

# Computer Vision-Based Optical Mark Recognition (OMR) System for Automated Exam Answer Sheet Correction

Muhammad Yusuf<sup>1</sup>, Afu Ihsan Pradana<sup>2</sup>, Rudi Susanto<sup>3</sup>

<sup>1,2,3</sup> Fakultas Ilmu Komputer, Universitas Duta Bangsa, Surakarta, 57154, Indonesia

## Informasi Artikel

Diterima : 14 Juni 2025  
Revisi : 17 Juli 2025  
Publikasi : 30 September 2025

## Kata Kunci:

Computer Vision  
Koreksi Otomatis  
OMR  
OpenCV

## ABSTRAK

Koreksi manual lembar jawaban ujian konvensional merupakan masalah utama karena memakan waktu dan rentan terhadap kesalahan, sementara perangkat OMR komersial berbiaya mahal. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem *Optical Mark Recognition* (OMR) berbasis *computer vision* yang efisien dan akurat untuk lembar jawaban konvensional. Metode yang digunakan adalah *pipeline* pemrosesan citra menggunakan OpenCV, yang mencakup pra-pemrosesan dengan *Adaptive Thresholding*, deteksi kontur, dan transformasi perspektif. Evaluasi sistem dilakukan menggunakan metode pengujian kotak hitam (*Black Box Testing*) pada dataset uji yang terdiri dari 90 gambar (3 kondisi x 30 sampel). Hasil pengujian kinerja menunjukkan sistem mencapai akurasi 100% pada kondisi arsiran tebal, 99.22% pada arsiran tipis, dan 86.22% pada kondisi dengan banyak coretan. Disimpulkan bahwa sistem sangat efektif untuk lembar jawaban dengan arsiran bersih namun rentan terhadap *noise* berupa coretan. Kontribusi utama penelitian ini adalah menyajikan sebuah sistem OMR yang terjangkau dan dapat diimplementasikan pada lembar jawaban konvensional tanpa memerlukan formulir khusus, dengan mengandalkan metode *Adaptive Thresholding* yang terbukti tangguh terhadap variasi kontras.

## ABSTRACT

*Manual correction of conventional exam answer sheets is a primary problem due to being time-consuming and prone to human error, while commercial OMR devices are costly. This research aims to design and implement an efficient and accurate computer vision-based Optical Mark Recognition (OMR) system for conventional answer sheets. The method employed is an image processing pipeline using OpenCV, which includes preprocessing with Adaptive Thresholding, contour detection, and perspective transformation. System evaluation was conducted using a Black Box Testing method on a test dataset of 90 images (3 conditions x 30 samples). The performance test results show that the system achieved 100% accuracy on thick markings, 99.22% on thin markings, and 86.22% in conditions with numerous scribbles. It is concluded that the system is highly effective for answer sheets with clean markings but is vulnerable to visual noise from scribbles. The main contribution of this research is presenting an affordable OMR system that can be implemented on conventional answer sheets without requiring special forms, relying on the Adaptive Thresholding method which proved to be robust against contrast variations.*

This is an open-access article under the [CC BY-SA](#) license



## \*Penulis Koresponden

Email: [210103110@mhs.udb.ac.id](mailto:210103110@mhs.udb.ac.id)

Cara sitasi IEEE:

---

M. Yusuf, A. I. Pradana, dan R. Susanto, "Computer Vision-Based Optical Mark Recognition (OMR) System for Automated Exam Answer Sheet Correction," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 3, pp. 962-971, September 2025, doi: 10.30811/jaise.v5i3.8142

---

## 1. PENDAHULUAN

Evaluasi hasil pembelajaran melalui ujian merupakan komponen krusial dalam sistem pendidikan untuk mengukur pemahaman siswa [1]. Selama beberapa dekade, ujian konvensional berbasis kertas telah menjadi standar, namun proses koreksi manualnya sering kali menjadi kendala utama. Koreksi secara manual sangat memakan waktu dan rentan terhadap kesalahan manusia (*human error*), terutama ketika dihadapkan pada jumlah peserta ujian yang besar [2]. Inefisiensi ini dapat menghambat proses penilaian yang cepat dan akurat, yang merupakan kebutuhan mendasar bagi institusi pendidikan untuk memberikan umpan balik yang efektif kepada siswa dan pengajar.

Permasalahan tersebut mendorong adopsi Lembar Jawaban Komputer (LJK) yang dapat diproses oleh mesin *Optical Mark Recognition* (OMR). Namun, solusi ini menghadirkan tantangan baru: perangkat pemindai OMR komersial memiliki harga yang sangat mahal, sehingga penggunaannya terbatas bagi institusi dengan anggaran besar [3],[4]. Sebagai alternatif, banyak institusi, seperti MTS Nurul Iman, kembali melakukan koreksi manual yang lambat dan tidak efisien. Oleh karena itu, muncul kebutuhan mendesak untuk mengembangkan sebuah sistem koreksi lembar jawaban otomatis yang akurat, cepat, dan berbiaya rendah, sehingga dapat diakses oleh lebih banyak institusi pendidikan, termasuk SMA Batik 1 Surakarta yang masih mengandalkan ujian berbasis kertas konvensional.

Untuk menjawab tantangan tersebut, berbagai penelitian telah memanfaatkan teknologi *computer vision* sebagai alternatif yang lebih terjangkau. Beberapa pendekatan awal berfokus pada penggunaan deteksi tepi sebagai langkah utama. Penelitian oleh Pratama & Hanif mengimplementasikan metode deteksi tepi Canny untuk mengidentifikasi area jawaban dan mencapai akurasi 93% pada 15 dataset LJK. Serupa dengan itu, Hermawan, dkk. juga menggunakan kombinasi Canny, *thresholding*, dan *contour sorting* untuk mendeteksi LJK di MTS Nurul Iman dan melaporkan akurasi sebesar 97% pada 20 dataset dengan variasi arsiran. Sementara itu, penelitian yang lebih komprehensif oleh Zahari, dkk. menerapkan alur kerja (*pipeline*) yang lengkap, mencakup *grayscale*, *blur*, Canny, pencarian *biggest contour*, dan transformasi perspektif untuk menangani kemiringan serta variasi pencahayaan.

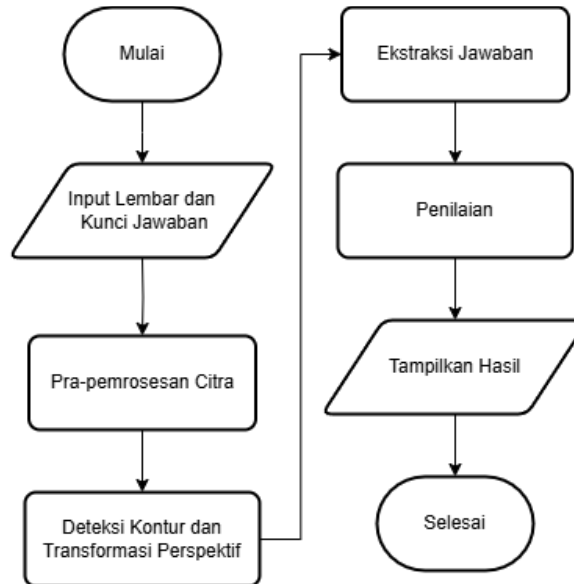
Dari tinjauan penelitian terdahulu, dapat dilihat bahwa pendekatan OMR berbasis *computer vision* telah banyak dieksplorasi. Namun, masih terdapat celah penelitian dalam evaluasi metode *Adaptive Thresholding* secara spesifik untuk kasus lembar jawaban konvensional non-standar, di mana variasi kualitas arsiran dan kondisi pencahayaan menjadi tantangan utama. Banyak penelitian sebelumnya berfokus pada LJK yang terstruktur rapi atau menggunakan metode yang kompleks secara komputasi seperti *deep learning*. Oleh karena itu, kontribusi utama dari penelitian ini adalah untuk mengisi celah tersebut dengan secara sistematis mengimplementasikan dan menguji kinerja metode *Adaptive Thresholding* pada lembar jawaban konvensional, dan menyajikan sebuah solusi OMR yang ringan secara komputasi (tanpa *deep learning*) sehingga lebih mudah diakses oleh institusi pendidikan dengan sumber daya terbatas.

Penelitian ini mengusulkan sebuah pendekatan sistem OMR berbasis *computer vision* yang dibangun menggunakan *library OpenCV* dengan bahasa pemrograman Python. Berbeda dengan penelitian sebelumnya yang banyak mengandalkan deteksi tepi Canny sebagai langkah krusial [5], penelitian ini mengimplementasikan *Adaptive Thresholding* sebagai metode utama untuk segmentasi area jawaban. Pendekatan ini dipilih karena potensinya yang lebih *robust* dalam menangani variasi pencahayaan dan kontras arsiran yang tidak seragam, sebuah tantangan umum pada lembar jawaban yang di-scan atau difoto menggunakan perangkat umum. Alur kerja sistem ini secara lengkap mencakup tahapan pra-pemrosesan (termasuk *resize*, *grayscale*, *blur*, dan *adaptive thresholding*), deteksi kontur lembar jawaban, transformasi perspektif untuk koreksi kemiringan, segmentasi area jawaban menjadi sel-sel individual, dan ekstraksi pilihan berdasarkan analisis kepadatan piksel.

Inovasi dan nilai baru dari penelitian ini terletak pada evaluasi sistematis terhadap efektivitas *Adaptive Thresholding* dalam konteks OMR pada lembar jawaban konvensional, serta pengujiannya terhadap berbagai kondisi kualitas arsiran yang realistis, yaitu arsiran tebal, arsiran tipis. Dengan demikian, penelitian ini bertujuan untuk menghasilkan sistem koreksi lembar jawaban otomatis yang tidak hanya akurat dan efisien, tetapi juga terbukti andal dalam menghadapi variasi input yang umum terjadi di lingkungan pendidikan nyata, sehingga dapat menjadi solusi yang praktis dan terjangkau.

## 2. METODE

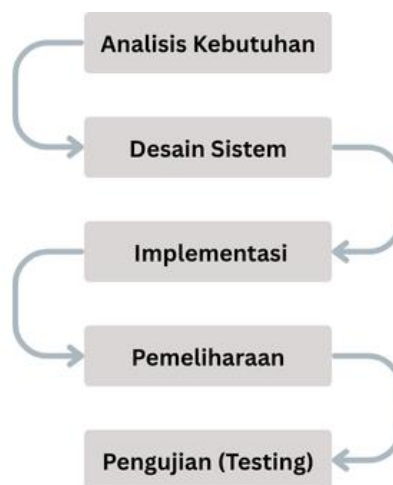
Penelitian Metode penelitian yang digunakan dalam pengembangan sistem OMR ini mengikuti sebuah alur kerja (pipeline) yang terstruktur [6]. Secara garis besar, prosesnya dimulai dari input gambar hingga menghasilkan skor akhir, yang mencakup tahapan pra-pemrosesan, deteksi dan transformasi, segmentasi, ekstraksi jawaban, serta penilaian. Alur kerja sistem secara lengkap diilustrasikan pada Gambar 1.



Gambar 1. Flowchart Sistem

Seperti yang diilustrasikan pada diagram alir di atas, proses sistem dimulai dengan input berupa gambar lembar jawaban dan kunci jawaban. Gambar tersebut kemudian melewati tahap pra-pemrosesan untuk meningkatkan kualitas citra. Selanjutnya, dilakukan deteksi dan transformasi untuk meluruskan gambar, diikuti dengan segmentasi dan ekstraksi untuk mengidentifikasi jawaban yang ditandai. Tahap terakhir adalah penilaian, di mana jawaban terdeteksi dibandingkan dengan kunci untuk menghasilkan skor akhir yang kemudian ditampilkan kepada pengguna.

Penelitian ini dilaksanakan dengan mengikuti serangkaian tahapan yang terstruktur untuk memastikan validitas dan reliabilitas hasil. Tahapan-tahapan tersebut meliputi desain penelitian, prosedur pengumpulan data, implementasi model proses algoritmik OMR, serta prosedur pengujian sistem.



Gambar 2. Metode Waterfall

Penelitian ini menggunakan pendekatan kuantitatif dengan metode pengembangan sistem model Waterfall. Model Waterfall dipilih karena sifatnya yang sekuensial dan sistematis, yang sangat cocok untuk proyek dengan ruang lingkup dan kebutuhan fungsional yang telah terdefinisi dengan jelas di awal [7]. Alur kerja penelitian secara keseluruhan, mulai dari pengumpulan data hingga penarikan kesimpulan, digambarkan pada Gambar 1. Desain ini memastikan bahwa setiap tahapan diselesaikan sebelum melanjutkan ke tahap berikutnya, sehingga proses pengembangan menjadi lebih terkontrol.

## 2.1. Pengumpulan Dataset

Sumber data dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer berupa Lembar Jawaban konvensional yang digunakan dalam ujian di SMA Batik 1 Surakarta. Data sekunder mencakup artikel jurnal dan prosiding konferensi yang relevan dengan topik OMR dan *computer Vision*.

## 2.2. Model Proses Algoritmik OMR

Inti dari sistem yang dikembangkan adalah sebuah model proses algoritmik atau *pipeline* pemrosesan citra yang dirancang menggunakan Python dan *library* OpenCV. Model ini secara deterministik mengubah gambar lembar jawaban mentah menjadi skor akhir melalui tahapan-tahapan berikut:

### 1. Pra-pemrosesan Citra

Tahap ini bertujuan untuk standarisasi dan peningkatan kualitas gambar input. Prosesnya meliputi perubahan ukuran gambar menjadi dimensi standar, konversi citra menjadi format *grayscale*, dan aplikasi filter *Gaussian blur* untuk reduksi *noise*. Penghalusan dengan *Gaussian blur* bekerja dengan melakukan konvolusi antara citra dengan sebuah *kernel* yang nilainya dibentuk dari fungsi Gaussian dua dimensi (2D) [8]. Langkah ini membantu dalam mencegah deteksi tepi palsu atau kontur yang tidak relevan pada tahap berikutnya. Fungsi Gaussian 2D direpresentasikan oleh rumus berikut:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

Dimana:

- $G(x,y)$  adalah nilai fungsi Gaussian pada posisi  $(x,y)$  dari pusat *kernel*.
- $x$  adalah jarak dari titik pusat pada sumbu horizontal.
- $y$  adalah jarak dari titik pusat pada sumbu vertical
- $\sigma$  (sigma) adalah standar deviasi dari distribusi Gaussian, yang mengontrol tingkat kehalusan atau *blur* pada gambar.

Langkah krusial berikutnya adalah binerisasi citra menggunakan teknik *Adaptive Thresholding*. [9] Teknik ini dipilih karena kemampuannya yang efektif dalam menangani variasi pencahayaan yang tidak seragam pada permukaan lembar jawaban, berbeda dengan *thresholding global*. *Adaptive thresholding* menentukan nilai ambang batas  $T(x,y)$  untuk setiap piksel secara individual berdasarkan analisis pada area lokal di sekitarnya. Aturan binerisasi untuk setiap piksel  $(x,y)$  adalah sebagai berikut:

$$dst(x, y) = \begin{cases} \maxval & \text{jika } src(x, y) > T(x, y) \\ 0 & \text{maka sebaliknya} \end{cases} \quad (2)$$

Dimana:

- $dst(x,y)$  adalah nilai piksel output (hasil binerisasi).
- $src(x,y)$  adalah nilai piksel input pada citra *grayscale*.
- $\maxVal$  adalah nilai maksimum untuk piksel (biasanya 255, untuk warna putih).
- $T(x,y)$  adalah nilai ambang batas yang dihitung untuk piksel tersebut.

Dalam penelitian ini, metode yang digunakan adalah *Gaussian Adaptive Thresholding*, di mana nilai  $T(x,y)$  adalah rerata terbobot Gaussian dari piksel-piksel tetangga dalam sebuah blok berukuran tertentu, kemudian dikurangi dengan sebuah konstanta  $C$  [10]. Ini memastikan bahwa ambang batas selalu relevan dengan kondisi pencahayaan lokal di sekitar setiap piksel.

### 2. Deteksi Lembar Jawaban dan Transformasi Perspektif

Setelah pra-pemrosesan, sistem melakukan deteksi kontur pada citra biner untuk mengidentifikasi objek-objek di dalamnya. Kontur terbesar yang memiliki bentuk persegi panjang diidentifikasi sebagai area lembar jawaban. Titik-titik sudut dari kontur ini kemudian digunakan untuk melakukan transformasi perspektif (*warping*) [11]. Proses ini meluruskan gambar lembar jawaban yang mungkin miring, menghasilkan gambar dengan orientasi tegak lurus yang penting untuk akurasi segmentasi selanjutnya.

3. Segmentasi dan Ekstraksi Jawaban

Gambar lembar jawaban yang telah diluruskan kemudian diisolasi pada bagian area jawaban. Area ini selanjutnya disegmentasi atau dibagi menjadi sel-sel individual yang merepresentasikan setiap opsi jawaban. Untuk setiap sel, sistem menghitung jumlah piksel putih (yang menandakan area terarsir setelah *thresholding inverse*) [12]. Opsi jawaban dengan jumlah piksel putih tertinggi dalam satu baris soal dianggap sebagai jawaban yang dipilih.

4. Penilaian dan Perhitungan Skor

Setelah semua jawaban yang ditandai berhasil diekstraksi menjadi sebuah daftar, sistem memasuki tahap penilaian atau *grading* [13]. Proses ini membandingkan daftar jawaban yang terdeteksi oleh sistem dengan daftar kunci jawaban ujian yang telah ditentukan sebelumnya. Jika jawaban yang terdeteksi untuk suatu nomor soal sama dengan kunci jawabannya, maka jawaban tersebut dianggap benar dan diberi nilai 1. Sebaliknya, jika tidak sama, jawaban dianggap salah dan diberi nilai 0. Total skor akhir kemudian dihitung menggunakan rumus persentase standar, di mana jumlah jawaban yang benar dibagi dengan jumlah total soal, lalu dikalikan 100 [14]. Penggunaan rumus ini merupakan praktik umum dalam penilaian dan juga diimplementasikan dalam penelitian OMR serupa.

$$\text{nilai} = \frac{\text{jumlah jawaban benar}}{\text{jumlah total soal}} \times 100\% \quad (3)$$

### 2.3. Prosedur Pengujian

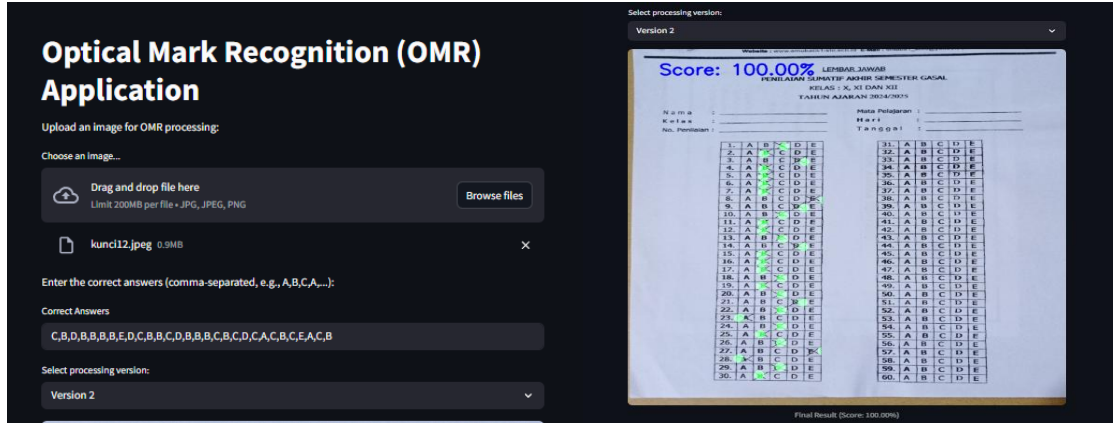
Pengujian sistem dilakukan secara otomatis menggunakan skrip *batch processing* untuk memproses seluruh gambar dalam dataset uji dan menyimpan hasilnya secara terstruktur. Kinerja sistem dievaluasi berdasarkan metrik utama yaitu Rata-rata Akurasi Sistem (%) [15]. Metrik ini mengukur persentase jawaban yang berhasil dideteksi dengan benar oleh sistem dibandingkan dengan jawaban yang secara fisik ditandai pada lembar jawaban (*ground truth*). Perhitungan akurasi ini sejalan dengan metode evaluasi yang digunakan pada penelitian-penelitian OMR sebelumnya.

## 3. HASIL DAN PEMBAHASAN

Pada bagian ini, disajikan hasil dari implementasi dan pengujian sistem *Optical Mark Recognition* (OMR) yang dikembangkan. Penyajian hasil mencakup output visual dari setiap tahapan pemrosesan citra dan data kuantitatif dari uji akurasi sistem. Pembahasan akan menganalisis temuan yang diperoleh dan mengaitkannya dengan penelitian-penelitian terkait.

### 3.1. Hasil Implementasi Sistem

Sistem OMR berhasil diimplementasikan dengan alur kerja pemrosesan citra yang telah dirancang. Antarmuka pengguna yang dibangun menggunakan Streamlit, seperti yang ditunjukkan pada Gambar 2, memudahkan pengguna dalam mengunggah gambar lembar jawaban dan melihat hasil koreksi. Alur kerja sistem secara visual menghasilkan output spesifik pada setiap tahapan penting.

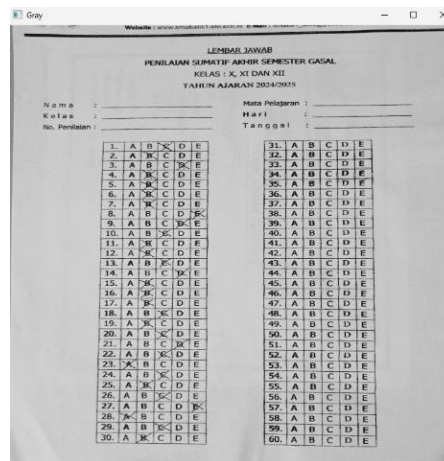


Gambar 3. Antarmuka Sistem

### 3.2. Tahap Pra-Pemrosesan

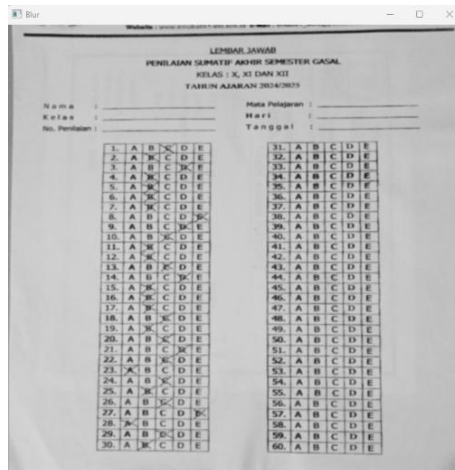
Tahap pra-pemrosesan adalah langkah fundamental yang bertujuan untuk meningkatkan kualitas citra lembar jawaban sehingga proses deteksi dan analisis selanjutnya dapat berjalan lebih akurat dan efisien [16]. Tahapan ini krusial untuk mengatasi variasi pada gambar input, seperti perbedaan pencahayaan atau adanya noise.

Perubahan Ukuran dan Konversi (*Grayscale*): Citra lembar jawaban yang diinput oleh pengguna pertama-tama diubah ukurannya menjadi dimensi standar (misalnya, 700x700 piksel) untuk memastikan konsistensi dalam pemrosesan selanjutnya. Setelah itu, citra berwarna dikonversi menjadi citra *grayscale* (keabuan). Konversi ini menyederhanakan informasi citra dari tiga kanal warna (RGB) menjadi satu kanal intensitas, yang efektif mengurangi kompleksitas komputasi tanpa menghilangkan fitur penting yang relevan untuk deteksi arsiran [17].



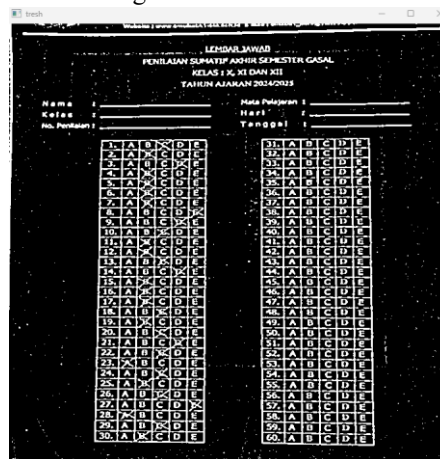
Gambar 4. Hasil Grayscale

*Gaussian Blur*: Untuk mengurangi *noise* minor pada citra *grayscale* dan menghaluskan gambar, filter *Gaussian blur* diaplikasikan. Langkah ini membantu dalam mencegah deteksi tepi palsu atau kontur yang tidak relevan pada tahap berikutnya, sebuah praktik yang juga umum dalam pemrosesan citra lembar jawaban.



Gambar 5. Hasil *Gaussian Blur*

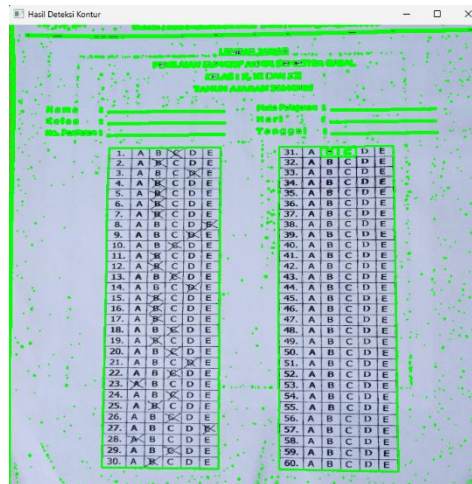
*Adaptive Thresholding*: Tahap selanjutnya adalah binerisasi citra menggunakan teknik *adaptive thresholding* dengan metode *Gaussian*. Teknik ini dipilih karena kemampuannya untuk menentukan nilai ambang batas secara lokal, sehingga sangat efektif dalam menangani variasi pencahayaan yang tidak merata pada permukaan lembar jawaban, berbeda dengan *thresholding* global yang mungkin kurang optimal [18]. Hasilnya adalah citra biner (hitam-putih) yang dengan jelas memisahkan area arsiran (menjadi piksel putih dengan *thresholding inverse*) dari latar belakang kertas.



Gambar 6. Hasil *Adaptive Thresholding*

### 3.3. Deteksi Lembar Jawab dan Transformasi Perspektif

Setelah citra dipra-pemrosesan, sistem melanjutkan dengan mengidentifikasi area lembar jawaban, meluruskannya, dan membagi area jawaban menjadi komponen-komponen individual untuk dianalisis. Deteksi Kontur Lembar Jawaban: Pada citra biner hasil *thresholding*, sistem melakukan pencarian kontur untuk mengidentifikasi objek-objek dalam gambar. Kontur terbesar yang memiliki ciri-ciri bentuk persegi panjang (empat sisi) diidentifikasi sebagai kandidat utama area lembar jawaban.



Gambar 7. Hasil Deteksi Kontur

Transformasi Perspektif: Setelah kontur lembar jawaban utama teridentifikasi, titik-titik sudutnya diekstraksi dan diurutkan. Titik-titik ini kemudian digunakan untuk melakukan transformasi perspektif. Tujuannya adalah untuk "meluruskan" gambar lembar jawaban yang mungkin diambil dari sudut miring, sehingga tampilannya seolah-olah dilihat tegak lurus dari atas [19]. Proses ini menghasilkan gambar lembar jawaban yang terstandarisasi ukurannya (misalnya, 700x700 piksel) dan orientasinya, yang sangat krusial untuk akurasi segmentasi area jawaban selanjutnya.



Gambar 8. Hasil Transformasi Perspektif

**3.4. Segmentasi dan Ekstraksi Jawaban**

Segmentasi Area Pilihan Jawaban dan Ekstraksi Jawaban: Dari gambar lembar jawaban yang telah diluruskan dan di-threshold ulang (jika diperlukan), area spesifik yang memuat barisan pilihan jawaban diisolasi terlebih dahulu. Area ini kemudian disegmentasi atau dibagi lagi menjadi kotak-kotak (sel) individual yang masing-masing merepresentasikan satu opsi jawaban (misalnya, A, B, C, D, E) untuk setiap nomor soal. Setelah segmentasi, untuk setiap kotak jawaban, sistem menghitung jumlah piksel putih (yang menandakan area terarsir). Opsi jawaban dengan jumlah piksel putih terbanyak dalam satu baris soal dianggap sebagai jawaban yang dipilih oleh pengguna.



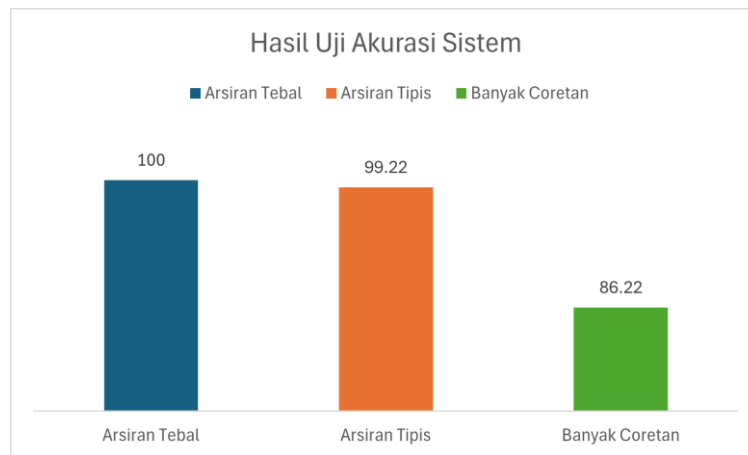
Gambar 9. Hasil Segmentasi

### 3.5. Hasil Uji Akurasi dan Analisis Kinerja

Pengujian kuantitatif dilakukan untuk mengevaluasi akurasi sistem pada tiga kondisi simulasi arsiran. Setiap kondisi diuji menggunakan 30 sampel lembar jawaban yang dihasilkan melalui *data augmentation*. Hasil pengujian disajikan pada Tabel 1.

Tabel 1. Hasil Uji Akurasi Sistem OMR pada Berbagai Kondisi

No	Deskripsi Kondisi	Jumlah Uji	Rata-rata Akurasi (%)
1	Arsiran Tebal	30	100
2	Arsiran Tipis	30	99.22
3	Arsiran Tipis, Banyak Coretan	30	86.22



Gambar 10. Grafik Perbandingan Akurasi Sistem

Berdasarkan Tabel 1 dan Gambar 10, pada Kondisi 1 (Arsiran Tebal), sistem mencapai akurasi sempurna 100.00%, menunjukkan bahwa alur kerja algoritma berfungsi dengan sangat baik pada kondisi input yang ideal. Pada Kondisi 2 (Arsiran Tipis), akurasi sistem tetap sangat tinggi sebesar 99.22%, membuktikan bahwa teknik *adaptive thresholding* cukup robust terhadap variasi kontras. Penurunan performa paling signifikan terjadi pada Kondisi 3 (Banyak Coretan), di mana akurasi turun menjadi 86.22%, mengindikasikan bahwa coretan menjadi faktor pengganggu utama bagi sistem saat ini.

Penurunan akurasi yang signifikan pada kondisi dengan coretan (86.22%) memerlukan analisis yang lebih mendalam. Kegagalan ini terjadi karena fundamental dari logika ekstraksi jawaban sistem saat ini, yang bekerja dengan menghitung jumlah piksel putih (*cv2.countNonZero*) pada setiap sel opsi dan menentukan pilihan berdasarkan kepadatan piksel tertinggi (*np.argmax*). Coretan acak, meskipun tidak disengaja, menambahkan "massa" piksel yang signifikan pada sel jawaban yang salah. Jika massa piksel dari coretan ini secara kebetulan lebih besar daripada massa piksel dari tanda silang (X) yang valid di sel jawaban yang benar, maka sistem akan keliru mengidentifikasi coretan tersebut sebagai jawaban yang dipilih. Hal ini menunjukkan tantangan teknis utama sistem, yaitu membedakan antara tanda valid yang dibuat pengguna dengan noise visual yang tidak terstruktur seperti coretan.

### 3.6. Pembahasan Lanjutan

Performa sistem OMR yang dikembangkan, terutama pada lembar jawaban dengan arsiran yang relatif bersih (99.22% - 100%), menunjukkan hasil yang sangat kompetitif jika dibandingkan dengan penelitian sebelumnya. Hasil ini melampaui akurasi yang dilaporkan oleh Pratama & Hanif (93% dengan metode Canny) dan sebanding dengan Hermawan, dkk [4]. (97% dengan kombinasi metode). Keunggulan ini dapat diatribusikan pada pemilihan *Adaptive Thresholding* yang lebih efektif dalam menangani variasi kontras dan pencahayaan dibandingkan metode deteksi tepi.

Penurunan akurasi pada kondisi lembar jawaban dengan coretan (86.22%) mengindikasikan keterbatasan utama sistem saat ini. Coretan dapat menginterferensi proses analisis piksel, sebuah tantangan yang tidak secara ekstensif dibahas dalam banyak literatur OMR konvensional [10]. Kontribusi penelitian ini adalah pada pengembangan sistem OMR yang efisien untuk lembar jawaban konvensional dengan penekanan pada *adaptive thresholding*, serta identifikasi dan kuantifikasi penurunan kinerja akibat adanya coretan.

Keterbatasan penelitian mencakup pengujian pada satu format lembar jawaban dan jenis coretan yang mungkin belum sepenuhnya mewakili semua skenario nyata [20]. Untuk pengembangan selanjutnya, peningkatan ketahanan terhadap coretan menjadi prioritas, misalnya melalui teknik segmentasi yang lebih canggih atau filter untuk eliminasi *noise* non-arsiran. Pengujian dengan format lembar jawaban berbeda serta analisis pengaruh variasi pencahayaan dan kemiringan yang lebih terkontrol juga akan sangat bermanfaat.

#### 4. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem *Optical Mark Recognition* (OMR) berbasis *computer vision* sebagai solusi koreksi otomatis yang akurat dan efisien untuk Lembar Jawaban konvensional. Berdasarkan hasil implementasi dan pengujian yang telah dipaparkan pada bagian sebelumnya, dapat disimpulkan bahwa tujuan tersebut telah berhasil dicapai. Sistem yang dikembangkan dengan *pipeline* pemrosesan citra yang menekankan pada penggunaan *Adaptive Thresholding* terbukti mampu mendeteksi jawaban dengan tingkat akurasi yang sangat tinggi pada kondisi yang umum ditemui. Sistem mencapai akurasi sempurna 100.00% pada kondisi arsiran tebal dan mempertahankan akurasi 99.22% pada kondisi arsiran tipis.

Meskipun demikian, penelitian ini juga mengidentifikasi batasan utama sistem, yaitu penurunan performa pada lembar jawaban yang memiliki banyak coretan, di mana akurasi turun menjadi 86.22%. Hal ini menunjukkan bahwa sistem saat ini sensitif terhadap gangguan visual (*noise*) yang signifikan pada area jawaban. Oleh karena itu, untuk pengembangan di masa depan, beberapa prospek penelitian lanjutan dapat dieksplorasi. Disarankan untuk mengimplementasikan algoritma yang lebih tahan dalam menangani *noise*, seperti menggunakan analisis bentuk (*shape analysis*) untuk memvalidasi arsiran atau menerapkan teknik pemfilteran morfologis untuk menghilangkan coretan kecil sebelum tahap ekstraksi jawaban. Selain itu, pengujian sistem pada berbagai format lembar jawaban dan di bawah kondisi lingkungan yang lebih beragam, seperti variasi pencahayaan dan kemiringan yang terkontrol, akan sangat bermanfaat untuk meningkatkan generalisasi dan ketahanan sistem.

#### REFERENSI

- [1] D. B. Gleneagles, F. Larasyifa, and R. Fawaiz, "Peran Teknologi Kecerdasan Buatan (AI) dalam Meningkatkan Efisiensi Proses Belajar dan Pembelajaran," *J. Ilm. Multidisiplin*, vol. 2, no. 5, pp. 107–116, 2024, [Online]. Available: <https://doi.org/10.5281/zenodo.11364580>
- [2] R. Al Fijrin and H. Ekawati, "Rancang Bangun Sistem Koreksi Lembar Jawaban Komputer Berbasis Web Menggunakan Metode Deteksi Tepi Canny Using the Canny Edge Detection Method," *J. Sebatik*, vol. 28, no. 2, pp. 1–6, 2024, doi: 10.46984/sebatik.v28i2.0000.
- [3] I. Zahari, Z. Pratama, W. Mahmud, and A. Wibowo, "Sistem Pengecekan Lembar Jawaban Komputer Dengan Optical Mark Recognition (OMR) Berbasis Open Computer Vision Python," *Senovtek*, pp. 9–16, 2022, [Online]. Available: <https://ejournal.pnc.ac.id/index.php/senovtek>
- [4] R. Pratama and F. Hanif, "Implementasi Metode Canny dalam Deteksi Tepi pada Aplikasi OMR (Optical Mark Recognition) Menggunakan Pengembangan Sistem Waterfall," *AT-TAWASSUTH J. Ekon. Islam*, vol. VIII, no. I, pp. 1–19, 2023.
- [5] M. U. Habibah and M. Kurniawan, "Segmentasi Citra Wajah dengan Implementasi Adaptif Threshold- Integral Image," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 5, pp. 919–928, 2021, doi: 10.25126/jtiik.2021853840.
- [6] S. M. D. Lainez, L. J. Q. Gresos, and V. K. A. Maganggo, "An Automated Entrance Examination Checker Using Optical Mark Recognition," *J. Comput. Software, Progr.*, vol. 1, no. 1, pp. 8–13, 2024, doi: 10.69739/jcsp.v1i1.43.
- [7] W. Harjono and K. J. Tute, "Perancangan Sistem Informasi Perpustakaan Berbasis Web Menggunakan Metode Waterfall," *SATESI J. Sains Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 47–51, 2022, doi: 10.54259/satesi.v2i1.773.
- [8] N. S. Hapsari, Y. Fatman, and B. Sugiarto, "Image Extraction in OpenCV Using the Local Binary Pattern Method," *J. Softw. Eng., Inf. Commun. Technol. (SEICT)*, vol. 5, no. December, pp. 75–86, 2024, doi: 10.17509/seict.v5i2.76251.
- [9] Q. Hafeez, W. Aslam, R. Aziz, and G. Aldehim, "An Enhanced Fault Tolerance Algorithm for Optical Mark Recognition Using Smartphone Cameras," *IEEE Access*, vol. 12, no. September, pp. 121305–121319, 2024, doi: 10.1109/ACCESS.2024.3451972.
- [10] H. Hadiq, S. Solehatin, D. Djuniharto, M. A. Muslim, and S. N. Salahudin, "Comparison of the Suitability of The Otsu Method Thresholding and Multilevel Thresholding for Flower Image Segmentation," *J. Soft Comput. Explor.*, vol. 4, no. 4, pp. 242–249, 2023, doi: 10.52465/joscex.v4i4.266.
- [11] B. Putra, I. K. D. Nuryana, and R. A. J. Firdaus, "Rancang Bangun Aplikasi Koreksi Lembar Jawaban Komputer Menggunakan Metode Deteksi Tepi Canny," *Inov. J. Ilm. Inov. Teknol. Inf. J. Ilm. Inov. Teknol. Inf.*, vol. 04, pp. 16–24, 2019, doi: 10.33752/inovate.v4i1.768.
- [12] G. Sugiarta, D. P. Andini, and S. Hidayatullah, "Ekstraksi Informasi/Data e-KTP Menggunakan Optical Character Recognition Convolutional Neural Network," *JTERA (Jurnal Teknol. Rekayasa)*, vol. 6, no. 1, p. 1, 2021, doi: 10.31544/jtera.v6.i1.2021.1-6.
- [13] M. Mirda, M. Sarjan, and U. Khairat, "Aplikasi Ujian Essay Koreksi Otomatis Menggunakan Metode Cosine Similarity," *J. Pegguruang Conf. Ser.*, vol. 4, no. 1, p. 418, 2022, doi: 10.35329/jp.v4i1.2344.
- [14] R. N. Rosa Puno *et al.*, "E-Chequer: a Smart Exam-Checking Machine Using Image Processing Technique," *Asian J. Multidiscip. Stud.*, vol. 6, no. 1, pp. 77–87, 2023.
- [15] M. F. Gunardi, "Implementasi Augmentasi Citra pada Suatu Dataset," *J. Inform.*, vol. 9, no. 1, pp. 1–5, 2023.
- [16] M. I. Jauhari and R. Wulanningrum, "Sistem Deteksi Kendaraan Menggunakan StreamLit Metode Yolo," *INOTEK*, vol. 8, pp.

- 1331–1336, 2024.
- [17] J. Masula, *et al.*, “Development of an Offline Android-based Test Paper Checker Application for Northwestern Visayan Colleges,” *J. Innov. Technol. Conver.*, vol. 6, no. 3, pp. 31–40, 2024, doi: 10.69478/jitc2024v6n3a04.
- [18] A. Maulana, F. Auliatunnajah, N. Rosidin, M. Ramadien Rizki Darmawan, and P. Rosyani, “Implementasi OpenCV dengan Metode Image Thresholding pada Gambar,” *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 2, no. 1, pp. 27–32, 2024, [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [19] D. Aaisyah Salmaa Putri Atmaja and U. Kalimantan Timur, “Implementation of Grayscale Image Transformation and Histogram Equalization Methods in Digital Image Processing,” *Sist. Kendali Jaringan) E-ISSN*, vol. 4, no. 2, pp. 2808–3520, 2025, doi: 10.58982/krisnadana.v4i2.739.
- [20] S. Susandri, S. Sumijan, A. Zamsuri, R. Rahmiati, and A. Asparizal, “Enhancing Dental Image Segmentation Techniques: Edge Detection and Color Thresholding,” *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 15, no. 1, pp. 94–104, May 2024, doi: 10.31849/digitalzone.v15i1.18757.