

**IMPLEMENTASI METODE CANNY DALAM DETEKSI TEPI
PADA APLIKASI OMR (OPTICAL MARK RECOGNITION)
MENGUNAKAN PENGEMBANGAN SISTEM WATERFALL**

Muhamad Rizki Pratama¹, Isa Faqihuddin Hanif²

Teknik Informatika, Universitas Muhammadiyah Prof Dr Hamka, Jakarta, Indonesia

E-mail: Rizkiprtma03@gmail.com¹, isa@uhamka.ac.id²

INFO ARTIKEL

Diterima: 9
Februari 2022

Direvisi: 12
Februari 2022

Disetujui: 20
Februari 2022

ABSTRAK

Pengolahan citra digital merupakan suatu teknologi input berupa gambar seperti foto atau video yang dapat dimanfaatkan dalam meningkatkan kualitas suatu citra, sedangkan output dari pengolahan citra visual dapat berupa gambar atau sejumlah karakteristik yang berkaitan dengan sebuah gambar. Serta proses dalam identifikasi citra tersebut dibantu dengan menggunakan beberapa metode dengan maksud mengembangkan sistem yang bersumber pada citra tersebut. Dalam hal ini, penerapan teknik pengolahan citra digital akan digunakan pada objek lembar jawaban untuk mempercepat dan mempermudah proses mengoreksi hasil ujian dengan konsep visi komputer (Computer Vision). Pengembangan teknologi dengan standar Computer Vision ini, sebagaimana diterapkan dalam proses mengoreksi hasil ujian layaknya Lembar Jawaban Komputer (LJK) yang menggunakan alat mesin pemindai OMR. Pada penelitian ini, sebelum melakukan implementasi terhadap citra lembar jawaban dan teknologi OMR maka dilakukan tahapan preprocessing, segmentasi threshold hingga tahapan melakukan deteksi jawaban pada lembar jawaban. Aplikasi ini diimplementasikan dengan memanfaatkan library opencv serta metode Canny dengan kombinasi alur pengembangan sistem Waterfall. Penelitian ini menggunakan 15 dataset citra lembar jawaban. Dimana masing masing komposisi citra yaitu 8 hasil foto lembar jawaban sebagai citra arsiran penuh, 5 hasil foto lembar jawaban sebagai citra arsiran separuh dan 2 hasil foto lembar jawaban sebagai citra arsiran salah. Adapun hasil akurasi yang diberikan pada uji coba semua citra yang dilakukan ialah 93%.

Kata kunci: Citra; Lembar Jawaban; OMR; Sistem Waterfall

ABSTRACT

Digital image processing is an input technology in the form of images such as photos or videos that can be used to improve the quality of an image, while the output of visual image processing can be in the form of images or a number of characteristics related to an image. As well as the process of identifying the image is assisted by using several methods with the aim of developing a system based on the image. In this case, the application of digital image processing techniques will be used on answer sheet objects to speed up and simplify the process of correcting exam results with the concept of computer vision (Computer Vision). Technology development with the Computer Vision standard, as implemented in the process of correcting exam results like a Computer Answer Sheet (LJK) that uses an OMR scanner. In this study, before implementing the answer sheet images and OMR technology, the preprocessing stage, threshold segmentation, to the stage of detecting answers on the answer sheet were carried out. This application is implemented by utilizing the opencv library and the Canny method with a combination of the Waterfall system development flow. This research uses 15 answer sheet image datasets. Where each image composition is 8 answer sheet photos as full shaded images, 5 answer sheet photos as half shaded images and 2 answer sheet photos as wrong shaded images. The results of the accuracy given in the trials of all images carried out were 93%.

Keywords: Image; Answer Sheet; OMR; Waterfall System



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International

PENDAHULUAN

Banyak teknologi aplikasi pemrosesan gambar dan visi komputer (*Computer Vision*) telah secara signifikan menyebarkan penggunaan deteksi tepi sebagai alat utama untuk ekstraksi sebuah informasi Click or tap here to enter text. (Chaudhary, Patel, Kumar, & Tomar, 2017). Salah satu contoh teknologi digital dalam pemrosesan gambar ialah pengolahan citra digital. Pengolahan citra merupakan bagian penting dalam mendukung berbagai aplikasi seperti pengenalan pola (Riadi, Yudhana, & Sulisty, 2020).

Adapun dalam pengolahan citra digital perlu adanya sebuah data yang di input agar dapat diproses dan menghasilkan sebuah output citra digital. Dalam pemrosesan pengolahan citra digital mencakup perbaikan citra, pemampatan citra, segmentasi citra dan deteksi tepi citra (Fawwaz & Dharshinni, 2021). Pendeteksi tepi merupakan objek-objek gambar yang diproses dengan menghasilkan tepi-tepi citra dan dapat mempresentasikan bentuk pada objek tersebut (Kuswandi & Fadillah, 2019). Objek dalam pendeteksian ini menggunakan objek lembar jawaban yang sering ditemukan sebagai media tes ujian yang dilakukan dalam bentuk alat tulis seperti kertas, pensil dan pena. Ujian merupakan salah satu cara untuk menilai keberhasilan proses pengajaran di sekolah (Hermawan, 2021). Oleh karena itu ujian

dirancang untuk mengukur potensi siswa terhadap tingkat pemahaman sehingga siswa dapat mengetahui kemampuannya di bidang pelajaran yang sedang ditempuh. Ujian dapat digunakan juga untuk mengukur dan mengevaluasi keberhasilan materi pelajaran yang telah diajarkan oleh guru kepada siswa (Sampebua & Membala, 2018).

Dalam hal ini, penerapan teknik pengolahan citra digital akan digunakan pada objek lembar jawaban untuk mempercepat dan mempermudah proses mengoreksi hasil ujian dengan konsep visi komputer (*Computer Vision*). *Computer vision* adalah proses pengolahan gambar dan video untuk mendapatkan hasil analitik (Kaloh, Poekoel, & Putro, 2018). Dengan kata lain, *Computer Vision* memiliki tujuan untuk membangun sebuah mesin cerdas yang dapat “melihat” (Gautama, Hendrik, & Hendaya, 2016) artinya, *computer vision* memiliki fungsi yang sama dengan mata manusia yaitu *vision* atau penglihatan (Kaloh et al., 2018).

Pengembangan teknologi dengan standar *Computer Vision* ini, sebagaimana diterapkan dalam proses mengoreksi hasil ujian layaknya Lembar Jawaban Komputer (LJK) yang menggunakan alat mesin pemindai OMR (Hermawan, 2021). Namun, dalam mengoreksi lembar jawaban komputer perlu mempunyai alat tertentu seperti scanner yang nantinya akan digunakan untuk proses mendeteksi pilihan pada lembar jawaban komputer tetapi alat mesin tersebut memiliki harga yang cukup mahal. Salah satu alternatif lain adalah mengoreksi lembar jawaban komputer secara manual, akan tetapi kekurangannya membuat banyak waktu terbuang dan dapat menyebabkan kesalahan dalam mengoreksi serta pengerjaan yang lambat.

OMR (*Optical Mark Recognition*) itu sendiri merupakan kumpulan informasi dari dokumen dengan mengenali karakter pada kertas atau pembacaan tanda optik. OMR (*Optical Mark Recognition*) adalah teknik untuk mengumpulkan data yang telah ditandai pada dokumen seperti kuesioner dan lembar ujian yang sudah ditandai (Tümer & Küçükkara, 2018).

Dalam proses pendeteksian lembar jawaban komputer diperlukan teknik pemrograman dimana membuat komputer dapat mengenali objek sebuah citra atau video dengan konsep *computer vision* yang dibekali pemindai OMR (*Optical Mark Recognition*). Tentunya OMR ini di implemetasikan menggunakan bahasa pemrograman *python*, sebab *python* adalah bahasa pemrograman yang memiliki sintaks yang jelas dan mudah dipelajari. Selain itu, *pyhton* dapat menggunakan library dan modul yang dapat membuat code menjadi lebih terstruktur dan mudah untuk diimplementasikan sehingga memiliki tingkat keefisienanan yang baik (Zulkhaidi, Maria, & Yulianto, 2020). Pada penelitian ini lebih ditekankan pada proses langkah-langkah membangun program pendeteksian lembar jawaban komputer dan menguji seberapa akurat tingkat akurasi menggunakan deteksi tepi metode *Canny* dalam sistem OMR (*Optical Mark Recognition*) pada lembar jawaban komputer yang dirancang menggunakan IDE *pycharm* dan library *opencv* yang sering digunakan dalam pengolahan citra *computer vision* (Isum, Maryati, & Tryatmojo, 2019). Sebab operator *Canny* menggunakan Kernel Gaussian Derivative dapat menyaring noise dari gambar asli untuk mendeteksi objek dengan permukaan halus (Rizal Adi Saputra, Reskal, 2022).

Banyak penelitian yang merancang sebuah aplikasi OMR seperti pada penelitian tahun 2019 oleh Samuel Lukas pada penelitiannya menjelaskan sistem yang dirancang menggunakan bahasa pemrograman java itu memberikan hasil pengukuran terhadap ujian yang dilakukan dengan menunjukkan nilai validitas, reliabilitas, serta tingkat kesukaran. Berdasarkan percobaan yang telah dilakukan, penilaian lembar jawaban memiliki akurasi sebesar 99.6% yang dipengaruhi oleh ketepatan pengenalan tanda X sebesar 99.3% dan tanda X yang dikoreksi sebesar 99.8% (Samuel Lukas, Hendra Tjahyadi, Dion Krisnadi, 2019). Namun pada penelitian tersebut tidak membahas metode apa yang digunakan dalam merancang sistem OMR tersebut. Selanjutnya, pada penelitian tahun 2022 yang berjudul “*Mobile Optical Mark Recognition Application As A Non- Inferior Alternative To Manual*

Marking For Lower- And Middle-Income Countries”, dalam kesimpulan hasil penelitian tersebut menunjukkan bahwa hasil pendeteksian lembar jawaban secara manual menggunakan aplikasi pihak ketiga yang diinstall pada *smartphone* memiliki rata-rata persen akurasi adalah 99,78%, dengan standar deviasi 0,11% dan standar error 0,05% sedangkan hasil lembar jawaban yang menggunakan *mobile optical recognition app* memiliki rata-rata persen akurasi adalah 99,90%, dengan standar deviasi 0,08% dan standar error 0,03%. (Optical et al., 2022).

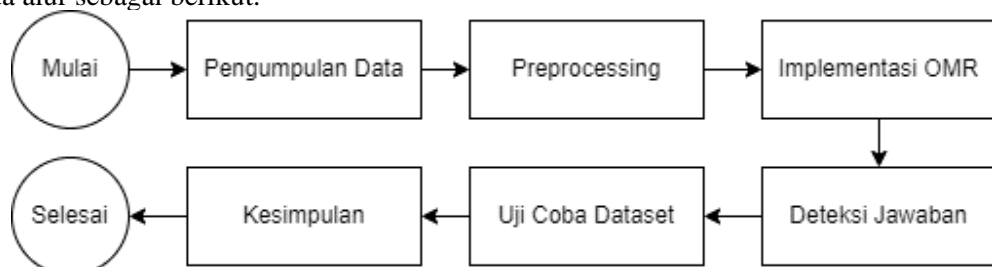
Selain itu penelitian berjudul “*Optical Mark Reading (OMR) Dengan Pencocokan Gambar Menggunakan Metode Fast Approximate Nearest Neighbors*” aplikasi yang dibangun menggunakan bahasa pemrograman *Python* dan library *OpenCV* yang ditulis pada tahun 2018 tersebut membahas bahwa ketika proses penilaian ujian menggunakan OMR (*Optical Mark Recognition*) yang telah diubah menjadi komponen aplikasi yang dapat mengonversi data dari gambar lembar jawaban komputer (LJK) dengan pengamanan berupa pencocokan logo antara gambar yang dikonfigurasi pada sistem dan pada gambar LJK menggunakan metode *fast nearest neighbor*. Pada pengujian pemeriksaan kelayakan citra LJK didapatkan bahwa persentase keberhasilan komponen sebesar 88,5% dan pada pengujian akurasi konversi data didapatkan bahwa persentase keberhasilan komponen adalah sebesar 100% yang berarti aplikasi ini diimplementasi dengan sangat baik (Nyoto, 2018).

Berdasarkan hasil review dari beberapa penelitian sebelumnya bahwa penerapan OMR (*Optical Mark Recognition*) dapat mempermudah mendeteksi lembar jawaban komputer dengan akurat tetapi pada penelitian sebelumnya tidak ada yang membahas perancangan sistem OMR yang dirancang menggunakan metode Canny serta kombinasi alur perancangan aplikasi ini, oleh karena itu penelitian ini akan membahas bagaimana sistem aplikasi teknologi OMR (*Optical Mark Recognition*) dibangun menggunakan *Integrated Development Environment (IDE) pycharm* yang sudah mendukung library *opencv* sebagai modul dengan menggunakan metode Canny sebagai deteksi tepi dan alur pengembangan sistemnya dari bahasa pemrograman *python*.

METODE

Metode Penelitian

Metode penelitian berisikan penjelasan alur dan tahapan yang akan dilakukan selama proses penelitian dilaksanakan. Metode penelitian berfungsi sebagai alur pedoman yang akan peneliti ikuti selama proses penelitian dilakukan agar konteks penelitian tetap berjalan sehingga tujuan penelitian berhasil dicapai. Dalam tahapan ini, terdapat tahapan-tahapan yang dilakukan dari proses pengumpulan data hingga memperoleh nilai pada lembar jawaban melalui proses implementasi teknologi OMR. Adapun tahapan metode penelitian dapat dilihat pada alur sebagai berikut.



Gambar 1 Metode Penelitian

Adapun penjelasan pada Gambar 1 diatas dapat dijelaskan pada penjelasan berikut ini.

- 1) Pengumpulan Data

Tahapan pengumpulan dataset merupakan tahapan dimana peneliti melakukan kegiatan pengumpulan data. Dimana data yang dikumpulkan ialah foto atau scan lembar jawaban berbentuk objektif.

2) Preprocessing

Tahapan preprocessing merupakan tahapan untuk menghasilkan citra yang lebih mudah dikenal pada saat melakukan implementasi OMR. Jadi pada tahapan ini, citra akan dipersiapkan sehingga mudah untuk diolah pada tahapan implementasi OMR. Secara umum, terdapat beberapa tahapan pada preprocessing ini yaitu tahapan transformasi perubahan citra asli menjadi citra greyscale, tahapan pembluran citra untuk menghindari terjadinya noise atau kehilangan pada objek citra tahapan deteksi tepi objek citra serta tahapan cropping pada citra.

3) Implementasi OMR

Tahapan implementasi OMR (*Optical Mark Recognition*) merupakan tahapan implementasi terhadap deteksi citra lembar jawaban. Pada implementasi OMR ini, dilakukan tahapan segmentasi threshold dimana tahapan tersebut dilakukan untuk memisahkan objek citra yang ingin dideteksi dengan menggunakan pemisahan melalui gelap atau terangnya sebuah objek.

4) Implementasi deteksi jawaban

Tahapan implementasi deteksi jawaban merupakan tahapan pengecekan jawaban pada citra lembar jawaban menurut arsiran yang telah dideteksi oleh tahapan threshold. Pada tahapan ini juga terdapat perhitungan melalui rumus penilaian. Adapun rumus penilaian yang digunakan pada lembar jawaban dapat dilihat pada rumus berikut

$$\text{Nilai} = \left(\frac{\text{Jumlah Jawaban Benar}}{\text{Jumlah Soal}} \right) \times 100 \quad (1)$$

5) Uji Coba

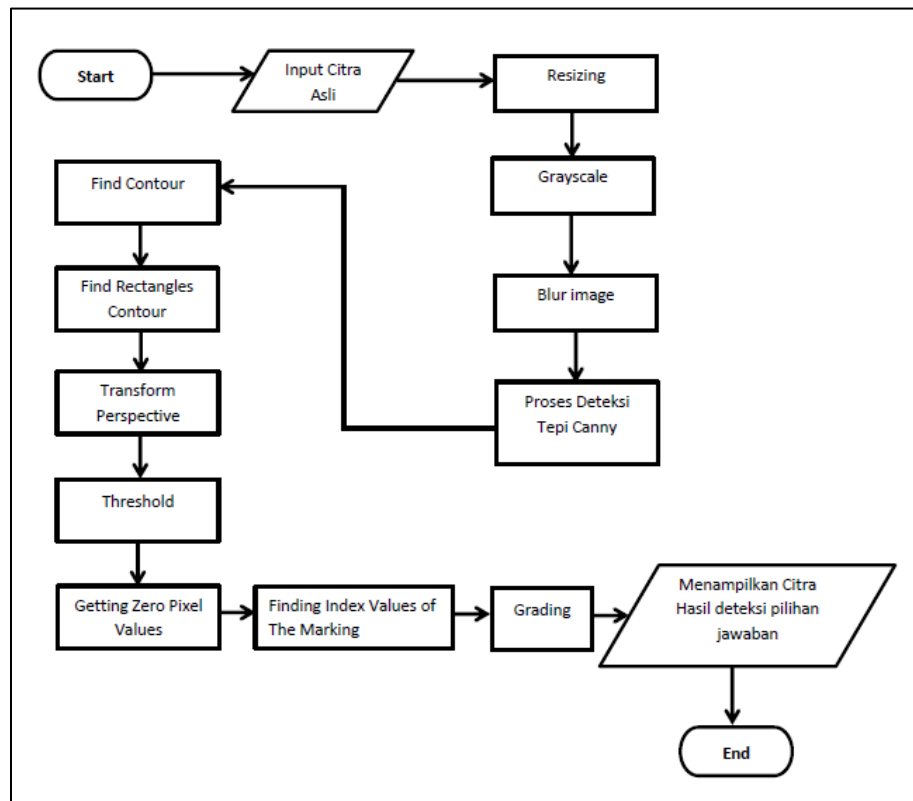
Tahapan uji coba merupakan tahapan uji coba terhadap 15 dataset citra lembar jawaban. Dimana masing masing komposisi citra yaitu 8 hasil foto lembar jawaban sebagai citra arsiran penuh, 5 hasil foto lembar jawaban sebagai citra arsiran separuh dan 2 hasil foto lembar jawaban sebagai citra arsiran salah. Pada tahapan uji coba ini, terdapat rumus akurasi terhadap deteksi citra yang dilakukan pendeteksian. Adapun rumus akurasi yang digunakan pada tahapan uji coba pada dataset dapat dilihat pada rumus berikut.

$$\text{Akurasi} = \left(\frac{\text{Jumlah Deteksi Benar}}{\text{Jumlah Total Soal}} \right) 100\% \quad (2)$$

6) Kesimpulan hasil uji coba

Tahapan kesimpulan hasil uji coba merupakan tahapan pemaparan hasil uji coba pada dataset citra lembar jawaban yang telah dilakukan.

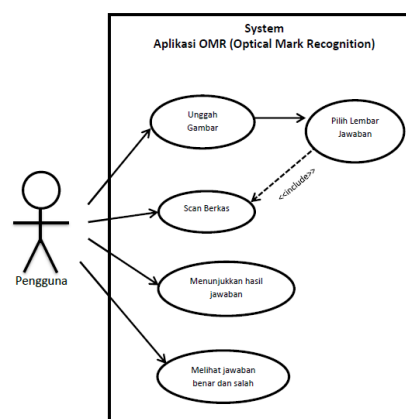
Perancangan Sistem Flowchart



Gambar 2 Alur Flowchart Aplikasi OMR dengan Algoritma Canny

Usecase Diagram dan User Interface

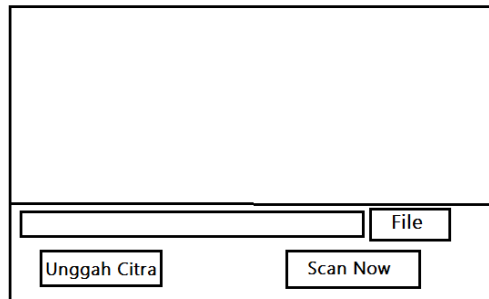
Aplikasi ini mengimplementasikan teknologi OMR (*Optical Mark Recognition*) untuk deteksi lembar jawaban. Aplikasi ini dirancang menggunakan *tools Integrated Development Environment (IDE) pycharm*, serta menggunakan bahasa pemrograman python dan menggunakan modul library *opencv*. Adapun user yang akan menggunakan aplikasi ini merupakan user pengguna. Dengan spesifik akses user pengguna disajikan pada gambar usecase diagram berikut ini.



Gambar 3 Usecase Diagram

User pengguna dapat melakukan pengunggahan gambar lembar jawaban melalui scan pada lembar jawaban yang akan dideteksi jawabannya. Selanjutnya user pengguna dapat melihat hasil penilaian yang diberikan oleh aplikasi terhadap deteksi jawaban pada lembar jawaban yang diunggah serta user pengguna dapat melihat hasil jawaban benar dan salah pada lembar jawaban didalam aplikasi.

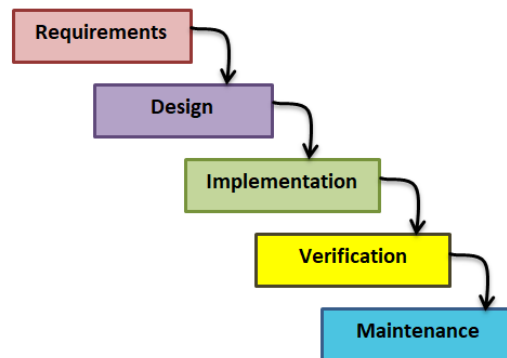
Tampilan implementasi *Userinterface* dari perancangan aplikasi yang dapat dilihat pada gambar berikut.



Gambar 4 Tampilan Antarmuka Aplikasi OMR

Model Pengembangan Sistem

Selanjutnya pada penelitian ini, dilakukan beberapa tahapan dalam perancangan aplikasi OMR (*Optical Mark Recognition*) menggunakan pustaka *opencv* dengan metode *Waterfall* dalam pembangunan aplikasi tersebut. *Waterfall* sendiri merupakan jenis model penelitian yang diperkenalkan ke dalam *classic life cycle* (siklus hidup klasik), yang berarti metodenya berurutan dan sistematis (Saputro & Kuddi, 2022). Dibawah ini merupakan gambar alur tahapan pada metode *waterfall*.



Gambar 5 Model *Waterfall*

Adapun penjelasan setiap proses metode *waterfall* pada Gambar 2 dijelaskan pada point dibawah ini.

a) Analisis Kebutuhan

Tahapan pertama yaitu Analisis kebutuhan yang sangat diperlukan dalam persiapan membangun aplikasi OMR pada lembar jawaban komputer dengan cara mencari tahu kebutuhan apa saja yang diperlukan. Dimulai dari kebutuhan

package open source, bahasa pemrograman hingga kebutuhan fungsional yang mencakup kebutuhan *hardware*, *software* dan kebutuhan informasi.

- b) Desain
Tahapan kedua ini, dilakukan perancangan visual seperti tampilan *user interface*, sebab elemen ini sangat penting dalam tahap *software development* untuk penyegaran mata. Dalam merancang tampilan UI pada aplikasi beberapa hal yang menentukan agar UI menjadi menarik, seperti penggunaan *font*, warna dan tata letak tombol hingga warna *background* yang digunakan dengan tujuan agar aplikasi terlihat lebih sempurna dari segi visual.
- c) Perancangan Kode Program
Tahapan ini melakukan perancangan terhadap sketsa desain yang sudah disusun pada tahapan kedua menjadi alur kode program dan *library* programming, selanjutnya akan diintegrasikan untuk dijadikan sebuah aplikasi yang sesuai dengan yang diharapkan.
- d) Pengujian
Tahapan ini berfokus pada segi fungsional hal ini dimaksudkan apakah aplikasi berjalan baik menurut fungsionalitas-fungsionalitasnya. Pengujian pada aplikasi ini dipastikan bahwa semua bagian program sudah berjalan dengan baik untuk meminimalisir kesalahan (*error*), sehingga program yang di bangun sesuai dengan keluaran yang dihasilkan (*output*).
- e) Pemeliharaan
Pada tahapan terakhir ialah pemeliharaan yaitu dimana jika terjadi kesalahan program atau bug pada aplikasi tersebut maka satu hal yang dilakukan yaitu tahap perbaikan untuk menunjang sistem kearah lebih baik lagi.

Algoritma Canny

Deteksi Canny ini memiliki pendeteksian paling optimum beberapa kriteria pada algoritma Canny(Pratama, Muhamad Rizki Bhayangkara & Ishlah, 2022):

- a. Mendeteksi dengan baik (kriteria deteksi)
Sesuai pada nilai parameter-parameter konvolusi yang dilakukan pada operator Canny ini mampu untuk menandai semua tepi yang ada. Sekaligus dapat menentukan tingkat pendeteksian pada ketebalan tepi sesuai yang diinginkan.
- b. Melokalisasi dengan baik (kriteria lokalisasi)
Dengan algoritma Canny sangat memungkinkan untuk menghasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.
- c. Respon yang jelas (kriteria respon)
Tiap tepi hanya ada satu respon sehingga mudah untuk dideteksi dan tidak menimbulkan kesalahan atau kerancuan pada pengolahan citra selanjutnya. Sebab itu dalam pemilihan parameter deteksi tepi Canny sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut antara lain:

1) Nilai Standart Deviasi Gaussian

2) Nilai Ambang

Pada konvolusi fungsi gambar dengan operator gaussian serta turunan-turunannya hanya bisa dilakukan oleh pendeteksian algoritma canny. Turunan pertama dari fungsi citra yang dikonvolusikan dengan fungsi gaussian,

$$g(x,y) = D[\text{gauss}(x,y) * f(x,y)]$$

ekivalen dengan fungsi citra yang dikonvolusikan dengan turunan pertama dari fungsi gaussian,

$$g(x,y) = D[\text{gauss}(x,y)] * f(x,y)$$

Oleh karena itu, tingkat kehalusan dan pendeteksian tepi sangat memungkinkan untuk mengkombinasi ke dalam suatu konvolusi dalam satu dimensi dengan dua arah yang berbeda (vertical dan horizontal)

Adapun tahapan yang dibuat dalam proses deteksi tepi sebagai berikut

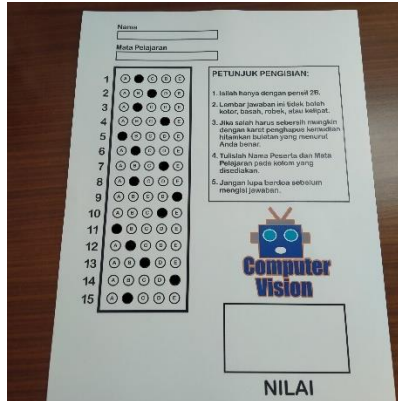
1. Pengenalan citra asli yang akan di proses deteksi tepi
2. Kemudian diubah citra menjadi greyscale
3. Haluskan citra dengan tujuan menghilangkan noise pada citra menggunakan filter gaussian blur
4. Melakukan operasi Canny
5. Memberikan nilai batas ambang

Selanjutnya hasil citra Canny akan diproses untuk pendeteksian lembar jawaban computer

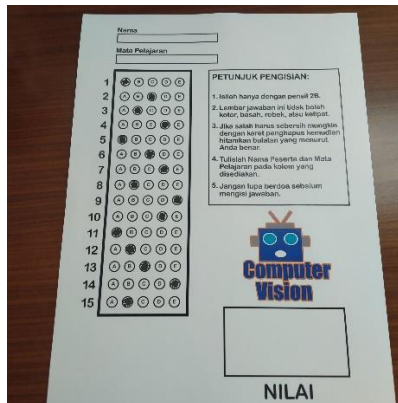
HASIL DAN PEMBAHASAN

Pengumpulan Dataset

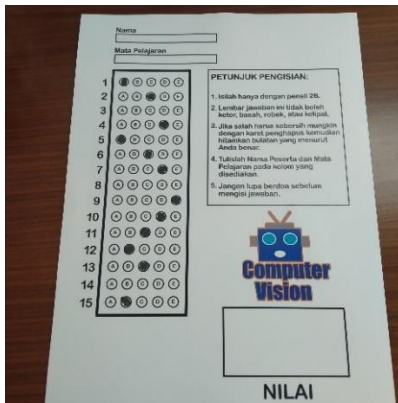
Tahapan pengumpulan dataset merupakan tahapan pengumpulan dataset dimana dataset yang digunakan berjumlah 15 gambar hasil foto lembar jawaban. Dimana masing-masing komposisi gambar yaitu 8 hasil foto lembar jawaban sebagai citra arsiran penuh, 5 hasil foto lembar jawaban sebagai citra arsiran separuh dan 2 hasil foto lembar jawaban sebagai citra arsiran salah. Adapun kumpulan gambar hasil foto lembar jawaban dapat dilihat pada gambar berikut ini.



Gambar 6 Hasil Foto Lembar Jawaban Arsiran Penuh



Gambar 7 Hasil Foto Lembar Jawaban Arsiran Separuh



Gambar 8 Hasil Foto Lembar Jawaban Arsiran Salah

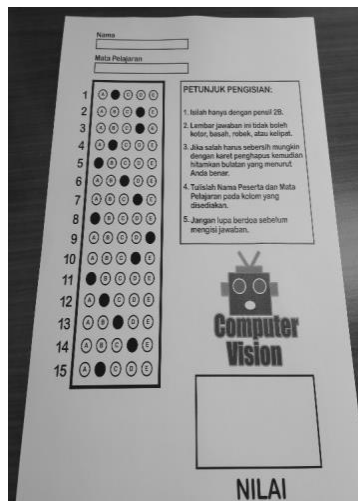
Tahapan Preprocessing

Tahapan selanjutnya yang dilakukan pada dataset yang sebelumnya telah dikumpulkan ialah melakukan tahapan preprocessing. Tahapan ini akan menghasilkan citra yang lebih mudah dikenal pada saat melakukan implementasi OMR. Jadi pada tahapan ini, citra akan dipersiapkan untuk di input ke aplikasi tersebut dalam format *jpeg*, *jpg* ataupun *png* sehingga mudah untuk diolah pada tahapan implementasi OMR.



Gambar 9 Tampilan citra setelah diinput ke aplikasi OMR

Secara umum, terdapat beberapa tahapan pada preprocessing ini yaitu tahapan transformasi perubahan citra asli menjadi citra greyscale, tahapan pemburlan citra untuk menghindari terjadinya noise atau kehilangan pada objek citra tahapan deteksi tepi objek citra serta tahapan cropping pada citra. Citra lembar jawaban akan dilakukan perubahan atau transformasi menjadi citra greyscale. Adapun hasil citra pada tahapan transformasi greyscale ini dapat dilihat pada gambar berikut.



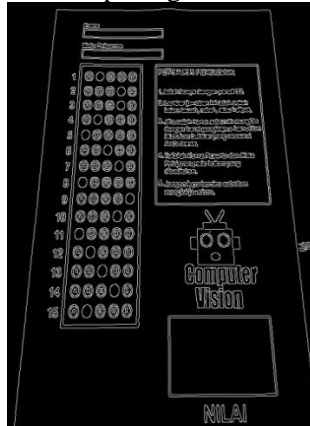
Gambar 10 Hasil Tahapan Greyscale pada Citra Lembar Jawaban

Tahapan selanjutnya ialah melakukan pemburlan menggunakan filter *gaussian blur* pada citra lembar jawaban agar menghilangkan noise pada citra. Adapun hasil citra pada tahapan blur pada citra ini dapat dilihat pada gambar berikut.



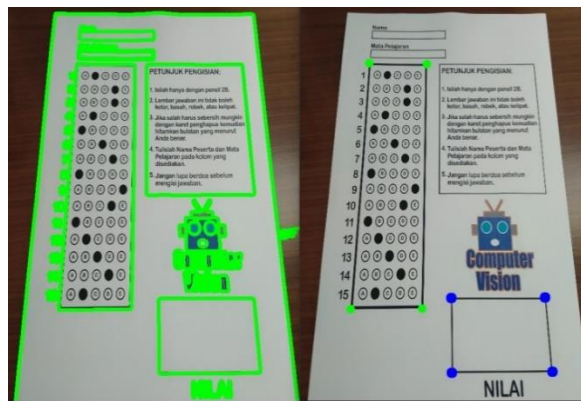
Gambar 11 Hasil Tahapan Blur pada Citra Lembar Jawaban

Tahapan selanjutnya ialah melakukan deteksi tepi objek citra lembar jawaban. Tahapan deteksi objek citra ini menggunakan metode canny. Adapun hasil citra pada tahapan metode canny pada citra ini dapat dilihat pada gambar berikut.



Gambar 12 Hasil Tahapan Deteksi Tepi Objek pada Citra Lembar Jawaban

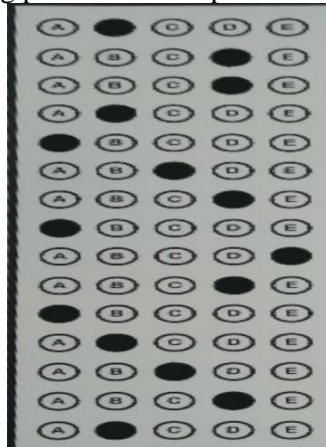
Tahapan selanjutnya ialah melakukan deteksi Contours dan Biggest Contours pada citra lembar kerja jawaban. Adapun hasil citra pada tahapan deteksi Contours dan Biggest Contours pada citra ini dapat dilihat pada gambar berikut.



Gambar 13 Hasil Tahapan Deteksi Contours dan Biggest Contours pada Citra Lembar Jawaban

Tahapan selanjutnya ialah melakukan cropping pada citra lembar jawaban. Adapun

hasil citra pada tahapan cropping pada citra ini dapat dilihat pada gambar berikut.

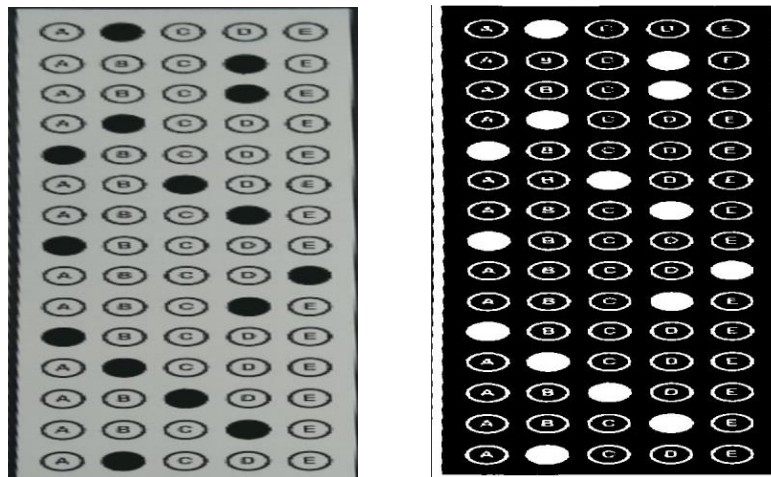


Gambar 14 Hasil Tahapan Cropping pada Citra Lembar Jawaban

Selanjutnya, apabila telah dilakukan maka citra telah siap digunakan pada tahapan selanjutnya yaitu implementasi OMR.

Tahapan Implementasi OMR

Pada tahapan implementasi OMR ini, terdapat beberapa tahapan sebelum dilakukannya deteksi jawaban pada citra lembar jawaban. Setelah dilakukan tahapan deteksi tepi objek citra pada preprocessing, maka tahapan selanjutnya ialah melakukan segmentasi threshold pada citra. Tahapan segmentasi threshold ini dilakukan untuk memisahkan objek citra yang ingin dideteksi dengan menggunakan pemisahan melalui gelap atau terangnya sebuah objek. Adapun hasil dari segmentasi threshold pada citra serta perbandingan hasil segmentasi threshold dan tahapan cropping dapat dilihat pada gambar berikut ini.



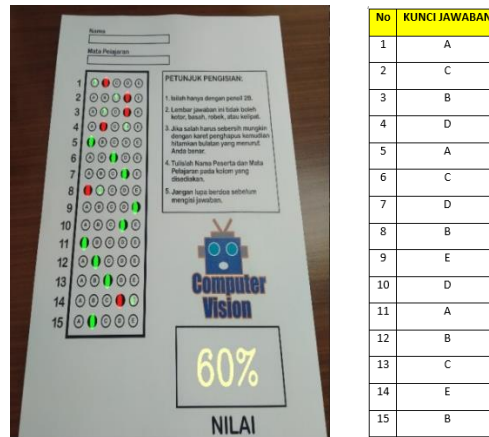
Gambar 15 Hasil Tahapan Threshold pada Citra Lembar Jawaban

Selanjutnya, setelah melakukan tahapan segmentasi threshold, maka dilakukan tahapan deteksi jawaban pada citra lembaran jawaban.

Tahapan Deteksi Jawaban pada Lembar Jawaban

Tahapan ini dilakukan pendeteksi jawaban pada citra lembar jawaban yang telah dilakukan tahapan preprocessing dan tahapan implementasi OMR. Tahapan ini mendeteksi

jawaban melalui arsiran jawaban yang terdapat pada lembar jawabanf. Jawaban yang terdeteksi akan memberikan penilaian apakah jawaban yang diarsir termasuk kategori jawaban benar atau salah. Adapun hasil deteksi jawaban pada lembar jawaban dapat dilihat pada gambar berikut.



Gambar 16 Hasil Tahapan Deteksi Jawaban pada Lembar Jawaban

Dilihat dari Gambar 12 diatas, tanda merah merupakan jawaban yang memiliki kategori salah diketahui dari deteksi arsiran jawaban pada lembar jawaban. Dan tanda hijau merupakan jawaban yang memiliki kategori benar diketahui dari deteksi arsiran jawaban pada lembar jawaban. Proses ini dilakukan dengan penyikronan indeks jawaban pada lembar jawaban terhadap file kunci jawaban yang telah disediakan. Setelah dilakukan deteksi jawaban melalui kategori jumlah benar dan salah, maka akan diberikan skor pada kolom nilai yang telah dideteksi sebelumnya pada tahapan preprocessing. Adapun proses pemberian skor pada lembar jawaban yang dideteksi melalui rumus perhitungan sebagai berikut:

$$\text{Nilai} = \left(\frac{\text{Jumlah Jawaban Benar}}{\text{jumlah Soal}} \right) \times 100$$

$$\text{Nilai} = \left(\frac{9}{15} \right) \times 100 = 60$$

Tahapan Uji Coba

Tahapan ini akan dilakukan uji coba terhadap semua dataset uji coba yang disediakan dan digunakan pada penelitian ini. Pada tahapan uji coba ini akan dilakukan proses deteksi lembar jawaban pada 8 hasil foto lembar jawaban sebagai citra arsiran penuh, 5 hasil foto lembar jawaban sebagai citra arsiran separuh dan 2 hasil foto lembar jawaban sebagai citra arsiran salah. Dengan tabel uji coba pada seluruh citra sebagai berikut.

Dengan didapatlah hasil akurasi pada uji coba pada semua citra yang dilakukan ialah 93% melalui rumus perhitungan sebagai berikut.

$$\text{Akurasi} = \left(\frac{\text{Jumlah Deteksi Benar}}{\text{Jumlah Total Soal}} \right) 100\%$$

$$\text{Akurasi} = \left(\frac{210}{225}\right) 100\% = 93\%$$

KESIMPULAN

Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa rancangan lembar jawaban berhasil dikenali oleh sistem. Pengenalan yang diterapkan untuk mengenali lembar jawaban tersebut terdiri dari tiga tahap, yaitu preprocessing, implementasi OMR, dan tahap uji coba. Aplikasi ini diimplementasikan menggunakan teknologi OMR (Optical Mark Recognition) deteksi lembar jawaban komputer dengan menggunakan algoritma canny yang tergolong cukup baik dalam proses deteksi tepi, sehingga hasil dalam proses deteksi lembar jawaban semakin akurat. Aplikasi ini pun telah berhasil dirancang menggunakan tools Integrated Development Environment (IDE) pycharm, menggunakan bahasa pemrograman python dengan memanfaatkan library opencv. Penelitian ini menggunakan 15 dataset citra lembar jawaban. Dimana masing masing komposisi citra yaitu 8 hasil foto lembar jawaban sebagai citra arsiran penuh, 5 hasil foto lembar jawaban sebagai citra arsiran separuh dan 2 hasil foto lembar jawaban sebagai citra arsiran salah. Adapun hasil akurasi yang diberikan pada uji coba semua citra yang dilakukan ialah 93%.

REFERENCES

- Chaudhary, Ruchika, Patel, Anuj, Kumar, Sushil, & Tomar, Sanjeev. (2017). Edge detection using particle swarm optimization technique. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017, 2017-Janua(1)*, 363–367. <https://doi.org/10.1109/CCAA.2017.8229843> [Google Scholar](#)
- Fawwaz, Insidini, & Dharshinni, N. P. (2021). *Jurnal Sistem Informasi dan Teknologi Jaringan Perbandingan Deteksi Tepi Citra Menggunakan Operator Robert , Canny , dan Frei Chen Pada Citra Bitmap dan JPEG*. 2, 41–45. [Google Scholar](#)
- Gautama, Tjatur Kandaga, Hendrik, Antonius, & Hendaya, Riskadewi. (2016). Pengenalan Objek pada Computer Vision dengan Pencocokan Fitur Menggunakan Algoritma SIFT Studi Kasus: Deteksi Penyakit Kulit Sederhana. *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(3), 437–450. <https://doi.org/10.28932/jutisi.v2i3.554> [Google Scholar](#)
- Hermawan, Moh Novi. (2021). Deteksi Lembar Jawaban Komputer Menggunakan Omr (Optical Mark Recognition) Di Mts Nurul Iman. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(3), 1361–1372. <https://doi.org/10.35957/jatisi.v8i3.1078> [Google Scholar](#)
- Isum, Raden, Maryati, Suryani, & Tryatmojo, Burhanuddin. (2019). *Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi Dengan Metode Haar Cascade*. (Cv). [Google Scholar](#)

- Kaloh, Karina M., Poekoel, Vecky C., & Putro, Muhamad D. (2018). Perbandingan Algoritma Background Subtraction dan Optical Flow Untuk Deteksi Manusia. *Jurnal Teknik Informatika*, 13(1), 1–9. <https://doi.org/10.35793/jti.13.1.2018.20186> [Google Scholar](#)
- Kuswandi, Eko Rahmad, & Fadillah, Nurul. (2019). Perbandingan Metode Robert dan Metode Prewitt untuk Deteksi Tepi pada Citra Tanda Tangan. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 3(2), 155–158. <https://doi.org/10.30743/infotekjar.v3i2.997> [Google Scholar](#)
- Nyoto, Rudy Dwi. (2018). *Optical Mark Reading (OMR) Dengan Pencocokan Gambar Menggunakan Metode Fast Approximate Nearest Neighbors Edy Tanto , Arif Bijaksana Putra Negara , .* 444–450. [Google Scholar](#)
- Optical, Mobile, Recognition, Mark, As, Application, Alternative, Inferior, Manual, To, For, Marking, & Countries, Middle income. (2022). *Mobile Optical Mark Recognition Application As A Non- Inferior Alternative To Manual Marking For Lower- And Middle-Income Countries open access.* 2022, 10–19. [Google Scholar](#)
- Pratama, Muhamad Rizki Bhayangkara, Erlangga Putra, & Ishlah, Juan Mohammad. (2022). MODEL APLIKASI DOCUMENT SCANNER MENGGUNAKAN OPERATOR. *MODEL APLIKASI DOCUMENT SCANNER MENGGUNAKAN OPERATOR CANNY DAN CONTOUR PADA OPEN CV BERBASIS DESKTOP*, 10(2). [Google Scholar](#)
- Riadi, Imam, Yudhana, Anton, & Sulisty, Wicaksono Yuli. (2020). Analisis Perbandingan Nilai Kualitas Citra pada Metode Deteksi Tepi. *Rekayasa Sistem Dan Teknologi Informasi*, 4(2), 345–351. [Google Scholar](#)
- Rizal Adi Saputra, Reskal, Frida Mimi Wahyuni. (2022). *Segmentasi Pada Plat Kendaraan Dinas dengan.* 6, 328–339. [Google Scholar](#)
- Sampebua, Mingsep R., & Membala, Samuel B. (2018). Penerapan aplikasi ujian berbasis komputer untuk meningkatkan kualitas pendidikan pada SMP. *Jurnal Pengabdian Papua*, 2(1), 24–31. [Google Scholar](#)
- Samuel Lukas, Hendra Tjahyadi, Dion Krisnadi, Steven Albert. (2019). Perancangan Aplikasi OMR berbasis Android. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, 16(2), 116–125. [Google Scholar](#)
- Saputro, Agung Dwi, & Kuddi, Bobi Frans. (2022). *Pembuatan Aplikasi Try Out Cat (Computer Assisted Test) Penerimaan Pegawai Negeri Sipil Bidang Tes Intelegensi Umum Berbasis Desktop.* 6, 467–475. [Google Scholar](#)
- Tümer, Abdullah Erdal, & Küçükçara, Zeki. (2018). An Image Processing Oriented Optical Mark Recognition and Evaluation System. *International Journal of*

Applied Mathematics, Electronics and Computers, 6(4), 59–64.
<https://doi.org/10.18100/ijamec.2018447788> [Google Scholar](#)

Zulkhaidi, Tengku Cut Al Saidina, Maria, Eny, & Yulianto, Yulianto. (2020).
Pengenalan Pola Bentuk Wajah dengan OpenCV. *Jurnal Rekayasa Teknologi
Informasi (JURTI)*, 3(2), 181. <https://doi.org/10.30872/jurti.v3i2.4033> [Google
Scholar](#)