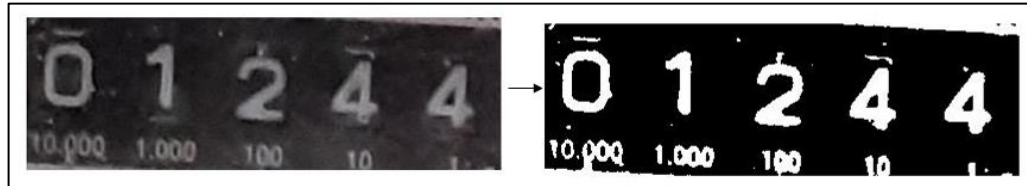
3.4.4 Proses Binerisasi

Tahap binerisasi bertujuan untuk mengetahui daerah yang termasuk objek (*foreground*) dan *background* dari citra. Tahapan ini citra yang berupa input berwarna *grayscale* dikonversi menjadi citra *biner*.

```
% konversi grayscale ke biner
BW1 = imadjust(Img);
BW1 = im2bw(BW1);
figure, imshow (BW1);
```

Berikut merupakan hasil keluaran dari proses binerisasi. Terlihat bahwa masih adanya objek yang terdeteksi angka kecil diluar angka yang ditargetkan,

sehingga tahap selanjutnya adalah pemberian *blob area* terhadap area yang menjadi target.



Gambar 3.6 Hasil dari Proses Binerisasi

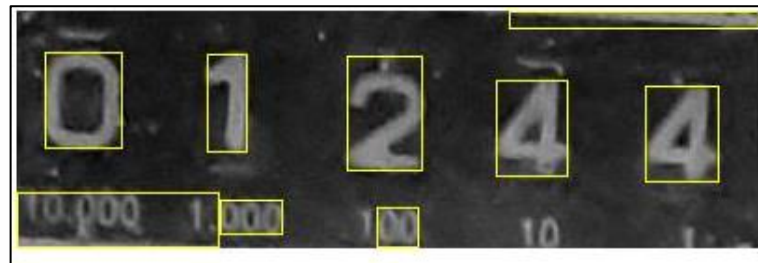
3.4.5 Proses Segmentasi Citra

Proses segmentasi citra dapat dilakukan dengan menggunakan metode *bounding box* yang digunakan untuk mengukur properti dari area citra. Metode ini dapat digunakan untuk permasalahan pendeteksian angka pada meteran listrik. Segmentasi merupakan bagian dari tahap *preprocessing* yang bertujuan untuk memisahkan objek tertentu yang dikehendaki (*foreground*) dengan objek lain yang tidak dikehendaki (*background*).

Segmentasi menggunakan metode *thresholding* yang banyak dipilih karena umumnya objek memiliki nilai intensitas yang menonjol dibandingkan dengan *background*-nya. Nilai yang sangat berbeda antara objek dan *background* inilah yang kemudian digunakan untuk membedakan keduanya berdasarkan suatu batas nilai *threshold*. Batas tersebut menjadikan sebuah piksel bernilai 0 atau 1.

```
blobAnalyzer = vision.BlobAnalysis('MaximumCount', 50);  
[area, centroids, roi] = step(blobAnalyzer, BW1);  
  
areaConstraint = area > 250;  
roi = double(roi(areaConstraint, :));  
cit_area_250px = bwareaopen(BW1, 250);  
  
Imgroi = insertShape(Img, 'rectangle', roi);  
figure, imshow(Imgroi);
```

Dalam menyeleksi bagian objek yang diinginkan maka diberikanlah perbandingan rasio dan mengolah *Region Of Interest* (ROI) dengan menampilkan data ROI yang hanya di atas 250 *pixel*.



Gambar 3.7 Citra dengan Menggunakan Bounding Box

Dari gambar di atas masih terlihat bahwa area di luar angka yang dibutuhkan masih terdeteksi sehingga untuk mengatasi hal tersebut dapat dilakukan penentuan nilai pada *aspectRatio*. Nilai *aspectRatio* didapatkan dari pembagian lebar (*width*) dengan tingginya (*height*).

```
width = roi(:,3);
height = roi(:,4);
aspectRatio = width ./ height;

roi = roi(aspectRatio > 0.25 & aspectRatio < 1, :);
[yroi, xroi] = size(roi);
mask = zeros(size(BW1));

for mulaiyroi = 1:yroi
    mask(roi(mulaiyroi,2):roi(mulaiyroi,2)+roi(mulaiyroi,4)-1,...
        roi(mulaiyroi,1):roi(mulaiyroi,1)+roi(mulaiyroi,3)-1) = 1;
end

Img_baru = logical(BW1) .* logical(mask);
figure, imshow(Img_baru);
citraakhir = bwareaopen(Img_baru, 250);
```

Berikut citra dengan *aspectRatio* yang telah ditentukan yaitu lebih besar dari 0.25 dan lebih kecil dari 1. Cara menghitung nilai *aspectRatio* secara manual adalah dengan membagikan nilai lebar (*width*) dengan tingginya (*height*). Hasil akhirnya

terlihat bahwa tidak adalagi objek yang terdeteksi di luar angka meteran yang diharapkan. Dengan seperti ini sistem dapat dengan mudah mencocokkan karakter yang berada di *database* dengan angka pada meteran listrik.



Gambar 3.8 Citra Setelah Penentuan Nilai aspectRatio

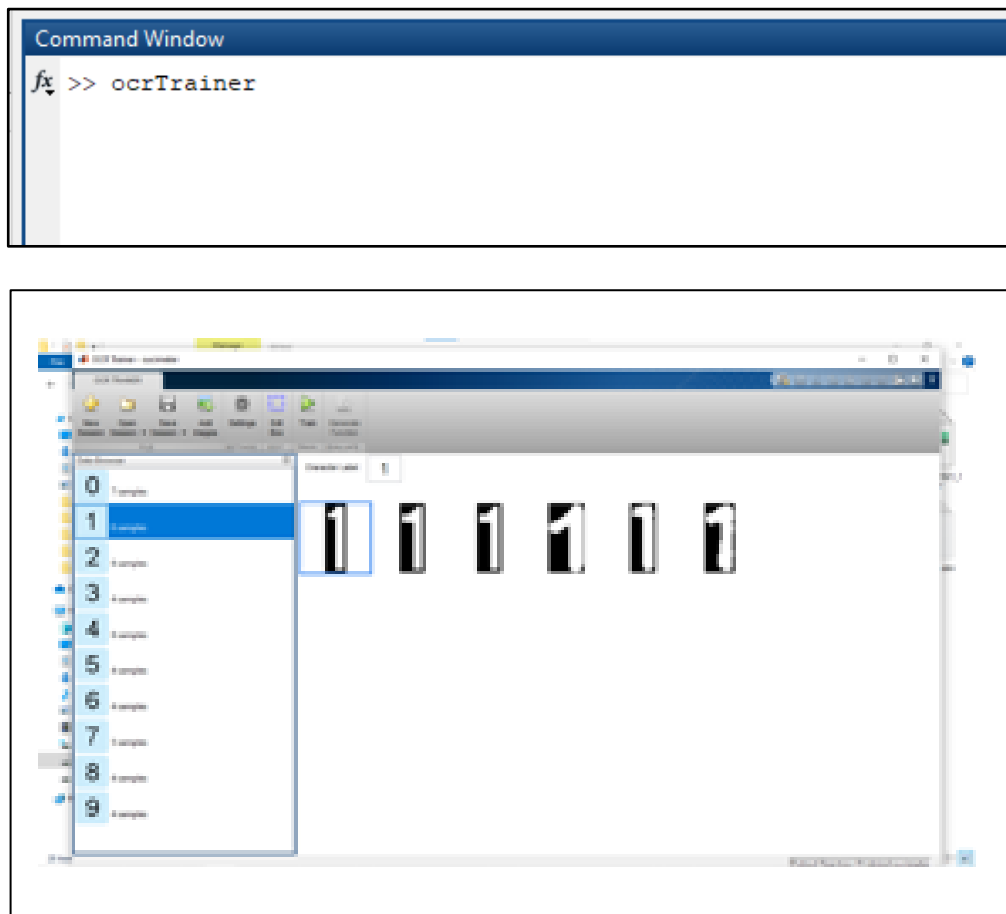
3.4.6 Proses Training Angka

Setelah proses segmentasi akan didapatkan potongan-potongan karakter, selanjutnya potongan-potongan karakter ini akan dikenali sebagai karakter tertentu. Ada banyak cara atau metode yang dapat digunakan untuk mengenali karakter yang sudah disegmentasi [14]. Pada penelitian ini digunakan dua jenis *font* yang akan digunakan dalam pengenalan karakter yaitu Bebas Neue dan Plat Nomor.



Gambar 3 9 Font yang digunakan dalam proses training angka

Dalam proses rekognisi terlebih dahulu menyediakan data training. Pada penelitian ini, peneliti menggunakan aplikasi CorelDRAW untuk menyatukan angka dengan *font* yang berbeda dan menjadikannya sebuah gambar. Selanjutnya pada *command window* MATLAB, diketikkan perintah *ocrTrainer* kemudian tekan *enter* sehingga muncul perintah berikutnya untuk memulai proses training.



Gambar 3.10 Proses Training Angka

Simpan *file training* bersama dengan folder program utama dari sistem pengenalan angka. Pada tahap selanjutnya sistem akan melakukan proses *Character Classification* yaitu menggabungkan dan mengklasifikasikan hasil ROI *pre-processing* dengan *file font* yang telah dilatih dengan fitur *ocrTrainer* pada matlab.

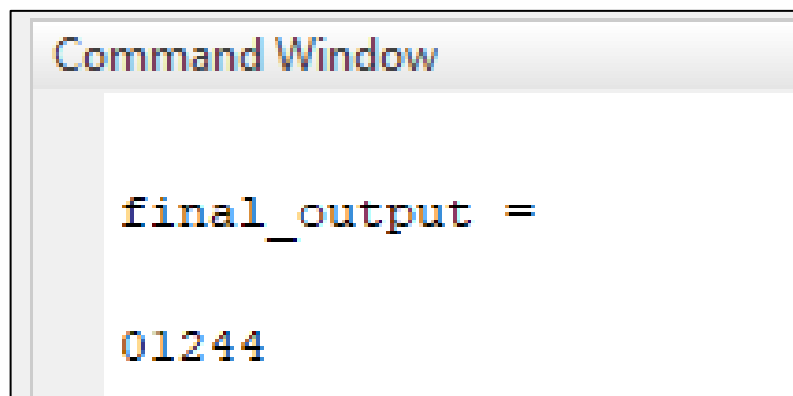
3.4.7 Proses Rekognisi Teks Karakter

Selanjutnya dilakukan proses rekognisi teks, dimana sistem akan melakukan *reading symbols* yaitu membaca simbol-simbol menjadi karakter dari angka meteran listrik. Tahap terakhir dari sistem ini yaitu menampilkan karakter hasil pemrosesan sebelumnya. Pada umumnya operator PLN melakukan pencatatan pada 5 (lima) angka pertama.

```
% Rekognisi karakter teks
results = ocr(citraakhir, 'Language', ...
    {'D:\skripsi\myLang\tessdata\myLang.traineddata'});

% Buat Var Final_Output
final_output = [];
final_output = [final_output deblank(results.Text)];

% Hilangkan spasi pada teks
final_output = final_output(~isspace(final_output))
```



Gambar 3.11 Hasil dari Rekognisi Teks

Ciri atau fitur merupakan segala sesuatu yang bisa dijadikan sebagai aspek pembeda antara satu objek dengan objek yang lain. Sedangkan pola adalah gabungan dari beberapa ciri yang menjadi sifat sebuah objek. Pada penelitian ini pola yang harus dikenali oleh sistem adalah pola angka dengan beberapa ciri dari

masing-masing angka. Misalnya angka 1 dan 0 memiliki ciri yang sangat berbeda sehingga dalam pengenalan polanya dapat lebih mudah.

Untuk mengukur atau mengetahui waktu yang digunakan sistem dalam memproses sebuah gambar dapat digunakan perintah *cputime*. Waktu yang didapatkan dalam satuan detik. Untuk mengukur waktu yang digunakan untuk menjalankan sebuah kode, dilakukan dengan menggunakan atau menjalankan dua kode yaitu *cputime* sebelum dan sesudah kode kemudian akan terhitung perbedaan waktu keluarannya. Waktu yang dibutuhkan sistem dalam memproses citra dapat dilihat dengan menggunakan perintah berikut:

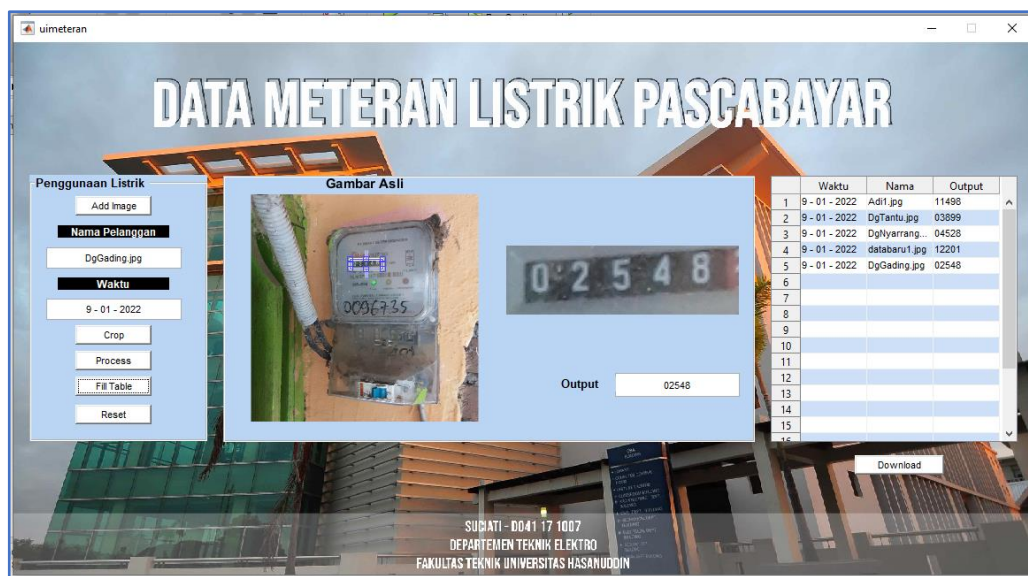
```
% Lamanya sistem memproses
waktu_selesai = cputime;
waktu_proses = waktu_selesai - waktu_mulai;
```

BAB 4

HASIL DAN PEMBAHASAN

4.1 Perancangan Tampilan *Graphical User Interface* (GUI)

Tampilan *interface* pada perancangan sistem pembacaan angka menggunakan GUI (*Graphical User Interface*) matlab. Hal ini bertujuan untuk membantu dalam proses pembacaan angka meteran listrik.



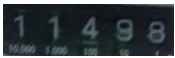
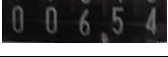
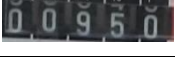
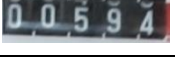
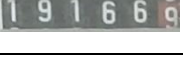
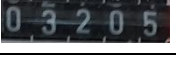
Gambar 4.1 Tampilan GUI dari proses OCR

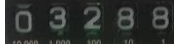
Alur kerja dari GUI pada gambar di atas dimulai dari *Adda Image* yaitu inputan yang berupa gambar kWh meter yang telah diambil, kemudian melakukan perintah *crop* untuk mendapatkan area yang lebih spesifik. Selanjutnya ketika menekan tombol *process*, gambar dari hasil *crop* sebelumnya akan diproses dan keluarannya dapat terlihat pada kotak yang berada di bawahnya. Fungsi tabel adalah menampung semua data dari Nama Pelanggan, Waktu dan Output. Setelah semua proses selesai, maka data dapat di download dan hasil akhirnya berupa file dengan format Excel.

4.2 Hasil Pengujian Sistem

Terdapat 22 gambar meteran listrik pascabayar yang telah diuji coba dengan sistem pembacaan angka meteran listrik, dari 22 gambar ada 4 gambar yang gagal terbaca oleh sistem. Berikut tabel hasil pengujian pembacaan angka pada meteran listrik pascabayar.

Tabel 4.1 Hasil Pembacaan Meteran Listrik Pascabayar

No.	Gambar Meteran	Hasil Pembacaan	Akurasi
1		11498	Benar
2		02548	Benar
3		04528	Benar
4		03448	Benar
5		03899	Benar
6		01244	Benar
7		02748	Benar
8		0712	Salah
9		00654	Benar
10		316	Salah
11		00950	Benar
12		0594	Salah
13		10224	Benar
14		19166	Benar
15		205	Salah

16		12077	Benar
17		01802	Benar
18		01815	Benar
19		12201	Benar
20		06553	Benar
21		03288	Benar
22		24295	Benar
Rata-rata persentase keberhasilan			81.81%

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah Meteran Benar}}{\text{Jumlah Meteran Keseluruhan}} \times 100\% \\
 &= \frac{18}{22} \times 100\% = 81.81\%
 \end{aligned}$$

Dari tabel di atas dapat dilihat bahwa perbedaan akurasi terhadap angka pada meteran listrik pascabayar dengan jenis meteran yang sama dapat disebabkan oleh beberapa hal diantaranya meteran yang sudah lama sehingga terlihat buram dan faktor dari cahaya yang menyebabkan bayangan terlihat ketika melakukan pengambilan gambar.

Data yang dinilai berhasil adalah data yang keseluruhan angkanya dapat terbaca oleh sistem, sedangkan data yang gagal adalah data yang salah satu atau keseluruhan angka tidak terbaca oleh sistem. Metode *Optical Character Recognition* (OCR) ini sangat bergantung terhadap tingkat kecerahan dari citra dan posisi angka yang nantinya akan direkognisi. OCR dapat merekognisi karakter yang ada pada

meteran listrik apabila data yang ada pada sistem sama dengan citra biner yang telah diolah, sehingga pengolahan citra dari citra berwarna menjadi citra biner menjadi bagian yang penting.



Gambar 4.2 Contoh meteran listrik yang gagal terbaca

Gambar tersebut gagal terbaca dikarenakan terdapat bayangan pada angkanya sehingga kedua angka nol tersebut terbaca oleh sistem satu kesatuan. Kemudian di dalam *database* tidak terdapat data angka yang cocok dari hasil training, inilah mengapa meteran tersebut gagal untuk terbaca oleh sistem.

Dari penelitian yang dilakukan didapatkan lamanya sistem memproses adalah selama 1.0469 detik.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang dilakukan pada sistem pembacaan angka pada meteran listrik pascabayar, kesimpulan yang didapatkan pada penelitian ini adalah:

1. Algoritma *Optical Character Recognition* (OCR) merupakan metode yang dapat digunakan dalam pembacaan angka meteran listrik pascabayar. Dapat dilihat dari hasil pengujian dengan persentase keberhasilan rata-rata 81,81%.
2. Intensitas cahaya dan bayangan yang terlihat pada meteran sangat mempengaruhi dalam proses rekognisi. Angka yang tidak tertampil secara penuh atau hanya setengah pada meteran listrik juga menjadi salah satu penyebab kegagalan dalam pembacaan.
3. Sistem mampu menyimpan data dari *Graphical User Interface* (GUI) ke dalam bentuk excel.

5.2 Saran

Hasil penelitian ini masih memiliki beberapa kekurangan sehingga dapat dikembangkan lebih lanjut dengan saran-saran sebagai berikut.

1. Agar citra yang diambil dapat terbaca dengan baik, gunakan kamera yang memiliki resolusi tinggi.
2. Memperbanyak data training dengan *font* yang berbeda serta keadaan angka yang berbeda, sehingga pembacaan dapat lebih akurat.

3. Diperlukan pengujian dalam pembacaan dengan menggunakan metode atau algoritma lain untuk mengetahui perbandingan tingkat akurasi dalam pembacaan angka.

DAFTAR PUSTAKA

- [1] Nuranita. 2017. “*Analisa Perbandingan Kwh Meter Prabayar Dengan Kwh Meter Non Prabayar Dari Segi Keekonomisannya,*” Jurnal Mahasiswa Sekolah Tinggi Teknik Medan, Vol. 11
- [2] Hidayatullah, P., 2016. *License plate detection and recognition for Indonesian cars.* 8 (2). Pp. 331-346.
- [3] Mohammad, F., Anarase, J., Shingote, M., & Ghanwat, P. (2014). *Optical Character Recognition Implementation Using Pattern Matching.* International Journal of Computer Science and Information Technologies, 2088-2090
- [4] Jayyid, Unzhil Latif. 2016. *Analisis Penggunaan kWh Meter Pascabayar dan kWh Meter Prabayar 1 Fasa di PT. PLN (Persero).* Medan: Universitas Sumatera Utara
- [5] Andono, Pulung Nurtantio dkk. 2017. *Pengolahan Citra Digital.* Yogyakarta: Penerbit ANDI
- [6] Fitriyah, H., Wihandika, R.C. 2021. *Dasar-dasar Pengolahan Citra Digital: warna, segmentasi, dan filter.* Malang: UB Press
- [7] Maula, Akhmad Robit dkk. 2013. *Optical Character Recognition Dengan Metode Naïve Bayes.* Malang: Universitas Brawijaya
- [8] Trilaksono, Mirza dkk. *Implementasi Optical Character Recognition (OCR) Dengan Pendekatan Metode Struktur Menggunakan Ekstraksi Ciri Vektor Dan Region.* Bandung: Universitas Telkom.

- [9] Ardianto, E., Lusiana, V., & Hadikurniawati, W. 2011. *Rancang Bangun Aplikasi Pengolah Gambar Digital untuk Segmentasi Otomatis Lokasi Objek Angka pada Meter Listrik*. Jurnal Teknologi DINAMIK Volume 16, 110-117
- [10] Meidi, Didit. 2018. *Implementasi OCR (Optical Character Recognition) Menggunakan Metode Otsu Treshold untuk Mendeteksi Tajwid Al-Qur'an*. Bandung: Universitas Islam Negeri Sunan Gunung Djati
- [11] Sudiarso, A., & Merischaputri R.J. (2014). *Back Propagation Neural Network Approach for Electricity Usage Meter Numeral Recognition*. International Journal of Mining, Metallurgy & Mechanical Engineering (IJMMME) Vol. 2, 2320-4060
- [12] Mansyur, Yusran. 2018. *Optical Character Recognition Untuk Deteksi Pelat Mobil dan Motor Kendaraan Pada Kampus Teknik Gowa*. Makassar: Universitas Hasanuddin
- [13] Pratomo, A.W., Kaswidjanti, W., & Mu'arifah S. 2020. *Implementasi Algoritma Region Of Interest (ROI) untuk Meningkatkan Performa Algoritma Deteksi dan Klasifikasi Kendaraan*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK) Vol. 7, 155-162

LAMPIRAN

Lampiran 1. List Gambar Meteran Listrik

No.	Gambar Meteran	No.	Gambar Meteran
1.		12.	
2.		13.	
3.		14.	
4.		15.	
5.		16.	

6.		17.	
7.		18.	
8.		19.	
9.		20.	
10		21.	
11.		22.	

Lampiran 2. Source Code

```
function varargout = uimeteran(varargin)
% UIMETERAN MATLAB code for uimeteran.fig
%     UIMETERAN, by itself, creates a new UIMETERAN or raises the
existing
%     singleton*.
%
%     H = UIMETERAN returns the handle to a new UIMETERAN or the
handle to
%     the existing singleton*.
%
%     UIMETERAN('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in UIMETERAN.M with the given input
arguments.
%
%     UIMETERAN('Property','Value',...) creates a new UIMETERAN
or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before uimeteran_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to uimeteran_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help uimeteran

% Last Modified by GUIDE v2.5 27-Oct-2021 15:08:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @uimeteran_OpeningFcn, ...
                  'gui_OutputFcn',  @uimeteran_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

% End initialization code - DO NOT EDIT

% --- Executes just before uimeteran is made visible.
function uimeteran_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to uimeteran (see VARARGIN)

% Choose default command line output for uimeteran
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
ah = axes ('unit', 'normalized', 'position', [0 0 1 1]);
bg = imread ('bgteknik.jpg'); imagesc(bg);
set(ah, 'handlevisibility','off','visible','off')

% UIWAIT makes uimeteran wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = uimeteran_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in btnimage.
function btnimage_Callback(hObject, eventdata, handles)
[nama_file,nama_folder] = uigetfile({'*.jpg'; '*.png'; '*.bmp'},
'Add Image');
% jika ada nama file yang dipilih maka akan mengeksekusi perintah
di bawah
% ini
if ~isequal(nama_file,0)
    % membaca file citra rgb
    I = imread(fullfile(nama_folder,nama_file));
    % menampilkan citra rgb pada axes
    axes(handles.axes1)
    imshow(I)
    title('Original Image')

```

```

        % menampilkan nama file citra pada edit text
        set(handles.editid, 'string', nama_file)

        % menyimpan variabel I pada lokasi handles
        handles.I = I;
        guidata(hObject, handles)
    else
        % jika tidak ada file yang dipilih maka akan kembali
        return
    end

% --- Executes on button press in btncrop.
function btncrop_Callback(hObject, eventdata, handles)
% hObject      handle to btncrop (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
citra = handles.I;
axes(handles.axes1)
imshow(citra)
title('Gambar Asli');

h = imrect;
position = wait (h);
hasil1 = imcrop(citra, position);
handles.hasil=hasil1;
axes(handles.axes2)
imshow(hasil1);

% menyimpan variabel hasil1 pada lokasi handles
handles.hasil1 = hasil1;
guidata(hObject, handles)

% --- Executes on button press in btnproses.
function btnproses_Callback(hObject, eventdata, handles)
% hObject      handle to btnproses (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
hasil1 = handles.hasil1;
% konversi RGB ke Grayscale
Img = rgb2gray(hasil1);
% Image = imresize(Img,[200 600]);
% figure, imshow(Image);

% menguatkan citra gelap
BW1 = imadjust(Img);
BW1 = im2bw(BW1);
% figure, imshow (BW1);

blobAnalyzer = vision.BlobAnalysis('MaximumCount', 50);
[area, centroids, roi] = step(blobAnalyzer, BW1);

```

```

areaConstraint = area > 250;
roi = double(roi(areaConstraint, :));
cit_area_250px = bwareaopen(BW1, 250);

Imgroi = insertShape(Img, 'rectangle', roi);
% figure, imshow(Imgroi);

width = roi(:,3);
height = roi(:,4);
aspectRatio = width ./ height;

roi = roi(aspectRatio > 0.25 & aspectRatio <1,:);
[yroi, xroi] = size(roi);
mask = zeros(size(BW1));

for mulaiyroi = 1:yroi
    mask(roi(mulaiyroi,2):roi(mulaiyroi,2)+roi(mulaiyroi,4)-1,...
        roi(mulaiyroi,1):roi(mulaiyroi,1)+roi(mulaiyroi,3)-1) = 1;
end

Img_baru = logical(BW1) .* logical(mask);
% figure, imshow(Img_baru);

citraakhir = bwareaopen(Img_baru, 250);

% Rekognisi karakter teks
results = ocr(citraakhir, 'Language', ...
    {'D:\skripsi\sucimeter\tessdata\sucimeter.traineddata'});

% Buat Var Final_Output
final_output = [];
final_output = [final_output deblank(results.Text)];

% Hilangkan spasi pada teks
final_output = final_output(~isspace(final_output))
set(handles.editangka, 'String', num2str(final_output))

% --- Executes on button press in btntable.
function btntable_Callback(hObject, eventdata, handles)
% hObject      handle to btntable (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

A = get(handles.editwaktu, 'String');
B = get(handles.editid, 'String');
C = get(handles.editangka, 'String');
Data = {A,B,C};
OldData = get(handles.uitable, 'Data');

% check empty data
r = size(OldData,1);

```

```

brk = false;
for m=1:r
    x = OldData{m,1};
    k = m;
    brk = true;
    if isempty(x)
        break;
    end
end

if (k==r && brk==false)
    NW_Data = [OldData;Data];
else
    NW_Data = OldData;
    NW_Data(k,:) = Data;
end
set(handles.uitable, 'Data', NW_Data);

% --- Executes on button press in btnreset.
function btnreset_Callback(hObject, eventdata, handles)
% hObject    handle to btnreset (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% mereset GUI
set(handles.editid, 'String', [])
set(handles.editwaktu, 'String', [])
set(handles.editangka, 'String', [])
set(handles.uitable, 'Data', [])

axes(handles.axes1)
cla reset
set(gca, 'Xtick', [])
set(gca, 'Ytick', [])

axes(handles.axes2)
cla reset
set(gca, 'Xtick', [])
set(gca, 'Ytick', [])

function editid_Callback(hObject, eventdata, handles)
% hObject    handle to editid (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of editid as text
%        str2double(get(hObject, 'String')) returns contents of
editid as a double

```

```

% --- Executes during object creation, after setting all
properties.
function editid_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editid (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function editwaktu_Callback(hObject, eventdata, handles)
% hObject    handle to editwaktu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editwaktu as
text
%         str2double(get(hObject,'String')) returns contents of
editwaktu as a double

```

```

% --- Executes during object creation, after setting all
properties.
function editwaktu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editwaktu (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function editangka_Callback(hObject, eventdata, handles)
% hObject    handle to editangka (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of editangka as
text
%         str2double(get(hObject,'String')) returns contents of
editangka as a double

% --- Executes during object creation, after setting all
properties.
function editangka_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editangka (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btndownload.
function btndownload_Callback(hObject, eventdata, handles)
% hObject    handle to btndownload (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
Data =get(handles.uitable,'Data');
HD = {'Waktu','ID','Output'};
SN = ['Data_', datestr(now,'ddmmyyyy_HHMMSS'),'xlsx'];

xlswrite(SN,[HD;Data]);
msgbox('Success!','Report');

```