



ISSN : 2339 - 1871

JURNAL ILMIAH BETRIK

Besemah Teknologi Informasi dan Komputer

Editor Office : LPPM Sekolah Tinggi Teknologi Pagar Alam, Jln. Masik Siagim No. 75
Simpang Mbacang, Pagar Alam, SUM-SEL, Indonesia
Phone : +62 852-7901-1390.
Email : betrik@sttpagaralam.ac.id | admin.jurnal@sttpagaralam.ac.id
Website : <https://ejournal.sttpagaralam.ac.id/index.php/betrik/index>

Implementasi Algoritma Run-Length Smearing Dalam Identifikasi Angka Meter Air Dengan Metode Waterfall

Aris Yolanda¹, Ahmad Syazili²

Program Studi Teknik Informatika Universitas Bina Darma Palembang¹²

Jl. Jendral Ahmad Yani No.3, 9/10 Ulu, Kec.Seberang Ulu I Kota Palembang

Sur-el : aris yolanda15@gmail.com¹, syazili@binadarma.ac.id²

Abstrak: Perusahaan Daerah Air Minum PDAM bertugas mengelola dan menyediakan air bagi masyarakat, air merupakan bagian penting bagi kehidupan manusia untuk memenuhi kebutuhan sehari-hari. Pihak PDAM menggunakan alat meter air untuk mendapatkan jumlah pemakaian air pelanggan. Petugas mencatat langsung angka meter air dari meteran analog kemudian dimasukkan ke dalam aplikasi. Pengambilan foto alat meter air sebagai konfirmasi untuk perhitungan yang mendapatkan biaya yang harus dibayar oleh pelanggan. Proses tersebut akan memakan waktu yang cukup lama dan akan terjadinya kesalahan dalam pencatatan angka meteran air. Sehingga keluhan-keluhan pelanggan pun banyak karena ada beberapa petugas yang melakukan kesalahan dalam proses pencatatan. Solusi yang dapat dilakukan untuk mengatasi lamanya waktu pencatatan dan kesalahan pencatatan angka meter air tersebut dengan cara menggunakan *Run-Length Smearing Algorithm* (RLSA) dengan metode *waterfall* ini akan dapat mengenali pola karakter angka pada citra angka meter air dimana akan mengurangi kesalahan pada saat penginputan data.

Kunci Utama: *Run-Length Smearing Algorithm*, Meter Air, Metode *Waterfall*

Abstract: The PDAM Regional Drinking Water Company is in charge of managing and providing water for the community, water is an important part of human life to meet daily needs. The PDAM uses a water meter to get the amount of water used by customers. The officer directly recorded the water meter number from the analog meter and then entered it into the application. Taking a photo of the water meter as a confirmation for the calculation that gets the cost to be paid by the customer. The process will take a long time and there will be errors in recording water meter numbers. So that there are many customer complaints because there are several officers who make mistakes in the recording process. The solution that can be done to overcome the length of recording time and the problem of recording the water meter number is by using the *Run-Length Smearing Algorithm* (RLSA) with this waterfall method will be able to recognize the pattern of number characters in the water meter number image which will reduce errors when entering data.

Keywords : *Run-Length Smearing Algorithm*, water meter, waterfall method

1. PENDAHULUAN (Font 12)

Air merupakan bagian yang sangat penting bagi kehidupan manusia untuk

memenuhi kebutuhan manusia sehari-hari. Dengan tujuan menyediakan air minum bagi masyarakat, PDAM (Perusahaan Daerah Air Minum) bertugas mengelola dan menyediakan air bagi Masyarakat. Perusahaan Daerah Air Minum (PDAM) untuk mengontrol penggunaan air pelanggannya menggunakan alat pencatat meter air (*water meter reading*) (Pasciana *et al.*, 2019).

Karena sistem lama yang digunakan oleh pihak PDAM untuk mencatat penggunaan air pelanggan, petugas harus mengunjungi rumah masing-masing pelanggan setiap bulan untuk mencatat langsung angka pada meteran analog. Angka-angka yang tertera pada meter air akan diinput ke dalam aplikasi, angka yang dimasukkan selanjutnya akan diproses sebagai dasar tagihan, untuk konfirmasi petugas telah mengunjungi rumah pelanggan petugas mengambil foto meteran air, foto tersebut juga untuk mengoreksi bahwa angka meter air yang dimasukkan ke dalam aplikasi telah sesuai dengan angka pada foto tersebut. Tetapi hal tersebut akan membuat pelanggan sering kali membayar lebih dari yang seharusnya, yang diakibatkan oleh salah dalam penginputan angka (Kabul *et al.*, 2023).

untuk mendapatkan keakuratan data pengolahan citra dengan menerapkan *Run-Length Smearing Algorithm (RLSA)* atau disebut juga *Smearing* dengan menggunakan metode *waterfall* untuk menghindari kesalahan dalam input angka oleh petugas Selain itu proses perhitungan memerlukan waktu yang lama karena jumlah pelanggan yang banyak. Sehingga diambil gambar angka yang ditunjukkan pada meteran air dan dilakukan proses menggunakan RLSA untuk mendeteksi angka pada gambar yang diambil.

Digunakannya algoritma *smearing*, untuk melakukan pemisahan bagian angka volume meteran air pada gambar meteran air dan diharapkan sistem mampu menyelesaikan

masalah dalam deteksi angka volume meteran air yaitu dapat menemukan angka volume meteran air dengan baik. Alasan menggunakan metode *waterfall* ini karena sudah teruji dari penelitian sebelumnya sangat memudahkan penggunaan website dengan penyelesaian sesuai tahapan atau langkah yang sudah disusun mengurut sesuai tahapan metode *waterfall*, karena setiap tahap akan mempengaruhi dan menjadi dasar dari keberlangsungan tahap pengembangan berikutnya.

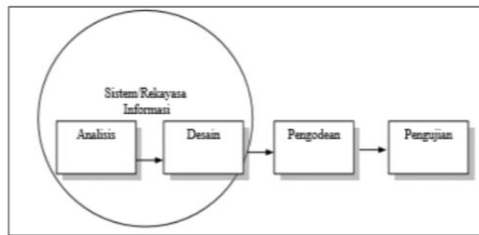
2. METODE PENELITIAN

2.1 Pengumpulan Data

Pengumpulan data berupa gambar digital RGB berformat jpeg dengan ukuran citra beragam dengan menggunakan metode observasi sebagai proses bagian dari analisis kebutuhan sistem serta melakukan studi literatur dari sumber-sumber tertulis seperti jurnal ilmiah dari penelitian terdahulu sebagai referensi dan acuan dalam pembuatan sistem yang akan dilakukan.

2.2 Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah menggunakan metode *waterfall*. Model *waterfall* adalah model yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean dan pengujian.



Gambar 2.1 Model *Waterfall* (Sukamto & Shalahuddin., 2018)

Berikut ini adalah tahap dari model waterfall (Kahfi *et al.*, 2023) yaitu :

a. Analisis

Melakukan analisis kebutuhan perangkat lunak, fungsi dan proses dari web yang dibuat, pengidentifikasian kendala dalam pembuatan web, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

b. Desain

Proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

c. Pengodean

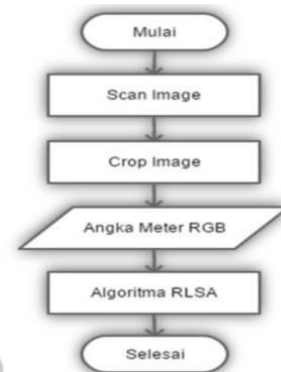
Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain. Atau tahapan penulis membuat program dengan bahasa program seperti php, html, scc dan lain-lain.

d. Pengujian

Dilakukan pengujian terhadap program yang telah dibuat untuk mengetahui kekurangan dari program tersebut. Seperti validasi halaman login, apakah sesuai dengan harapan.

2.3 Preprocessing

Peneliti membangun aplikasi pengolahan citra yang dapat mengenali dan menganalisa citra alat pencatat meter air dengan bantuan komputer. Adapun langkah-langkah penelitian yang digunakan untuk mendeteksi pengenalan citra digital dapat dilihat pada gambar dibawah ini.



Gambar 2.2 Tahap Preprocessing

(Putra *et al.*, 2021)

Tahap preprocessing, data akan diproses melalui langkah-langkah sebagai berikut:

1. Mulai

Tahap Mulai biasanya merupakan langkah awal yang menandai dimulainya proses preprocessing.

2. *Scan Image*

Scan Image adalah tahap di mana gambar dari objek (misalnya, meteran) diambil atau dipindai untuk mendapatkan representasi digital.

3. *Crop Image*

Crop Image adalah proses pemotongan bagian gambar yang tidak relevan untuk memfokuskan area yang berisi informasi penting, seperti angka pada meteran.

4. Angka Meter RGB

Angka Meter RGB merujuk pada tahap di mana gambar hasil crop diproses berdasarkan nilai warna RGB (*Red, Green, Blue*).

5. Algoritma RLSA (*Run Length Smearing Algorithm*)

RLSA (*Run Length Smearing Algorithm*) adalah teknik yang digunakan dalam

pengolahan gambar, khususnya dalam OCR (*Optical Character Recognition*), untuk menghubungkan komponen teks yang terputus-putus menjadi satu kesatuan.

3. HASIL DAN PEMBAHASAN

3.1 Preprocessing Data

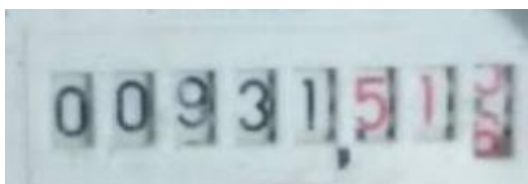
Sampel yang telah dikumpulkan selanjutnyadilakukan proses :

1. Crop Image

Crop Image merupakan tahap penting dalam melakukan preproses data dimana dilakukan pemotongan gambar untuk memperoleh bagian-bagian angka meter air saja yang akan digunakan untuk proses selanjutnya. Berikut pada gambar 3.1 merupakan sampel yang belum dilakukan pemotongan dan gambar 3.2 setelah sampel diproses pemotongan.



Gambar 3.1 Sampel sebelum di crop



Gambar 3.2 Sampel setelah di crop

2. Angka Meter RGB-Grayscale

Konversi gambar berwarna menjadi gambar *grayscale*, proses satu ini merupakan tahap yang tidak kalah penting karena dengan proses perubahan gambar menjadi *grayscale* akan mempermudah sistem algoritma mendeteksi angka-angka pada meter air tersebut dan juga bertujuan agar mengurangi kompleksitas data. Adapun hasil dari codingan *grayscale* dapat dilihat pada gambar 3.3.

```
python
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
```

Gambar 3.3 Codingan RGB-Grayscale

3. Penerapan Algoritma *Run-Length Smearing*

Algoritma ini diterapkan pada gambar biner untuk menghubungkan piksel-piksel yang berdekatan dan membentuk blok-blok yang lebih besar. Ini berguna dalam menyatukan bagian-bagian angka yang mungkin terpisah karena noise atau perbedaan kecil pada gambar.

```
python
processed_image = rlsa(binary_image, horizontal=True, vertical=True, value=rlsa_value)
```

Gambar 3.4 penerapan Algoritma *Run-Length Smearing*

3.2 Pengembangan Sistem

Setelah melakukan tahap coding, hasil yang dihasilkan adalah website deteksi meter air menggunakan algoritma *Run-Length Smearing* dengan metode *waterfall* dimana bentuk dan tampilannya dapat kita lihat di bahasan dibawah ini.

1. Analisis

Analisis Kebutuhan, yang bertujuan untuk mengidentifikasi dan mendokumentasikan persyaratan dan spesifikasi sistem yang akan dikembangkan. Ini adalah tahap kritis karena adanya masalah atau kelalaian di dalamnya, yang dapat menyebabkan masalah serius pada tahap-tahap berikutnya.

1). Rancangan yang akan dikembangkan

a. Use Case Diagram

Use case diagram adalah sebuah model untuk menjelaskan bagaimana aktor berinteraksi dengan sistem yang akan dikembangkan.

b. Aktivitas Diagram

Aktivitas Diagram adalah representasi skematis dari pergerakan atau aktivitas kerja suatu sistem atau proses bisnis. Berdasarkan gambar dari diagram use case

yang diusulkan, maka aktivitas diagram yang dirancang sebagai berikut:

- a. Admin masuk ke halaman website
- b. Sistem menampilkan halaman utama website
- c. Admin menginput gambar meteran air
- d. Admin mengsubmit gambar
- e. Sistem memvalidasi gambar alat pencatat meter air
- f. Sistem mendeteksi angka meter air
- g. Jika berhasil akan menampilkan prediksi angka meter air
- h. Jika gagal maka akan kembali ke input gambar
- i. Selesai

2. Desain

Pada tahap ini peneliti melakukan desain arsitektur, menentukan struktur keseluruhan sistem dan bagaimana komponen-komponennya berinteraksi. Dibawah ini adalah gambar rancangan halaman menu upload gambar dan menu halaman hasil prediksi.

1. Tampilan Menu Upload Gambar Water meter

Fitur Menu Upload Gambar Water Meter memungkinkan pengguna untuk mengunggah gambar meteran air mereka sendiri ke sistem. Fitur ini dimaksudkan untuk memudahkan proses pembacaan meteran air. Komponen utama dari menu ini adalah sebagai berikut:

1. Tombol Pilih File, Pengguna dapat menggunakan tombol ini untuk memilih gambar materan air dari perangkat mereka.
2. Tombol Unggah, Mengirim gambar yang dipilih ke server atau sistem untuk selanjutnya di deteksi oleh algoritma *Run-Length Smearing*.

///

Gambar 3.5 Menu Upload gambar wáter meter

2. Tampilan Menu Hasil Deteksi

Menu Hasil Deteksi Angka Meter Air adalah fitur dalam aplikasi atau sistem yang menampilkan hasil deteksi angka di udara setelah data gambar diolah menggunakan algoritma RLSA. Algoritma ini digunakan untuk secara otomatis menganalisis dan memprediksi sudut yang ada dalam pengukuran udara, meningkatkan akurasi dan efisiensi dalam pengukuran.

Komponen utama dari menu ini adalah sebagai berikut:

1. Hasil Deteksi dari gambar meteran air yang diunggah penggunaan, dengan angka yang ditandai atau disorot pada gambar ini yang dideteksi oleh algoritma *Run-Length Smearing* (RLSA).
2. Tingkat Deteksi Angka, Memberikan sudut tertentu yang diidentifikasi oleh algoritma RLSA biasanya disertai dengan penjelasan posisi sudut pada meteran.
3. Akurasi Deteksi memberikan informasi tentang keakuratan atau kepercayaan deteksi algoritma *Run-Length Smearing*.

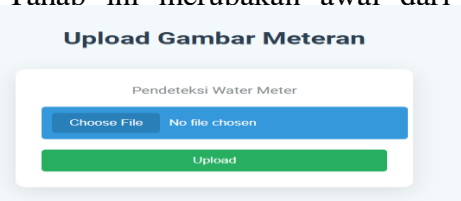


Gambar 3.6 menu hasil prediksi

3. Pengodean

Pada Tahap ini merupakan awal dari pro
Im
gan
diel
kon

ditentukan selama fase desain. Script



algoritma RLSA untuk mendeteksi angka meter air adalah implementasi dari algoritma RLSA, salah satu metode pembelajaran mesin (*machine learning*) yang sederhana nampak efektif untuk tugas klasifikasi biner dan, dengan beberapa modifikasi, untuk klasifikasi multi-kelas seperti dalam kasus deteksi angka pada meteran air. Di bawah ini adalah komponen utama dan sub-komponen script ini:

1. Inisiasi RLSA, Pembobotan awal membuat dan menerapkan bobot (bot) awal secara acak atau nol untuk setiap fitur dalam himpunan data
2. Bias, meningkatkan kata bias (nilai bias) yang akan digunakan secara optimal bersama dengan bot untuk mengurangi kemiringan data.
3. Data preprocessing penyesuaian ukuran gambar mengurangi ukuran gambar meteran udara ke standar yang lebih mudah dipahami oleh algoritma misalnya, 28x28 piksel untuk setiap angka.
4. Mengonversi ke Grayscale

```

def rlsa(img, horizontal=True, vertical=True, value=10):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img = cv2.resize(img, (28, 28))

    # Horizontal RLSA
    if horizontal:
        start = 0
        end = img.shape[0]
        while start < end:
            mid = (start + end) // 2
            binary_img = cv2.threshold(img[mid:], 125, 255, cv2.THRESH_BINARY)[1]
            if cv2.countNonZero(binary_img) > value:
                start = mid + 1
            else:
                end = mid

    # Vertical RLSA
    if vertical:
        start = 0
        end = img.shape[1]
        while start < end:
            mid = (start + end) // 2
            binary_img = cv2.threshold(img[:,mid:], 125, 255, cv2.THRESH_BINARY)[1]
            if cv2.countNonZero(binary_img) > value:
                start = mid + 1
            else:
                end = mid

    # Implementasi RLSA
    # Fungsi untuk membuat label dengan RLSA
    def create_labels_with_rlsa(image_dir, label_dir, rlsa_value=10):
        if not os.path.exists(label_dir):
            os.makedirs(label_dir)

        for filename in os.listdir(image_dir):
            # Nama gambar sebagai grayscale
            image_path = os.path.join(image_dir, filename)
            image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

            if image is None:
                print(f'Error: Gambar tidak ditemukan di {image_path}.')
                continue

            # Variasikan thresholding kasar jika diperlukan
            binary_image = cv2.threshold(image, 125, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)

            # Variasikan RLSA
            processed_image = rlsa(binary_image, horizontal=True, vertical=True, value=rlsa_value)

            # Simpan gambar hasil processing (optional)
            processed_image_path = os.path.join(image_dir, 'processed_' + filename)
            cv2.imwrite(processed_image_path, processed_image)
    
```

```

def create_labels_with_rlsa(image_dir, label_dir, rlsa_value=10):
    processed_image = rlsa(binary_image, horizontal=True, vertical=True, value=rlsa_value)

    # Simpan gambar hasil processing (optional)
    processed_image_path = os.path.join(image_dir, 'processed_' + filename)
    cv2.imwrite(processed_image_path, processed_image)

    # Ambil bagian "value" dari nama file untuk pebalikan
    filename_parts = filename.split('.')
    try:
        value_index = filename_parts.index('value') # Temukan indeks 'value'
        label_value = int(filename_parts[value_index + 1]) # Ambil nilai setelah 'value'
    except (ValueError, IndexError):
        label_value = 0 # Jika 'value' tidak ditemukan, atau tidak ada angka, beri nilai default

    # Buat nama file label
    label_filename = os.path.splitext(filename)[0] + '.txt'
    label_path = os.path.join(label_dir, label_filename)

    # Tulis label ke file
    with open(label_path, 'w') as f:
        f.write(f'{label_value}\n')
    print(f'Label file created: {label_path}')

# Siapkan gambar dan label
train_images_dir = 'data/train/train_images'
train_labels_dir = 'data/train/train_labels'
test_images_dir = 'data/test/test_images'
test_labels_dir = 'data/test/test_labels'

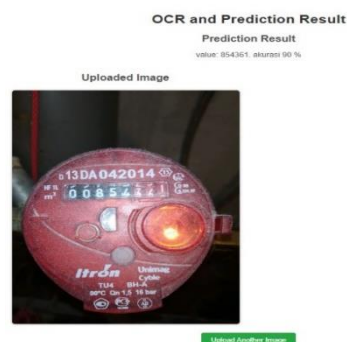
# Memulai label untuk data pelatihan dan pengujian
create_labels_with_rlsa(train_images_dir, train_labels_dir, rlsa_value=10)
create_labels_with_rlsa(test_images_dir, test_labels_dir, rlsa_value=10)
    
```

Gambar 3.7 Codingan Run-Length Smeraing Algorithm

4. Pengujian

Pada tahap ini dilakukan pengujian website, testing ini dilakukan secara berurutan sesuai dengan alur Waterfall, dan pengujian di setiap tahap harus diselesaikan dan divalidasi sebelum melanjutkan ke tahap berikutnya. Hasil uji akurasi ini menunjukkan bahwa algoritma perceptron dapat dimanfaatkan sebagai *Optical Character Recognition* untuk Membaca Angka Meter Air karena tingkat akurasi yang tinggi, berdasarkan data pengujian yang diperoleh sebanyak 10 data uji akurasi, dapat dilihat pada gambar dibawah ini.

1. Berdasarkan hasil pengujian pertama, teridentifikasi angka meter air dengan nomor 854361, menunjukkan tingkat akurasi pengujian 90%



Gambar 4.1 Prediksi Hasil Deteksi

2. Hasil pengujian kedua, merupakan nomor meter air yang teridentifikasi 112725 dengan ini menunjukkan bahwa akurasi yang diperoleh 90%



Gambar 4.2 Prediksi hasil deteksi

3. Pengujian ketiga, dengan hasil deteksi 141737 tingkat akurasinya 90%



Gambar 4.3 Prediksi hasil deteksi

4. Pengujian keempat, dengan nomor 43335 yang teridentifikasi, menunjukkan akurasi pengujian yang cukup tinggi yaitu 100%



Gambar 4.4 Prediksi hasil deteksi

5. Nomor 9999539 yang teridentifikasi pada pengujian kelima ini, Tingkat akurasi 90%



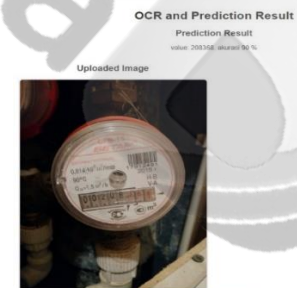
Gambar 4.5 Prediksi hasil deteksi

6. Selanjutnya hasil identifikasi dengan nomor 146443, akurasi yang diperoleh 90%



Gambar 4.6 Prediksi hasil deteksi

7. Dihasilkan Tingkat akurasi 90%, dengan nomor meter air yang terdeteksi 208368



Gambar 4.7 Prediksi hasil deteksi

8. Pengujian kedelapan memperoleh hasil identifikasi dengan nomor 1765869, Dimana akurasinya 90%



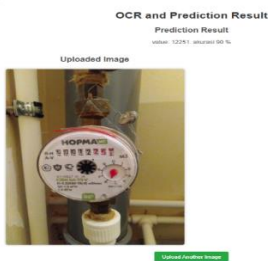
Gambar 4.8 Prediksi hasil deteksi

9. Hasil deteksi angka meter air dengan nomor 213785, dengan akurasi 90%



Gambar 4.9 Prediksi hasil deteksi

10. Pengujian terakhir memperoleh Tingkat akurasi 90%, dengan nomor 12251



Gambar 4.10 Prediksi hasil deteksi

Tabel 4.1 Pengujian BlackBox

Fitur yang Diuji	Deskripsi	Input	Ekspektasi Output	Status
Upload Gambar Meteran	Pengguna mengunggah gambar meteran air untuk dianalisis.	Gambar meteran dalam format yang didukung (JPEG, PNG)	Gambar berhasil diunggah dan ditampilkan pada antarmuka	Berhasil
Submit Gambar	Pengguna mengirim gambar yang telah diunggah untuk diproses oleh algoritma.	Klik tombol submit pada gambar yang diunggah	Gambar berhasil dikirim untuk pemrosesan	Berhasil
Deteksi Angka	Sistem memproses gambar menggunakan algoritma Run-Length Smearing untuk	Gambar meteran yang telah dikirim	Angka pada meteran terdeteksi dengan benar	Berhasil

	mendeteksi angka.		oleh sistem	
Hasil Prediksi Angka Meteran	Sistem menampilkan hasil prediksi angka dari gambar meteran yang telah diproses.	Gambar yang berhasil diproses	Angka yang terdeteksi ditampilkan dengan benar kepada pengguna	Berhasil

Berdasarkan Tabel 4.1 menunjukkan hasil pengujian blackbox terhadap fitur-fitur utama dalam sistem identifikasi angka meter air menggunakan algoritma *Run-Length Smearing*. Pengujian dilakukan pada empat fitur, yaitu Upload Gambar Meteran, Submit Gambar, Deteksi Angka, dan Hasil Prediksi Angka Meteran. Setiap fitur diuji dengan memberikan input yang sesuai, seperti gambar meteran air dalam format JPEG atau PNG, dan hasil yang diharapkan adalah sistem dapat mengunggah gambar, memprosesnya, mendeteksi angka pada meteran, serta menampilkan hasil prediksi angka dengan benar. Berdasarkan pengujian ini, semua fitur yang diuji berhasil berfungsi sesuai dengan ekspektasi, menunjukkan bahwa sistem mampu menjalankan tugas-tugas inti dengan baik.

4. KESIMPULAN

Implementasi Algoritma *Run-Length Smearing* untuk mendeteksi angka pada gambar alat pencatat meteran air menunjukkan bahwa meskipun sederhana, model ini mampu memberikan hasil yang akurat dan efisien dalam klasifikasi visual angka. Algoritma ini berhasil mendeteksi angka dengan tingkat akurasi yang tinggi dan kecepatan pelatihan serta prediksi yang cepat, menjadikannya pilihan yang efisien untuk sistem otomatisasi pembacaan angka meter air. Selain itu, kesederhanaan dan kemudahan implementasi *Run-Length Smearing* membuatnya ideal untuk digunakan dalam sistem dengan sumber daya komputasi terbatas. Namun, untuk mengatasi keterbatasannya dalam menangani dataset

yang kompleks atau non-linear, teknik lain seperti deep learning dapat dipertimbangkan. Model RLSA ini juga mudah diintegrasikan ke dalam sistem monitoring meteran air otomatis, yang dapat beroperasi secara real-time, memberikan manfaat dalam efisiensi operasional dan mengurangi kesalahan manual.

DAFTAR PUSTAKA

- KABUL, L. M. (2023). Rencana Bisnis Pdam Kabupaten Lombok Timur 2021-2026. *Ganec Swara*, 17(1), 222.
- Kahfi, A. H., Hasan, M., & Fazriansyah, A. (2023). Perancangan Program Pembayaran Administrasi Sekolah Berbasis Website Menggunakan Metode Waterfall. *Analisis Tingkat Kepuasan Pengguna Sistem Informasi Perpustakaan Menggunakan Pieces Framework*, 3(6), 1063–1069.
- Liliana, Budhi, G. S., & Hendra. (2010). *Segmentasi Plat Nomor Kendaraan Dengan Menggunakan Metode Run-Length Smearing Algorithm (RLSA)*.
- Maulani, J. (2020). Penerapan Metode Waterfall Pada Pengembangan Aplikasi Sistem Informasi Jasa Dan Penjualan Dengan Pemodelan Berorientasi Objek. *Technologia: Jurnal Ilmiah*, 11(2), 64.
- Ningsih, W., & Nurfauziah, H. (2023). Perbandingan Metode Waterfall Dan Metode Prototype Untuk Pengembangan Aplikasi Pada Sistem Informasi. *Jurnal Ilmiah Metadata*, 5(1), 83-95.
- Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. (2022). Rancangan Dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis Web. *Jurnal Teknik Dan Science*, 1(2), 88–103.
- Octaviani, T., Setiawan, H., & Kelana, O. H. (2023). Pytesseract dan Template Matching Untuk Otomatisasi Input Data KTP 147 dan Teknologi. *Kab. Malang*, 65151(1), 147–156.
- Pasciana, R., Sarbiti, & Saripin, S., P., L. (2019). Kepuasan Masyarakat Dalam Pelayanan di Perusahaan Daerah Air Minum (PDAM) Cabang Tarogong Kidul Kabupaten Garut. *Jurnal Pembangunan Dan Kebijakan Publik*, 10(1), 11–18.
- Putra, E. D., Utami, M., & Rifqo, M. H. (2021). Identifikasi Text Meteran Air Menggunakan Metode Run-Length Smearing Algorithm (Rlsa). *JUKOMIKA (Jurnal Ilmu Komputer Dan Informatika)*, 4(2), 98–106.
- Silalahi, F. J. Perancangan Sistem Informasi Penggajian Karyawan KSP Cahaya Ibu Sejahtera Bekasi.
- Sofwan, A. (2007). Belajar Mysql dengan Phpmyadmin. *Fakultas Teknologi Informasi, Universitas Budi Luhur*.
- Sukamto, R. A., & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak. Informatika*.
- Yuliano, T. (2007). Pengenalan Php. IlmuKomputer. com.