

APLIKASI ANDROID KOREKSI LEMBAR JAWABAN KOMPUTER MENGUNAKAN OPENCV

Android Application for Correcting Computer Answer Sheet using OpenCV

Yusril Ichsan Hernafi ¹, Tengku A. Riza, S.T., M.T ², Hafidudin, S.T., M.T ^{3 1,2,3} Prodi D3 Teknologi Telekomunikasi, Universitas Telkom

Riryichsan2@gmail.com, tengkuriza@telkomuniversity.ac.id

ABSTRAK

Lembar jawaban komputer (LJK) merupakan sebuah inovasi untuk mempercepat proses pemeriksaan hasil ujian yang ada, dengan memanfaatkan teknologi OMR (*Optical Mark Reader*) kita dapat melakukan koreksi dengan cepat, hanya dengan menggunakan komputer dan alat scanner. Umumnya OMR (*Optical Mark Reader*) yang ada sekarang masih menggunakan scanner dan komputer sebagai alat untuk melakukan koreksi pada LJK (Lembar Jawaban Komputer).

Proyek Akhir ini yang berjudul "Aplikasi Android Koreksi Lembar Jawaban Komputer Menggunakan OpenCV" bertujuan untuk membuat suatu aplikasi yang dapat menggantikan fungsi scanner dan komputer dalam melakukan koreksi jawaban, dengan menggunakan teknik *image processing* yang terintegrasi dengan *OpenCV* (*Open Source Computer Vision Library*) sebagai program utama yang bertindak sebagai otak untuk mengolah hasil dari LJK (Lembar Jawaban Komputer), dengan menggunakan *software* Android Studio untuk mengembangkan aplikasi Android. Dengan memanfaatkan kamera *Smartphone* sebagai media dalam melakukan *scanning* pada LJK (Lembar Jawaban Komputer), kamera akan memotret LJK (Lembar Jawaban Komputer) dan *OpenCV* akan mendeteksi tiap butir jawaban yang dipilih pada opsi jawaban. Dan selanjutnya hasil dari total jawaban yang benar akan ditampilkan pada tampilan aplikasi Android.

Pengolahan citra pada sistem aplikasi ini membaca opsi jawaban dengan menggunakan metode perhitungan jumlah pixel hitam. Akurasi sistem dalam mendeteksi jawaban yang dihitamkan dengan menggunakan pensil jenis 2B memiliki presentasi keberhasilan deteksi sebesar 100%, sedangkan akurasi mendeteksi jawaban dengan tingkat keberhasilan terendah, yaitu dengan menggunakan jenis pensil 3H, dengan tingkat keberhasilan rata-rata sebesar 4%. Sistem pada aplikasi dapat bekerja secara optimal dengan intensitas cahaya sebesar 50 Lux, dengan jarak optimal objek dan kamera sejauh 17 cm.

Kata Kunci : *Android, Image Processing, OpenCV, Lembar Jawaban Komputer.*

ABSTRACT

The computer answer sheet (LJK) is an innovation to speed up the process of examining existing exam results, by utilizing OMR (Optical Mark Reader) technology we can make corrections quickly, only by using a computer and a scanner. In general, the OMR (Optical Mark Reader) currently still uses scanners and computers as a tool to make corrections to the LJK (Computer Answer Sheet).

This Final Project entitled "Computer Application Answer Sheet Correction Using OpenCV" aims to create an application that can replace the functions of scanners and computers in correcting answers, using image processing techniques integrated with OpenCV (Open Source Computer Vision Library) as a program The main act as the brain to process the results of LJK (Computer Answer Sheet), using Android Studio software to develop Android applications. By utilizing the Smartphone camera as a medium for scanning the LJK (Computer Answer Sheet), the camera will take pictures of the LJK (Computer Answer Sheet) and OpenCV will detect each answer item selected in the answer options. And then the results of the total correct answers will be displayed on the Android application display.

Processing images on this application system reads the answer options using the method of calculating the number of black pixels. The accuracy of the system in converting answers blacked out using type 2B pencil has a 100% success detection presentation, while verification of approval with the lowest success rate, namely using 3H type pencil, with an average acquisition rate of 4%. The system in the application can work optimally with a light intensity of 50 Lux, with an optimal distance of objects and a camera of 17 cm.

Keywords: *Android, Image Processing, OpenCV, Computer Answer Sheet.*

BAB I PENDAHULUAN

1.1 Latar Belakang

Lembar Jawaban Komputer (LJK) lembar isian yang berupa kertas, dimana umumnya digunakan pada ujian berbentuk pilihan ganda. Bentuknya yang ringkas ini memudahkan pemeriksa melakukan evaluasi meskipun dengan jumlah soalnya yang banyak. Kegunaan dari LJK ini secara luas dimanfaatkan di berbagai bidang pekerjaan sampai pendidikan Dalam melakukan pemeriksaan pengguna membutuhkan *software* (OMR) Optical Mark Reader dan sebuah alat pemindai serta komputer yang saling terhubung. Seperti pada penelitian yang sudah pernah dilakukan [1] dan [2] dengan masih memanfaatkan teknologi pemindai dan webcam dan masih menggunakan laptop atau PC. Dengan metode ini masih dinilai kurang efisien dikarenakan dimensi pemindai dan PC masih tergolong tidak mudah untuk dibawa kemana saja.

Perkembangan zaman yang semakin maju ini, bidang teknologi juga semakin maju dalam menciptakan inovasi-inovasi yang dapat membantu manusia dalam meringankan pekerjaan dan membuatnya semakin efisien. Bidang teknologi masa ini telah memperkenalkan sebuah inovasi baru yang disebut sebagai *Artificial Intelegent* (AI) atau disebut juga kecerdasan buatan, dimana teknologi kecerdasan buatan ini mampu meniru dari kemampuan pola pikir manusia dalam memecahkan masalah dan mengenali objek, dan mengolah suatu pola tertentu mejadi suatu informasi [3]. *Image processing* merupakan salah satu cabang dalam bidang kecerdasan buatan, dimana pada awalnya teknik ini berfungsi dalam memperbaiki dan mengubah suatu kualitas dari suatu citra, dengan gabungan kecerdasan buatan dan teknik *Image processing* ini, manusia sekarang dapat memperoleh informasi dari suatu bentuk gambar .

Merujuk pada permasalahan efisiensi dan mobilisasi perangkat dalam melakukan evaluasi suatu lembar jawaban. Maka pada proyek akhir ini akan dibuat aplikasi koreksi hasil ujian pada smartphone menggunakan aplikasi Android, dimana aplikasi ini dapat dengan mudah digunakan oleh siapapun dan dimanapun karena efisien dan aplikatif. Aplikasi ini menggunakan *OpenCV library* yang akan mendeteksi jawaban dari setiap pilihan dan kemudian mencocokkannya dengan kunci jawaban. Oleh karena itu, aplikasi yang dibuat diharapkan mampu membantu *user* dalam melakukan koreksi hasil isian dengan efisien.

1.2 Tujuan dan Manfaat

Adapun tujuan dan manfaat dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Dapat melakukan koreksi isian jawaban melalui aplikasi *Smartphone Android* secara *Real-time*.

2. Dapat melakukan pengolahan citra dengan pendekteksi tanda (*Mark*) menggunakan *library OpenCV*.

Sedangkan Manfaatnya :

1. Memudahkan pengguna dengan cepat menentukan hasil isian dan nilai skor.
2. Memudahkan *user* dalam penggunaan, karena diterapkan dalam bentuk aplikasi di *Smartphone Android*.
3. Menigkatkan efisien waktu bagi *user*.

1.3 Rumusan Masalah

Berdasarkan Latar Belakang yang telah diuraikan diatas, maka permasalahan yang dapat dirumuskan adalah sebagai berikut:

1. Bagaimana cara perancangan sistem pada pengolahan citra dalam proses pembacaan *Mark* pada *library OpenCV*?
2. Bagaimana cara *library OpenCV* menghitung nilai akhir isian ?
3. Bagaimana cara menampilkan nilai hasil koreksi pada aplikasi di *Smartphone Android* ?

1.4 Batasan Masalah

Adapun Batasan Masalah pada Tugas Akhir ini adalah sebagai berikut:

1. Menggunakan kamera pada *Smartphone Android* dalam mendeteksi objek.
2. Menggunakan *OpenCV library* sebagai pengolahan citra.
3. Digunakan pada lembar jawaban dengan model Lembar Jawaban Komputer (LJK) dengan jumlah isian 5 nomor.
4. Perancangan menggunakan Bahasan Pemograman Java dan Software Android Studio.
5. Aplikasi yang dibuat diperuntukan bagi *Smartphone Android* dengan minimal OS versi Lollipop (5.0+).
6. Menggunakan Lembar jawaban yang dirancanagn sendiri.
7. Pengujian dilakukan dengan menggunakan alat tulis jenis pensil dengan tingkat kekerasan granit 2B,HB, dan 4H.

1.5 Metode Penelitian

Metode yang digunakan dalam pengerjaan Proyek Akhir ini,:

1. Studi Literatur

Pada pembuatan Proyek Akhir ini dilakukan pengumpulan data dengan cara studi pustaka dengan batasan materi mengenai bahasa pemrograman *java* dan android studio. Serta beberapa Literatur yang dapat dijadikan referensi dalam mewujudkan Proyek Akhir ini.

2. Perancangan dan Pembuatan Aplikasi

Dalam tahap ini, dilakukan perancangan dan pembuatan aplikasi yang akan digunakan untuk pengolahan citra pada Lembar Jawaban, berdasar penerapan teori-teori pada Studi Literatur.

3. Pengujian Aplikasi

Dalam Proses pengujian dilakukan pada fungsional per-substistem aplikasi dan aplikasi secara keseluruhan agar memperoleh hasil yang sesuai dengan harapan

4. Perbaikan Aplikasi

Pada Tahap ini, Apabila aplikasi telah melakukan tahap pengujian, dan tidak berjalan dengan sesuai kriteria dan parameter yang diharapkan, maka akan dilakukan perbaikan agar aplikasi dapat berjalan sesuai yang diharapkan.

5. Evaluasi

Setelah dilakukan Perbaikan, maka akan dilakukan pengukuran dari performansi aplikasi.

1.6 Sistematika Penulisan

Buku Proyek Akhir ini disusun ke dalam 5 Bab, Yaitu :

1. BAB I PENDAHULUAN

Pada bab ini membahas mengenai latar belakang, tujuan dan manfaat, rumusan masalah, batasan masalah dan metode penelitian.

2. BAB II DASAR TEORI

Bab ini berisi teori-teori yang menjadi dasar landasan dari permasalahan yang berhubungan dengan perancangan aplikasi ini.

3. BAB III PERCANCANGAN SISTEM

Bab ini menjelaskan mengenai langkah-langkah dalam menyelesaikan Proyek Akhir ini.

4. BAB IV HASIL DAN PENGUJIAN SISTEM

Bab ini menjelaskan tentang keluaran yang didapat dari hasil perencanaan serta analisa perencanaan dari hasil perencanaan dan pengujian yang dilakukan.

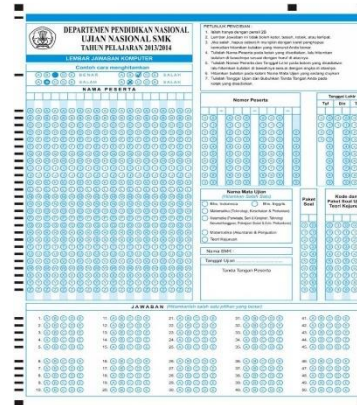
5. BAB V KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan dan saran dari hasil pengerjaan proyek akhir ini.

BAB II DASAR TEORI

2.1 Lembar Jawaban Komputer (LJK)

Lembar Jawaban Komputer (atau yg sering disingkat LJK) merupakan formulir isian berupa kertas yang kemudian akan dikenali isinya secara otomatis oleh komputer dengan alat scan. LJK biasanya digunakan untuk formulir yang berbentuk ujian pilihan ganda, kuesioner, formulir registrasi. Kegunaan dari LJK ini secara luas dimanfaatkan di berbagai bidang pekerjaan sampai Pendidikan. Pada gambar 2.1 ini kita dapat melihat contoh LJK.



Gambar 2.1 Contoh Model Lembar Jawaban Komputer

2.2 Pengolahan Citra digital

Derau (noise), kabur (blurring), dan kontras warna yang tajam, merupakan hal yang dapat menurunkan kualitas dari suatu citra dimana menyebabkan mesin maupun manusia sulit untuk menginterpretasikan citra yang ada. Suatu proses yang digunakan dalam mengolah citra yang mengalami penurunan kualitas disebut dengan *image processing* (pengolahan citra). Proses ini bertujuan agar manusia ataupun mesin tidak sulit untuk menginterpretasikan citra tersebut. Keluaran dari hasil pengolahan citra ini berupa citra juga, dimana masukan sebelumnya berupa citra juga, tetapi ketika sesudah diolah, keluaran citra yang didapatkan lebih baik dari sebelumnya [4].

2.3 Citra

Objek yang diterangi oleh suatu sumber cahaya dan berkas cahaya objek yang terpantul pada bidang dua dimensi dengan secara kontinu dinamakan citra. Citra yang ditangkap oleh alat-alat optik seperti kamera, pemindai (scanner), dan mata manusia disebabkan karena adanya intensitas pantulan cahaya dari objek yang merekam pererekam, memiliki sifat :

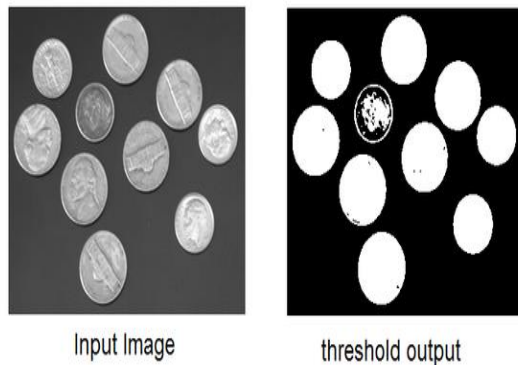
1. Sinyal Video (analog)
2. Foto
3. Digital (dapat disimpan secara langsung)

Citra Diam merupakan citra yang tidak bergerak yang hanya terdiri dari satu *Frame*. Citra Bergerak merupakan kumpulan *frame* dari citra diam yang berurutan dan bergerak sesuai dengan kecepatan fps (*frame per second*), sehingga membentuk suatu citra yang seolah bergerak [5] .

2.4 Thresholding

Suatu metode yang digunakan segmentasi dimana pada umumnya digunakan untuk memisahkan area yang utama gambar, dan latar

belakang gambar, dan digunakan dalam menentukan batas dari suatu gambar. Proses ini memisahkan bagian gambar (foreground) dengan latar belakangnya (background). Pemisahan antara objek dan latar belakang dengan mengubah citra warna ataupun grayscale dalam bentuk hanya hitam putih, dimana proses ini dilakukan dengan menetapkan nilai ambang batas (*threshold*) dan kemudian diubah menjadi dua intensitas yaitu 0 atau 1 [6].



Gambar 2.2 Contoh Hasil *Thresholding*

2.5 OpenCV

OpenCV (Open Source Computer Vision Library) merupakan program atau pustaka *open source* dari *computer vision*. Dimana pustaka nya ditulis dalam bahasa C dan C++ dan dapat berkerja pada sistem operasi Linux, Windows dan MacOS. OpenCV didesain untuk kemudahan dalam komputasi dan memfokuskan pada aplikasi realtime. OpenCV library memuat lebih dari 500 fungsi yang menjangkau banyak bidang pada penerapan *computer vision*, seperti inspeksi produk, pencitraan medis, keamanan, dan robotik [7]

2.6 Android Studio

Dalam membuat suatu Aplikasi Android kita membutuhkan suatu platform yang digunakan untuk mengembangkan dan meningkatkan produktifitas dalam mengembangkan suatu sistem operasi untuk perangkat mobile middleware yang disebut Android. Android studio merupakan bentuk dari kerjasama antara JetBrains dan Google, dimana Android Studio merupakan suatu perangkat lunak yang dibuat berdasar dari JetBrains IntelliJ yang membantu dalam membuat Aplikasi Android yang diinginkan [8]

Android Studio adalah IDE resmi dalam membuat dan mengembangkan Aplikasi Android berdasarkan IntelliJ IDE. IntelliJ IDE sendiri adalah Integrated Development Environment (IDE) yang dikembangkan oleh JetBrains, untuk

mengembangkan perangkat lunak komputer. IntelliJ IDE berfungsi untuk membantu pengembang dalam pemrograman baik dari segi navigasi, hingga *code editor* yang cerdas. Pembuatan *design* dari *interface* sebuah aplikasi pada setiap perangkat android [9].

2.7 Smartphone

Merupakan perangkat telpon genggam yang muda dibawa dan disimpan dimana saja, dengan bentuknya yang yang kecil dan kegunaan yang beragam, *Smartphone* hampir dapat disamakan dengan komputer, dimana kita dapat mengakses informasi dimana saja dan kapan saja [10]. Dengan perkembangan zaman, smartphone sendiri telah berkembang dengan pesat dalam hal penggunaan sistem operasi, pemrosesan yang lebih cepat, dan keamanan yang lebih baik dari sebelumnya [11].

2.8 Android

Android merupakan sistem operasi yang dikembangkan oleh Google, dengan dasar dari Linux kernel, dan didesain khusus untuk perangkat telepon genggam dengan spesifikasi layar sentuh. Middleware dan aplikasi inti yang dirilis oleh Google. Sebagai pelengkap nya berupa Android SDK (Software Development Kit) yang menyediakan Tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java [9].

2.9 Computer Vision

Manusia menggunakan mata dalam melihat objek yang ada disekitarnya, otak manusia akan membagi beberapa sinyal penglihatan, yang dimana akan mengalirkan berbagai jenis informasi ke otak, otak manusia akan mengidentifikasi keputusan apa yang akan dilakukan selanjutnya [12]. Computer vision sama halnya seperti mata manusia, dimana objek yang berupa gambar atau video akan diproses sesuai dengan model yang dibangun dalam mengotomatisasi dan mengintegrasikan proses yang akan menghasilkan dari persepsi penglihatan mesin [13].

2.10 Gaussian Blur

Gaussian Blur adalah Filter blur yang menempatkan warna transisi yang signifikan dalam sebuah image, kemudian membuat warna-warna pertengahan untuk menciptakan efek lembut pada sisi-sisi sebuah citra. Gaussian blur adalah salah satu filter blur yang menggunakan rumus matematika untuk menciptakan efek autofocus untuk mengurangi detail dan menciptakan efek berkabut [14].

2.11 Segmentasi Citra

Semua proses pengolahan citra umumnya bertujuan untuk mendapatkan visualisasi dan hasil

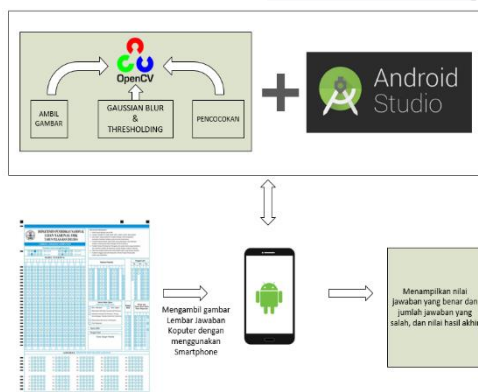
akurat dari objek yang ada. Segmentasi citra merupakan sebuah proses untuk pengelompokan sebuah citra digital menjadi beberapa bagian, dimana (set piksel atau dikenal juga sebagai super piksel) [15]. Lebih tepatnya segmentasi citra ini merupakan tahap proses dari menandai tiap piksel pada citra yang ada, dimana tiap tanda ini merepresentasikan karakteristik tiap piksel. Penerapan segmentasi citra ini digunakan dalam menandai objek dan batas seperti garis, kurva, dll pada citra. Umumnya segmentasi citra ini banyak diterapkan dalam bidang medis, deteksi objek, dan sistem kontrol lalu lintas. Segmentasi citra dapat dikategorikan menjadi 2 tipe yaitu, segmentasi lokal (terpusat hanya pada suatu bagian spesifik saja dalam suatu citra), sedangkan segmentasi global (terpusat pada keseluruhan citra, dengan jumlah piksel yang besar) [16].

BAB III PERANCANGAN SISTEM

Aplikasi yang dirancang ini diimplementasikan pada *smartphone* berbasis android. DIYScan merupakan nama dari aplikasi yang dibuat ini, dimana merupakan singkatan dari Do It Yourself Scan. Tujuan dari perancangan aplikasi ini, agar pengguna lebih mudah dalam melakukan evaluasi jawaban pada Lembar Jawaban Komputer.

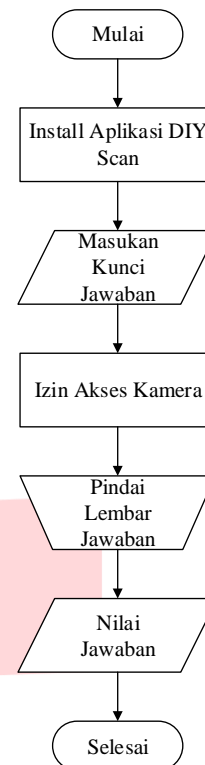
3.1 Blok Diagram Sistem

Pada gambar 3.1 merupakan blok diagram sistem dari aplikasi ini. Pada bab kali ini akan dijelaskan mengenai perancangan aplikasi android pemindai lembar jawaban komputer, yang dimana Android Studio dan OpenCV bertindak sebagai program utama dalam aplikasi ini :



Gambar 3.1 Blok diagram Perancangan Sistem

Android studio dan *OpenCV* 3.1.0 yang saling terintegrasi berfungsi membantu kamera *smartphone* saat aplikasi berjalan dalam proses pengolahan citra seperti mendeteksi lingkaran jawaban, sudut batas lembar jawaban. Saat proses pemindaian berlangsung, kamera pada aplikasi akan mengakuisisi dan menampilkan jumlah jawaban yang benar.



3.2 Flowchart sistem kerja Aplikasi

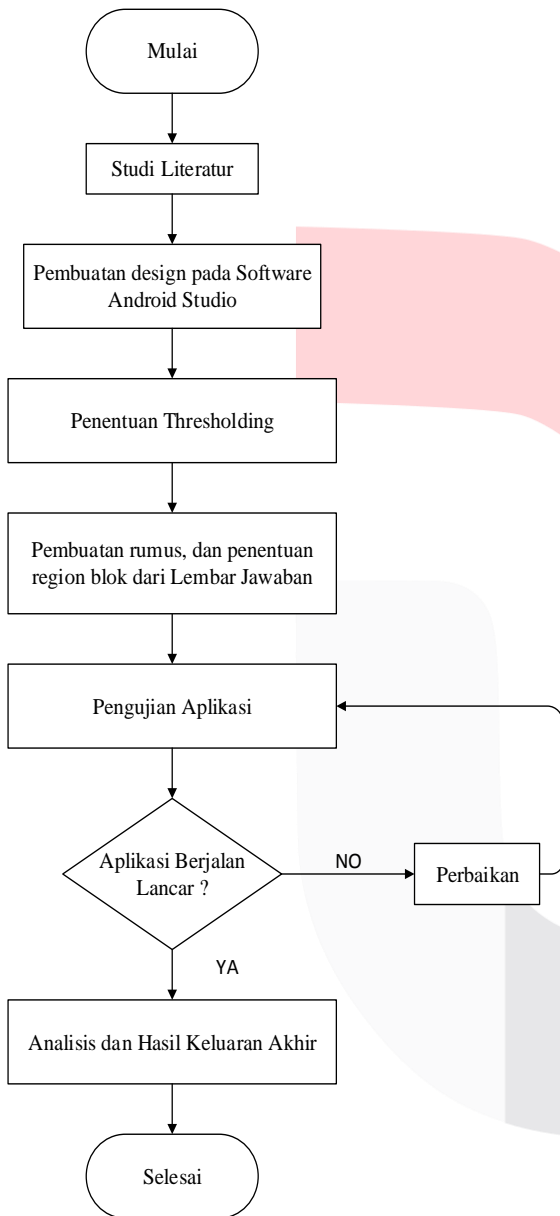
Sebagai program utama dalam aplikasi ini, *OpenCV* library memiliki peran penting, yaitu mendeteksi jumlah jawaban yang benar pada lembar jawaban, sub bab ini akan menjelaskan mengenai cara kerja dari aplikasi :

Gambar 3.2 Flowchart program pada *OpenCV* library

Aplikasi ini dirancang dimana sesaat sebelum pengguna aplikasi akan menggunakan aplikasi, pengguna akan diminta memasukkan kunci jawaban yang benar pada tampilan awal, setelah pengguna memasukkan kunci jawaban, maka pengguna dapat menekan tombol back pada *smartphone*, kunci jawaban akan otomatis tersimpan. Setelah proses memasukkan kunci jawaban, pengguna dapat sudah dapat melakukan proses pemindaian pada lembar jawaban. Pada proses pemindaian lembar jawaban, pengguna harus menyesuaikan jarak kamera dan kertas, hingga kamera dapat mengakuisisi sudut yang telah ditentukan, agar kamera dapat dengan mudah mengakuisisi citra pada saat pemindaian,

3.3 Flowchart Tahap Pengerjaan Aplikasi

Adapun flowchart tahap pengerjaan dari “Aplikasi Android Koreksi Lembar Jawaban Komputer Menggunakan *OpenCV*” yang dibuat adalah sebagai berikut:



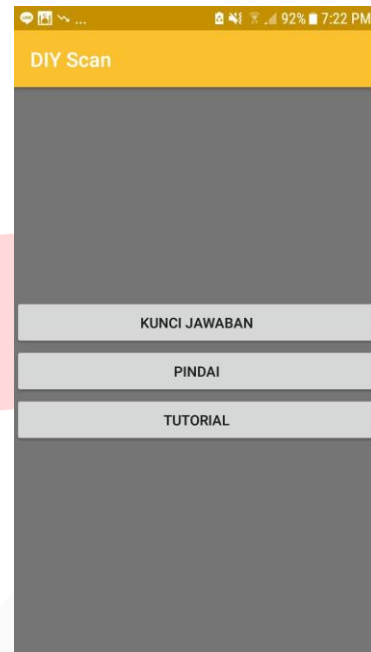
Gambar 3.3 Flowchart Tahapan Pengerjaan

3.4 User Interface pada Aplikasi DIYSan

Pada sub-bab ini membahas mengenai tampilan *interface* pada menu utama dan fungsi tombol-tombol yang ada pada aplikasi.

3.4.1 Home

Pada saat aplikasi dibuka, maka pengguna akan langsung terhubung dengan tampilan home, tanpa adanya halaman *splash screen*.

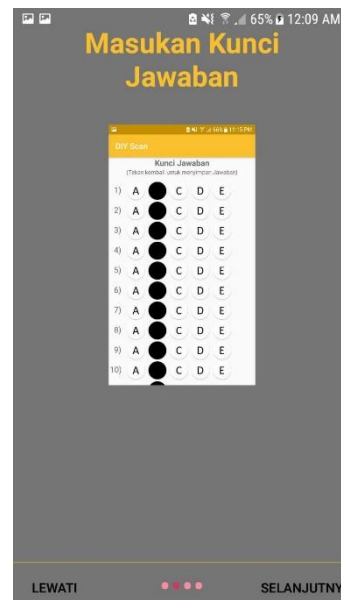


Gambar 3.4 Halaman *Splash screen*

Pada halaman Home ini user akan melihat tombol kunci jawaban, fungsi dari tombol kunci jawaban ini, dimana user dapat mengisi kunci jawaban yang benar. Terdapat Juga tombol Scan, fungsi dari tombol scan ini, dimana user dapat melakukan scan lembar jawaban.

3.4.2 Tutorial

Pada tombol tutorial aplikasi menampilkan langkah-langkah dan cara penggunaan aplikasi DIYSan



Gambar 3.5 Tampilan penggunaan Aplikasi DIYSan

3.5 Hardware dan Software Digunakan Pada Pembuatan Aplikasi

Dalam proses perancangan aplikasi yang berbasis OpenCV dan Android Studio ini, dibutuhkan Spesifikasi Hardware dan Software, yaitu sebagai berikut :

- 1.) Spesifikasi umum Smartphone untuk dapat menjalankan aplikasi DIYSan pada sistem operasi Android dengan versi minimal Lollipop (5+), Marshmallow (6+), Nougat (7+), Oreo (8+), Pie (9+)

2.) Smartphone Samsung A3

Spesifikasi perangkat keras :

Tabel 3.1 Spesifikasi Smartphone

Spesifikasi Smartphone	
Sistem Operasi	Android 7.0 Nougat
RAM	1 GB
Kamera Belakang	13 MP
Memori Internal	16 B

3.) Laptop

Spesifikasi Laptop

Tabel 3.2 Spesifikasi Laptop

Spesifikasi Laptop	
Processor	Intel® Core™ i5-3337U CPU @1.80Ghz
RAM	8 GB DDR3
Tipe Sistem	64-bit
Sistem Operasi	Windows 8.1 Pro

4.) Library OpenCV versi 3.4.5

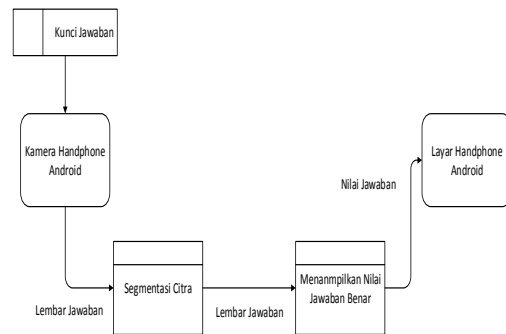
Library Opencv ini terintegrasi dengan software Android Studio. Open Computer Vision merupakan pustaka yang digunakan dalam pengolahan citra, sistem kerja OpenCV pada komputer hampir mirip dengan kemampuan visual pada manusia.

5.) Software Android Studio

Android Studio merupakan software yang digunakan dalam mengembangkan aplikasi Android. Android Studio menggunakan bahasa Java.

3.6 Diagram Data Flow Aplikasi DIYSan

Data Flow ini menunjukkan tahapan mengenai detail aplikasi yang setelah di unduh oleh pengguna, menunjukkan gambaran proses penggunaan aplikasi yang digunakan.



Gambar 3.7 Diagram data flow

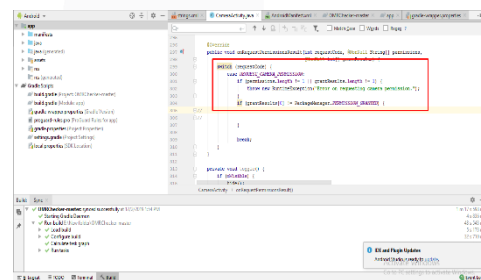
3.6.1 Desain Tampilan Aplikasi

Software android Studio merupakan software yang digunakan dalam pembuatan aplikasi DIYSan ini, berikut beberapa proses dalam pembuatan aplikasi ini :

3.6.1.1 Akses Kamera

Dalam aplikasi ini penggunaan kamera menjadi fungsi utama dalam proses pemindaian pada kertas lembar jawaban. User-permission Camera merupakan kode yang digunakan dalam mengakses kamera pada Software Android Studio, agar aplikasi dapat mengakses kamera saat dijalankan.

3.6.1.2 Pengolahan Citra pada Lembar

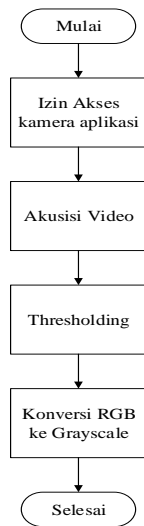


Gambar 3.6 User-permission

Jawaban

Pengolahan citra (preprocessing) pada aplikasi ini diperlukan agar dapat berlanjut ketahap pengenalan selanjutnya, dimana pengolahan citra pada tahap ini yaitu dengan memisahkan objek utama dan latar belakang gambar. Berikut flowchart pengolahan citra yang ditunjukkan pada gambar .

Gambar 3.8 Flowchart pengolahan citra



Ketika kamera aplikasi memindai objek, pengolahan citra dimulai dimana awal proses dimulai dari memisahkan objek dari *background*, atau disebut proses *thresholding*. Dimana pada proses *thresholding*-nya, ditampilkan dua citra, yaitu citra asli *grayscale* dan citra hasil *thresholding*, dimana nilai dari *thresholding* telah ditentukan, sehingga dapat dibedakan antara daerah *background* dan daerah objek. Fungsi *gaussian blur* juga diterapkan dengan tujuan untuk meredam noise yang ada pada gambar sehingga proses objek yang dipindai jelas dan utuh.

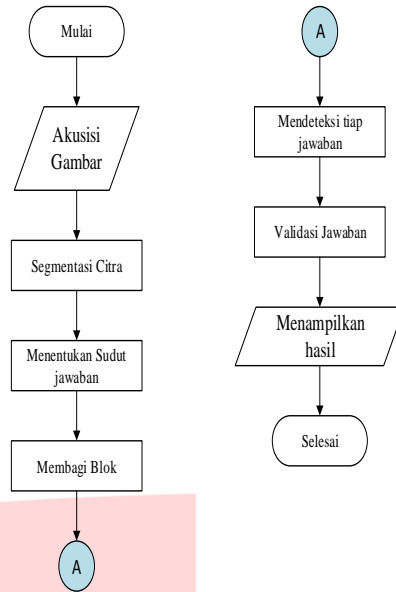
```

    Mat matGray = new Mat();
    Imgproc.cvtColor(matGaussianBlur, matGray, Imgproc.COLOR_RGB2GRAY);
    Mat matCircles = new Mat();

    Mat matThresholded = new Mat();
    Core.inRange(matGray, new Scalar(50,50,50), new Scalar(255,255,255), matThresholded);
    
```

3.6.1.3 Flowchart Menampilan Hasil Jawaban

Gambar 3. 9 Fungsi gaussian blur



Gambar 3. 10 Flowchart menampilkan hasil

Setelah melakukan pengolahan citra, gambar akan melalui proses penentuan bagian sudut dari lembar jawaban yang ada proses selanjutnya adalah mendeteksi bagian sudut dari lembar jawaban dengan menggunakan metode *Hough_Gradient*. Proses deteksi ini berfungsi dalam menemukan sudut pada lembar jawaban agar kamera dapat dengan mudah mengakusisi batasan dari lembar jawaban.

```

    Imgproc.HoughCircles(matGray, matCircles, Imgproc.CV_HOUGH_GRADIENT, dp: 0.9, minDist: matCircles.cols()/2, param1: 15, param2: 30);

    if(matCircles.cols() != 4){
        return null;
    }

    circle = new Circle(matCircles.cols());
    for (int x = 0; x < matCircles.cols(); x++) {
        double vCircle[] = matCircles.get( row: 0, x);

        if (vCircle == null)
            return null;
    }
    
```

Gambar 3.11 Proses deteksi

Setelah proses dari penentuan batas dari lembar jawaban selesai, proses selanjutnya yaitu dengan menggunakan fungsi *Converters.vector_Point2f_to_Mat* dimana dilakukan proses koreksi perspektif dengan memperbaiki posisi dan ukuran agar lebih memusatkan tampilan objek pada bagian tengah dengan tujuan tampilan akan lebih fokus ke objek yang akan dideteksi


```

Mat startM = Converters.vector_Point2f_to_Mat(src);

List<Point> dest = new ArrayList<>();

ptCornerPoints = getNewCornerPoints(previewWidth, previewHeight);

if (ptCornerPoints == null)
    return null;

for (int i=0; i< ptCornerPoints.length; i++) {
    dest.add(ptCornerPoints[i]);
}

Mat endM = Converters.vector_Point2f_to_Mat(dest);

Mat perspectiveTransform = Imgproc.getPerspectiveTransform(startM, endM);
    
```

Gambar 3.13 Converters.vector_Point2f_to_Mat

```

studentAnswers = new byte[numberOfQuestions][optionsPerQuestions];
for (int k = 0; k < omrSheet.getNumberofBlocks(); k++) {
    for (int i = 0; i < questionsPerBlock; i++) {
        for (int j = 0; j < optionsPerQuestions; j++) {
            Point pt[] = new OMRSheetUtil(omrSheet).getRectangleCoordinates(k, i, j);
            Point leftRectPoint = pt[0];
            Point rightRectPoint = pt[1];
        }
    }
}
    
```

Gambar 3.12 Proses pembagian tiap blok

Setelah memperbaiki tampilan kontur pada lembar jawaban, selanjutnya dilakukan proses pembagian tiap blok, jumlah nomor, dan jumlah pilihan jawaban yang ada. Hasil dari pembagian ini menghasilkan menjadi dua bagian blok, yaitu kanan dan kiri.

Setelah melakukan proses pembagian menjadi dua blok, selanjutnya. Terdapat proses pendeteksian lingkaran jawaban yang dipilih, dimana menggunakan fungsi *Core.countNonZero*, fungsi ini mengolompokan jumlah pixel hitam yang membentuk suatu lingkaran, dan menginisialisasi sebagai jawaban yang terdeteksi.

```

Rect rect = new Rect(leftRectPoint, rightRectPoint);
int noOfWhitePixels = Core.countNonZero(matThresholded.submat(rect));
int totalPixels = omrSheet.getTotalPixelsInBoundingSquare();
int noOfBlackPixels = totalPixels - noOfWhitePixels;
Log.d(tag, "noOfBlackPixels", msg: "i = " + (i + 1) + " j = " + (j + 1) + " " + noOfBlackPixels);
//NOT FILLED
if (noOfWhitePixels != totalPixels) {
    if (noOfBlackPixels >= omrSheet.getRequiredBlackPixelsInBoundingSquare()) {
        //FULL
        studentAnswers[i + (questionsPerBlock * k)][j] = OMRSheetConstants.CIRCLE_FILLED_FULL;
        System.out.println("noOfBlackPixels - " + "i = " + (i + 1) + " j = " + (j + 1) + " " + noOfBlackPixels);
    } else if (noOfBlackPixels >= thresholdNoOfBlackPixels) {
        //SEMI
        studentAnswers[i + (questionsPerBlock * k)][j] = OMRSheetConstants.CIRCLE_FILLED_SEMI;
        System.out.println("noOfBlackPixels - " + "i = " + (i + 1) + " j = " + (j + 1) + " " + noOfBlackPixels);
    }
}
    
```

Gambar 3.14 Core.countNonZero

Setelah melakukan deteksi jawaban yang

```

for (int i=0; i < numberOfQuestions; i++) {
    answerPerQuestion = 0;
    for (int j=0; j < optionsPerQuestions; j++) {
        if (correctAnswers[i] != 0 && (studentAnswers[i][j] == OMRSheetConstants.CIRCLE_FILLED_FULL || studentAnswers[i][j] == OMRSheetConstants.CIRCLE_FILLED_SEMI))
            answerPerQuestion++;
    }
}

if (answerPerQuestion == 1 && studentAnswers[i][correctAnswers[i]] == OMRSheetConstants.CIRCLE_FILLED_FULL) {
    score++;
}
    
```

dilingkar, maka dilakukan proses evaluasi jawaban, dimana dengan memanggil array *CIRCLE_FILLED_FULL* dimana fungsi ini memanggil dan menyimpan tipe lingkaran yang intensitas pengisiannya memenuhi seluruh lingkaran dengan tingkat kehitaman yang pekat.

Setelah dilakukannya proses evaluasi jawaban, dan dilakukannya pencocokan jawaban dengan kunci jawaban, langkah selanjutnya proses menampilkan nilai *score* dari jawaban yang benar, dan menampilkan deteksi jawaban yang benar, dan jawaban yang salah.

Gambar 3.15 Proses evaluasi jawaban

```

correctAnswers[i]=0;
if (answerPerQuestion == 1 && studentAnswers[i][correctAnswers[i]] == OMRSheetConstants.CIRCLE_FILLED_FULL) {
    camera.drawRect(new android.graphics.Rect((int) leftRectPoint.x, (int) leftRectPoint.y, (int) rightRectPoint.x, (int) rightRectPoint.y), greenPaint);
} else {
    for (int j = 0; j < optionsPerQuestions; j++) {
        pt = new OMRSheetUtil(omrSheet).getRectangleCoordinates((int) j / questionsPerBlock, (int) j % questionsPerBlock, j);
        leftRectPoint = pt[0];
        rightRectPoint = pt[1];
        camera.drawRect(new android.graphics.Rect((int) leftRectPoint.x, (int) leftRectPoint.y, (int) rightRectPoint.x, (int) rightRectPoint.y), redPaint);
    }
}

TextView textView = new TextView(mContext); // proses penampilan data
textView.setText(" Score: " + score);
textView.setTextColor(Color.BLACK);
textView.setTextSize(30);
    
```

3.7 Skenario Pengujian

Pada tahap menjelaskan beberapa tahapan-tahapan pengujian yang akan dilakukan uji fungsionalitas yang bertujuan dalam mengetahui tingkat akurasi dan error pada aplikasi DIY Scan.

3.7.1 Pengujian Fungsionalitas

Pengujian fungsionalitas ini merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui apakah aplikasi dapat diunduh dan dijalankan pada Handphone Android dengan merk berbeda dan operasi sistem yang berbeda.

3.7.2 Pengujian Jenis Pensil yang Digunakan

Gambar 3.16 Proses menampilkan score dan deteksi jawaban benar dan salah

Tahapan pengujian ini bertujuan untuk mengetahui jenis alat tulis apakah yang terbaik digunakan dalam mengisi jawaban pada lembar jawaban, sehingga aplikasi dapat mendeteksi tiap butir jawaban yang diisi dengan tepat.

3.7.3 Pengujian Akurasi Tiap Butir Jawaban

Tahapan ini bertujuan untuk mengetahui apakah aplikasi sudah dengan tepat mendeteksi tiap butir jawaban yang benar, yang sesuai dengan kunci jawaban yang ada pada kunci jawaban yang dimasukkan.

3.7.4 Pengujian Jarak dan Cahaya

Tahap Pengujian ini bertujuan mengetahui pengaruh kinerja aplikasi pada jarak dan intensitas cahaya yang ada disekitar. Hasil dari pengujian ini nantinya akan diperoleh tingkat keberhasilan dan kegagalan aplikasi dalam memindai lembar jawaban.

BAB IV

PENGUJIAN DAN ANALISA SISTEM

Pada Bab ini dipaparkan mengenai hasil pengujian sistem aplikasi DIY Scan. Tujuan dari pengujian ini dilakukan dengan tujuan mengetahui performa dari aplikasi.

4.1 Pengujian Fungsionalitas

Pengujian Fungsionalitas yang dilakukan pada aplikasi DIY Scan ini dengan cara mengunduh dan menjalankan aplikasi DIY Scan ini pada 10 tipe perangkat Android dengan merk serta sistem operasi yang berbeda. Hasil pengujian pemasangan aplikasi dapat dilihat pada tabel 1.4 dan Lampiran A

Tabel 4. 1 Pengujian Fungsionalitas

No.	Merk Perangkat Android	Sistem Operasi	Keterangan
1.	Samsung A3 2016	Nougat 7.0	Dapat dijalankan
2.	Asus Zenfone Maxpro M2	Nougat 7.2.1	Dapat dijalankan
3.	Realmi 3 Pro	Pie 9.0.0	Dapat dijalankan
4.	Oppo F7	Pie 9.0.0	Dapat dijalankan
5.	Xiaomi Redmi 3	Lollipop 5.1.1	Dapat dijalankan
6.	Oppo F7	Pie 9.0.0	Dapat dijalankan
7.	Samsung A6	Oreo 8.0.0	Dapat dijalankan

8.	Samsung J7 Pro	Oreo 8.2.1	Dapat dijalankan
9.	Samsung A5	Oreo 8.2.1	Dapat dijalankan
10.	Oppo F5 Series	Oreo 8.2.1	Dapat dijalankan

Berdasarkan dari hasil pengujian yang dilakukan pada tabel 4.1. Didapatkan kesimpulan bahwa aplikasi DIY Scan ini dapat berjalan pada spesifikasi Sistem Operasi Android dengan versi minimal Lollipop 5.0+. Pengujian fitur aplikasi juga dilakukan untuk mengetahui apakah fitur-fitur aplikasi dapat sebagaimana mestinya.

Tabel 4. 2 Pengujian fitur aplikasi

No	Aksi	Pengujian	Hasil Pengujian	Keterangan
1.	Mengunduh Aplikasi	Install pada Smartphone	Berhasil di Install pada Smartphone	Sesuai
2.	Menjalankan Aplikasi	Menampilkan <i>Splash Screen</i>	Berhasil Menampilkan <i>Splash Screen</i>	Sesuai
3.	Pengujian tombol pada pilihan jawaban	Menguji semua tombol	Berhasil menyimpan jawaban yang dipilih	Sesuai
4.	Pengujian tombol Tutorial	Menampilkan isi dari tombol tutorial	Berhasil menampilkan halaman dari tombol tutorial	Sesuai
5.	Pengujian tombol pindai selanjutnya	Memindai lembar jawaban selanjutnya	Berhasil untuk memindai Lembar jawaban selanjutnya	Sesuai

Berdasarkan hasil pengujian fungsionalitas tombol pada aplikasi, maka diperoleh hasil seperti pada tabel 4.2. Maka dapat disimpulkan bahwa

aplikasi DIY scan ini dapat digunakan dan fiturnya berfungsi sebagaimana mestinya.

4.2 Pengujian Jenis Pensil yang Digunakan

Pada tahapan ini dilakukan pengujian beberapa jenis alat tulis yang digunakan dalam mengisi lembar jawaban. Terdapat 3 jenis tingkat kekerasan pensil yang digunakan, yaitu : Pensil 2B, HB, dan 3H. Pengisian lembar jawaban dilakukan dengan menggunakan pola seperti yang terlampir pada Lampiran B, Lampiran C dan Lampiran D. Untuk pengujian ini peletakan handphone dibantu dengan alat *tripod*, dan ditetapkan sudut kamera sebesar 120°, dengan jarak kamera dan kertas sejauh 17cm, dan intensitas cahaya sebesar 39 lux. Presentase keberhasilan tiap pengujian dihitung dengan menggunakan persamaan berikut:

$$\% \text{keberhasilan} = \frac{\text{Total Pengujian dengan jawaban yang sesuai}}{\text{Total banyak pengujian}} \times 100\% \quad (1)$$

Tabel 4.3 Hasil pengujian pensil 2B

Jenis Pensil	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
2B	10 Pengujian	A	100%
		B	100%
		C	100%
		D	100%
		E	100%
Rata- Rata keberhasilan			100%
Tingkat Error			0%

Tabel 4.4 Hasil pengujian pensil HB

Jenis Pensil	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
HB	10 Pengujian	A	90%
		B	30%
		C	20%
		D	30%
		E	20%
Rata- Rata keberhasilan			38%
Tingkat Error			62%

Tabel 4.5 Hasil pengujian pensil 3H

Jenis Pensil	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
3H	10 Pengujian	A	0%
		B	0%
		C	10%
		D	0%
		E	10%
Rata- Rata keberhasilan			4%
Tingkat Error			96%

Berdasarkan hasil pengujian yang dilakukan pada tabel 4.3, 4.4, dan 4.5 dapat disimpulkan bahwa, tingkat akurasi deteksi penggunaan pensil jenis 2B memiliki rata-rata keberhasilan 100%, dengan tingkat kegagalan sebesar 0%. Dan tingkat keberhasilan deteksi dari pensil jenis HB memiliki rata-rata keberhasilan 38%, sedangkan untuk jenis pensil 3H memiliki tingkat keberhasilan 4%. Berdasarkan hasil analisis dari data pengujian ini, dapat disimpulkan bahwa jenis pensil dengan tingkat kegelapan yang pekat, dapat dengan mudah dideteksi oleh sistem aplikasi. Adapun faktor yang menyebabkan jenis pensil 3H dan HB memiliki tingkat error yang tinggi disebabkan pada nilai dari *Core.countnonZero* yang telah ditetapkan, sehingga sistem tidak dapat mendeteksi pixel hitam dengan baik.

4.3 Pengujian Akurasi

Pengujian akurasi ini dilakukan pada masing-masing opsi jawaban yang ada, dengan tujuan untuk mengetahui seberapa akurat sistem aplikasi dalam mendeteksi tiap opsi jawaban. Pengujian ini dilakukan dengan cara mengisi lembar jawaban dengan opsi A,B,C,D, dan E dengan pola seperti yang terlampir di Lampiran E, Lampiran F, dan Lampiran G. Jenis pensil yang digunakan dengan jenis 2B. Jarak dan sudut kamera menyesuaikan, dan intensitas cahaya sebesar 52 Lux.

Tabel 4.6 Pengujian Tingkat Akurasi

Opsi Jawaban	Jumlah Pengujian	Tingkat Keberhasilan
A	10 Pengujian	100%
B	10 Pengujian	100%
C	10 Pengujian	100%
D	10 Pengujian	100%
E	10 Pengujian	100%
Rata-rata keberhasilan		100%
Tingkat Error		0%

Hasil dari pengujian tingkat akurasi dapat dilihat pada tabel 4.6, dimana angka tingkat rata-rata keberhasilan pengujian mendapat angka 100%, faktor ini disebabkan oleh baiknya pencahayaan yang menerangi objek, sehingga kamera dapat dengan mudah mengakuisisi objek dengan baik.

4.4 Pengujian Cahaya

Pengujian cahaya ini bertujuan untuk mengetahui pengaruh sistem aplikasi terhadap intensitas cahaya yang ada disekitarnya saat aplikasi ini digunakan. Pengujian ini dilakukan dalam 3 kondisi ruangan yaitu: kondisi ruangan dengan lampu menyala disiang hari dengan intensitas cahaya sebesar 40 LUX, kondisi ruangan menggunakan lampu belajar dengan intensitas cahaya sebesar 50 LUX, pengujian ini juga menggunakan jenis pensil 2B, . Untuk pengukuran ini dilakukan pengujian terhadap lembar jawaban yang diisi dengan pola acak, seperti yang terlampir di Lampiran H, penetapan sudut kamera sebesar

120°, dengan jarak kamera dan objek sejauh 17 cm. Presentase keberhasilan pengujian dihitung dengan menggunakan persamaan berikut:

$$\% \text{keberhasilan} = \frac{\text{Total Pengujian tiap opsi}}{\text{Total banyak pengujian}} \times 100\% \quad (2)$$

Hasil pengujian akurasi deteksi tiap pilihan jawaban dapat dilihat pada tabel 4.3 dan lampiran D

Tabel 4.7 Pengujian pengaruh cahaya intensitas 39 Lux

Intensitas cahaya (LUX)	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
40	10 Pengujian	A	100%
		B	90%
		C	100%
		D	100%
		E	90%
Rata- Rata keberhasilan			96%
Tingkat Error			0%

Tabel 4.8 Pengujian pengaruh Cahaya intensitas 52 Lux

Intensitas cahaya (LUX)	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
52	10 Pengujian	A	100%
		B	100%
		C	100%
		D	100%
		E	100%
Rata- Rata keberhasilan			100%
Tingkat Error			0%

Berdasarkan hasil pengujian yang terdapat pada tabel 4.7, diketahui bahwa terdapat error dengan tingkat error sebesar 4%, dan tingkat keberhasilan 96% pada opsi jawaban B dan E, hal ini disebabkan karena kurangnya pencahayaan, serta adanya refleksi pada permukaan opsi jawaban yang ada sehingga sistem mengkategorikan opsi jawaban yang terkena refleksi tersebut sebagai jawaban yang tingkat kehitamannya kurang. Sedangkan pada hasil pengujian yang terdapat pada tabel 4.8 memiliki tingkat rata-rata keberhasilan deteksi sebesar 100%, dengan tidak error satupun hal ini disebabkan karena bagusnya pencahayaan yang menerangi objek, sehingga tidak terjadi refleksi pada permukaan opsi jawaban yang dihitamkan.

4.5 Pengujian Jarak

Pengujian pengaruh jarak ini dilakukan dengan tujuan, untuk mengetahui kinerja sistem dalam mendeteksi lembar jawaban yang dibuat. Dalam proses pengujian ini, terdapat 11 kondisi jarak,

dengan rentang jarak minimum 17 cm, dan jarak maksimal 27 cm. Pemilihan jarak maksimum dan minimum ini disesuaikan dengan kecocokan objek dan sistem aplikasi secara *default*. Pengujian dibantu dengan menggunakan alat bantu *tripod*, dengan sudut kamera terhadap objek yang akan dideteksi sebesar 120°, dengan bantuan intensitas cahaya sebesar 52 lux. Parameter pengujian ini dilakukan dengan mengisi lembar jawaban masing-masing dengan pola pengisian berbeda, seperti yang terlampir pada Lampiran F. Presentasi keberhasilan dihitung dengan menggunakan persamaan berikut:

$$\% \text{keberhasilan} = \frac{\text{Total Pengujian tiap pola jawaban}}{\text{Total banyak pengujian}} \times 100\%$$

Tabel 4.9 Pengujian Jarak

Jarak	Jumlah Pengujian	Opsi jawaban	Tingkat Keberhasilan
17 cm	10 Pengujian	A	100%
		B	100%
		C	100%
		D	100%
		E	100%
Rata- Rata keberhasilan			100%
Tingkat Error			0%

Berdasarkan hasil pengujian yang dilakukan pada tabel 4.7 dapat disimpulkan bahwa jarak 22 cm hingga 27 cm dalam pembacaan kertas lembar jawaban sudah tidak dapat mendeteksi lembar jawaban. Tidak dapatnya sistem mendeteksi objek pada jarak yang jauh disebabkan karena tidak kompatibelnya jarak dengan sistem yang telah dirancang. Hal lain juga disebabkan karena ukuran batas dari pinggir lembar jawaban sudah tidak dapat dideteksi oleh sistem dikarenakan oleh jarak yang jauh.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dan saran dari pengerjaan Proyek Akhir ini terlampir sebagai berikut:

1. Keberhasilan sistem dalam mendeteksi jawaban pada lembar jawaban sangat berpengaruh terhadap jenis pensil yang digunakan, untuk aplikasi ini diwajibkan untuk menggunakan pensil dengan jenis 2B.
2. Pencahayaan yang baik pada saat melakukan proses pemindaian sangat berpengaruh pada sistem aplikasi dalam mendeteksi tiap opsi jawaban pada lembar jawaban.
3. Aplikasi DIY Scan ini dapat dijalankan pada sistem operasi smartphone Android dengan minimal versi *Lollipop 5.0+*.
4. Aplikasi DIY Scan dapat dengan mudah memindai lembar jawaban dengan rentang

jarak dari 17 cm, sampai dengan maksimal 22 cm, dengan sudut pengambilan gambar optimal sebesar 120°.

