

APLIKASI PENDETEKSI UKURAN KEMEJA MENGGUNAKAN OPENCV PADA SMARTPHONE

SHIRT SIZE DETECTION APPLICATION USING OPENCV ON SMARTPHONES

Rendra Anggara Yudha ¹, Yuli Sun Hariyani, S.T.,M.T ², Dadan Nur Ramadan, S.Pd.,M.T ³

^{1,2,3}Prodi D3 Teknik Telekomunikasi, Universitas Telkom

¹rendra.anggara45@gmail.com, ²yulisun@tass.telkomuniversity.ac.id, ³dadan.nr@gamil.com

Abstrak

Dalam proses pembuatan pakaian setiap perusahaan harus memiliki pedoman kualitas untuk memastikan bahwa produk yang akan dikirimkan ke pelanggan adalah bebas dari cacat dan dapat diterima sesuai dengan persyaratan kualitas yang ditentukan. Salah satu prosedur yang harus diperhatikan yaitu ukuran pakaian apakah telah memenuhi persyaratan sesuai dengan prosedur yang telah ditetapkan. Untuk menentukan ukuran pakaian yang sesuai dengan prosedur pengendalian kualitas ukuran pakaian dibutuhkan alat pengukur panjang dan lebar kemeja yang praktis dan tidak membutuhkan waktu yang lama dalam menentukan ukuran kemeja.

Untuk mengatasi masalah diatas, dibuatlah aplikasi untuk mengukur ukuran kemeja menggunakan *OpenCV* pada *smartphone* untuk menangkap gambar kemeja yang diamati. Pembuatan program aplikasi ini menggunakan software *Android Studio* kemudian akan diimplementasikan pada *smartphone* dan nilai ukuran hasil pendeteksian kemeja dapat ditampilkan pada *smartphone*.

Dari hasil pengujian yang telah dilakukan menunjukkan bahwa akurasi sistem dalam mengukur panjang dan lebar kemeja dengan ukuran yang berbeda mulai dari S, M, L, XL dan XXL mendapatkan prosentase keakuratan sebesar 99.4% dengan menggunakan parameter jarak 2 Meter.

Kata Kunci : Kemeja, OpenCV, Smartphone

Abstract

In the process of making clothes every company should have quality guidelines to ensure that the product will be delivered to the customer is free from defects and acceptable in accordance with the requirements of the quality control. One of the procedures that must be observed, namely the size of the clothes do have meet the requirements in accordance with the established procedure. To determine the size of the clothes that are in accordance with the quality control procedures required clothing size gauges the length and width of the shirt that practical and doesn't need a long time to determine the size of the shirt.

To solve the problem above, an application was made to measure shirt size using OpenCV on a smartphones to capture images of the shirt is observed. The making of this application program using the Android Studio software will then be implemented on smartphones and the value of the size of the results of detection of the shirt can be displayed on the smartphones.

From the results of testing that has been done shows that the accuracy of the system in measuring the length and width of the shirts of different sizes ranging from S, M, L, XL and XXL get a percentage of 99.4% accuracy by using the parameters distance 2 meters .

Keywords : Shirts, OpenCV, Smartphone

1. Pendahuluan

Saat ini teknologi dan ilmu pengetahuan semakin berkembang diikuti dengan perkembangan mode, membuat kebutuhan manusia juga turut meningkat terutama kebutuhan pakaian. Pakaian merupakan salah satu kebutuhan yang sangat penting bagi setiap manusia dan termasuk ke dalam kebutuhan sandang untuk. Kebutuhan pakaian yang semakin berkembang menuntut perusahaan produsen pakaian dapat bersaing di pasar nasional dan internasional. Salah satu faktor yang sangat penting dalam persaingan pasar adalah kualitas suatu produk maupun layanan. Kualitas sering dijadikan sebagai suatu tolak ukur dan pembeda untuk suatu produk dan layanan antara satu produsen dengan produsen lainnya. Oleh karena itu, semua produsen pakaian dan penyedia layanan pakaian selalu mencari cara untuk meningkatkan mutu produk ataupun kualitas layanannya. Untuk menjaga dan meningkatkan mutu, perusahaan manufaktur pakaian umumnya akan menggunakan teknik pengendalian mutu atau *Quality Control*. Tujuan utama pengendalian mutu yaitu memastikan bahwa produk yang akan dikirimkan ke pelanggan adalah bebas dari cacat, mengendalikan, menyeleksi, menilai kualitas, sehingga konsumen merasa puas dan perusahaan tidak rugi serta produk dapat diterima sesuai dengan persyaratan mutu yang ditentukan. Jika ditemukan produk yang cacat maka diperlukan tindakan perbaikan yang sesuai. Dalam pengendalian mutu ukuran kemeja pada proses *Quality Control* belum adanya alat yang dapat menentukan ukuran kemeja sesuai dengan prosedur pembuatan. Hal tersebut membuat perusahaan produsen pakaian memakan waktu yang cukup lama untuk mengetahui keakuratan ukuran pakaian yang diproduksi. Berdasarkan dari penelitian sebelumnya yang dilakukan oleh Muhammad Soleh Gangsarestu tentang “Rancang Bangun Sistem Pengukuran Badan Pria Untuk Menentukan Ukuran Baju Berbasis Kamera Kinect” [6], penulis ingin mengembangkan sistem pengukuran baju yang sebelumnya berbasis kamera *Kinect* dan menggunakan komputer untuk menampilkan nilai keluaran dengan sistem pendeteksian ukuran kemeja menggunakan *OpenCV* dan dapat memunculkan nilai keluaran ukuran baju sesuai dengan objek yang diamati pada *smartphone*.

Untuk mempermudah perusahaan produsen pakaian dalam menentukan ukuran kemeja maka dibuatlah sebuah aplikasi pada *smartphone* untuk dapat mendeteksi ukuran kemeja menggunakan *OpenCV* dengan mengambil gambar objek kemeja yang kemudian dilakukan proses pemotongan gambar sehingga menjadi lebih kecil oleh aplikasi yang sudah dibuat dan dapat dianalisa ukuran pakaian yang tepat sesuai dengan kemeja yang diamati.

Pada Proyek Akhir ini, penulis menggunakan metode *image cropping* dengan *OpenCV* dan kamera belakang *smartphone* untuk menampilkan nilai hasil gambar yang diamati. Sistem analisa gambar dimulai dari tahap pengambilan gambar, memotong ukuran objek kemeja sesuai dengan ukuran sebenarnya dan dapat menghasilkan keluaran indikasi ukuran kemeja. Semua informasi akan ditampilkan di layar *smartphone* yang berupa gambar objek kemeja, daftar ukuran kemeja dan nilai keluaran hasil pendeteksian kemeja. Proyek Akhir ini dapat membantu kinerja perusahaan produsen pakaian dalam proses *Quality Control* agar lebih efisien dan dapat menentukan ukuran kemeja yang tepat tanpa perlu waktu lama sesuai dengan ukuran kemeja yang diamati.

2. Dasar Teori

2.1 Pakaian

Pakaian merupakan salah satu kebutuhan pokok manusia selain makanan dan tempat tinggal atau rumah. Manusia membutuhkan pakaian untuk melindungi dan menutupi tubuh dirinya dari manusia lain dan dari pengaruh cuaca. Namun, seiring dengan berkembangnya kehidupan manusia, pakaian juga digunakan sebagai simbol status, maupun kedudukan dari individu yang mengenakannya [7].

Salah satu tujuan utama dari pakaian yaitu untuk menjaga tubuh pemakainya dari pengaruh lingkungan sekitar agar pemakainya merasa nyaman dan aman. Dalam cuaca panas pakaian juga menjadi pelindung agar kulit tidak terbakar oleh sinar matahari, sedangkan di cuaca dingin pakaian berfungsi untuk menghangatkan tubuh manusia. Pakaian juga dapat melindungi bagian tubuh yang tidak terlihat dari pandangan manusia lainnya. Pakaian berfungsi sebagai pelindung dari faktor-faktor yang dapat merusak tubuh, termasuk hujan, salju, matahari, angin, maupun kondisi cuaca lainnya. Pakaian juga berguna dalam mengurangi tingkat risiko dalam aktivitas, seperti bekerja, maupun berolahraga. Pakaian juga digunakan untuk perlindungan diri dari bahaya lingkungan tertentu seperti hewan, bahan kimia berbahaya, senjata, dan sebagainya.

2.2 Citra

Citra (*image*) merupakan gambar pada penampang bidang dua dimensi. Citra dapat diartikan sebagai sebuah fungsi intensitas cahaya. Intensitas cahaya sebuah citra didapat dari sumber cahaya yang menerangi objeknya yang kemudian objek dapat memantulkan kembali sebagian dari cahaya tersebut. Pantulan-pantulan cahaya dari citra tersebut dapat ditangkap oleh alat-alat optik seperti mata pada manusia, kamera, pemindai (*scanner*) dan sebagainya, sehingga objek *image* tersebut dapat terekam menjadi sebuah gambar. Citra adalah sebagai keluaran dari suatu sistem perekaman data yang bersifat :

1. Optik dalam bentuk foto
2. Analog dalam bentuk sinyal video yang meliputi gambar pada monitor televisi
3. Digital yang dapat langsung disimpan pada suatu pita magnetik.

Jika diamati dari sudut pandang matematisnya, citra adalah fungsi nalar (*continue*) dari intensitas cahaya pada penampang bidang dwimatra. Citra itu sendiri dapat digolongkan menjadi dua jenis yaitu :

1. Citra Foto
2. Citra foto dalam bentuk lembaran-lembaran foto.
3. Citra Non Foto

Citra nonfoto merupakan citra yang diolah dengan menggunakan sensor elektronik. Sistem pengolahannya diawali dari spektrum elektromagnetik yang diterima oleh sensor, kemudian direkam pada pita magnetik. Bentuk pita ini meliputi pita video tape. Untuk prosedur perekamannya adalah dengan menggunakan sistem scanning sehingga sensor ini dapat disebut scanner. Sinyal elektronik yang terekam dapat ditampilkan pada layar monitor. Dari layar monitor, citra dapat diolah menjadi foto atau bentuk objek citra yang lainnya. Sedangkan Citra yang tidak bergerak disebut dengan citra diam [5].



Gambar 2.1 Citra Diam

Citra bergerak (*moving images*) adalah kumpulan dari citra diam yang disajikan dengan cara berurutan sehingga terlihat seperti gambar yang sedang bergerak. Citra diam tersebut disebut dengan *frame*. *Frame* akan bergerak sesuai dengan kecepatan fps (*frame per second*). Objek-objek gambar yang tampak pada film layar lebar ataupun televisi pada dasarnya terdiri dari ratusan sampai ribuan *frame*.

2.3 Pengolahan Citra

Citra atau gambar tidak selalu pada kondisi yang sempurna. Terkadang gambar yang diamati mengalami penurunan kualitas (*degradasi*), seperti mengandung cacat atau derau (*noise*), warna yang terlalu kontras, kurang tajam bahkan kabur (*blurring*), dan sebagainya. Penurunan kualitas ini dapat mempersulit untuk mengolah informasi yang terdapat pada gambar tersebut untuk diolah.

Agar gambar yang mengalami penurunan kualitas dapat diolah, maka diperlukan proses manipulasi agar gambar menjadi lebih baik kualitasnya. Metode memanipulasi gambar adalah sebuah bidang yang disebut dengan pengolahan citra (*image processing*) [4].

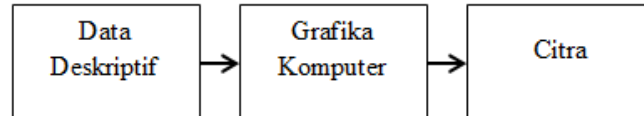
Pengolahan citra merupakan pemrosesan citra, khususnya dengan menggunakan komputer kemudian diproses sehingga menjadi gambar yang memiliki kualitas yang lebih baik. Pada dasarnya, sistem operasi pada proses pengolahan citra diterapkan pada citra jika :

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan mutu penampakan objeknya atau untuk menunjukkan beberapa aspek informasi yang terdapat pada gambar tersebut
1. Elemen-elemen pada gambar perlu untuk dikelompokkan, dicocokkan maupun diukur kembali
2. Sebagian bentuk gambar perlu digabungkan dengan bagian gambar yang lain

Di dalam bidang komputer, terdapat 3 jenis bidang studi yang berkaitan dengan citra, yaitu :

1. Grafika Komputer (*computer graphics*).

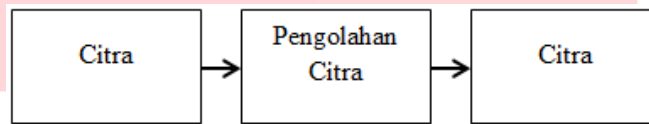
Grafika Komputer bertujuan untuk menghasilkan gambar atau citra (*grafik* atau *picture*) dengan fungsi-fungsi geometri seperti garis, lingkaran, dan lainnya. Fungsi-fungsi geometri tersebut memerlukan data yang mendeskripsikan sebuah citra yang meliputi koordinat titik, panjang garis, warna dan lainnya. Grafika komputer memiliki peran penting dalam visualisasi dan *virtual reality* suatu objek citra [5].



Gambar 2.2 Blok Grafika Komputer [4]

2. Pengolahan Citra (*image processing*)

Pengolahan Citra mempunyai bertujuan untuk memperbaiki kualitas citra agar lebih mudah diolah informasinya oleh manusia ataupun mesin (dalam hal ini komputer). Dalam pengolahan citra, input dari proses merupakan citra dan output juga merupakan citra, namun citra outputan memiliki kualitas yang lebih baik dibandingkan dengan citra inputannya. Pemampatan citra (*image compression*) juga termasuk dalam kategori pada bidang ini.

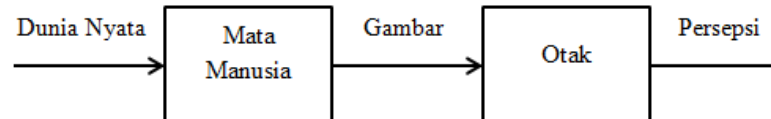


Gambar 2.3 Blok Pengolahan Citra [11]

2.4 Computer Vision

Computer vision adalah sebuah disiplin ilmu yang mempelajari proses menyusun deskripsi tentang objek yang terkandung pada suatu gambar atau mengenali objek yang ada pada gambar. *Computer vision* berusaha menerjemahkan citra menjadi deskripsi atau suatu informasi yang merepresentasikan citra tersebut [10].

Pada hakikatnya, *Computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*). *Computer vision* merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual seperti akuisisi citra, pengolahan citra, klasifikasi, pengenalan (*recognition*), dan membuat keputusan.

Gambar 2.4 Ilustrasi *Computer Vision* [8]

Computer vision terdiri dari teknik-teknik untuk mengestimasi ciri-ciri objek di dalam citra, pengukuran ciri yang berkaitan dengan geometri objek, dan menginterpretasi informasi geometri tersebut. Mungkin berguna bagi anda untuk mengingat persamaan berikut :

$$\text{Vision} = \text{Geometry} + \text{Measurement} + \text{Interpretation}$$

Proses-proses di dalam *computer vision* dapat dibagi menjadi tiga aktivitas :

1. Memperoleh atau mengakuisisi citra digital.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra (operasi-operasi pengolahan citra).

3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur dan lain-lain.

Dari penjelasan di atas, dapat dilihat bahwa pengolahan citra dan pengenalan pola merupakan bagian dari *computer vision*. Pengolahan citra merupakan proses awal (*preprocessing*) pada *computer vision*. Sedangkan pengenalan pola merupakan proses untuk menginterpretasi citra. Teknik-teknik di dalam pengenalan pola berperan penting dalam *computer vision* untuk mengenali suatu objek.

2.5 Android

Android merupakan sistem operasi yang berbasis Linux untuk telepon seluler yang meliputi telepon pintar dan komputer tablet [1]. Android menawarkan platform *Open Source* untuk membuat aplikasi yang dapat digunakan oleh bermacam perangkat bergerak.

Fitur-fitur yang terdapat pada Android yaitu :

1. *Framework* aplikasi, yang memungkinkan daur ulang dan penggantian komponen.
2. Browser terintegrasi berbasis *engine Open Source WebKit* yang juga digunakan di *browser* iPhone dan Nokia S60v3.
3. Rancangan handset, *Platform* disesuaikan dengan kebutuhan *Video Graphics Adapter (VGA)* yang lebih besar, library grafik 2D dan 3D yang berdasarkan pada spesifikasi OpenGL ES 1.0 serta *layout smartphone* yang tradisional.
4. *Multi-touch*, Android memiliki dukungan bawaan untuk multi-touch yang tersedia pada handset terbaru seperti HTC Hero.
5. Dukungan hardware tambahan. Android yang mendukung penggunaan kamera, layar sentuh, *Global Positioning System (GPS)*, pengukur kecepatan, magnetometer, akselerasi 2D bit blits (dengan orientasi *hardware, scaling*, konversi format *pixel*) dan akselerasi grafis 3D [3].

2.6 Quality Control

Quality Control adalah sebuah proses yang memastikan apakah produk dari pengolahan tersebut layak dilayak untuk dibuat dan sesuai dengan prosedur pengendalian mutu. Umumnya *Quality Control* ini digunakan oleh suatu perusahaan atau pabrik manufaktur yang bergerak dibidang seperti tekstil maupun perusahaan manufaktur kulit pada pembuatan tass dan sepatu. Pada tiap perusahaan yang menerapkan *Quality Control (QC)*.

Pada dasarnya, *Quality Control* ini berkenaan dengan kegiatan operasional dan teknik yang digunakan untuk memenuhi persyaratan mutu dari produk yang diolah tersebut. Tujuan dari *Quality Control* atau pengendalian mutu ini yaitu untuk memastikan bahwa produk yang akan dikirimkan ke konsumen adalah bebas dari cacat dan dapat diterima konsumen sesuai dengan persyaratan mutu yang sudah ditentukan.

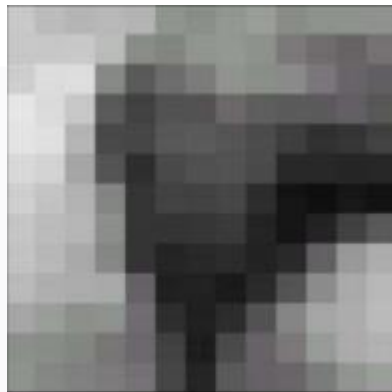
2.7 Image Cropping

Cropping merupakan proses pemotongan citra atau gambarr pada titik koordinat tertentu di penampakan area gambar tersebut. Dalam proses pemotongan satu bagian dari gambar sehingga didapat gambar yang berukuran

lebih kecil. Proses ini mempunyai tujuan untuk memisahkan objek yang satu dengan objek yang lainnya dalam sebuah gambar untuk mempercepat tahap selanjutnya. Untuk memotong bagian dari citra digunakan dua titik koordinat, yaitu titik koordinat awal yang merupakan awal koordinat bagi gambar hasil pemotongan dan titik koordinat akhir yang merupakan titik koordinat akhir dari gambar hasil pemotongan. Sehingga terbentuk penampakan objek segi empat yang memiliki kumpulan titik-titik *pixel* yang terdapat pada area koordinat tertentu kemudian akan disimpan dalam citra atau gambar yang baru.

2.8 Pixel

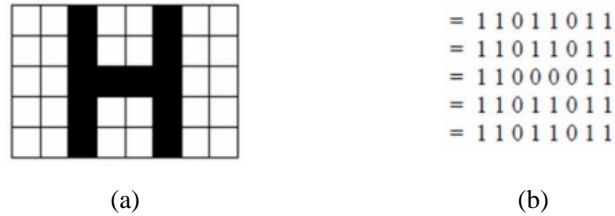
Pixel berasal dari akronim bahasa Inggris *Picture Element* yang disingkat menjadi *pixel*. *Pixel* merupakan unsur dari suatu gambar atau representasi dari sebuah titik-titik terkecil didalam sebuah gambar grafis yang dihitung per inchi. *Pixel* juga disebut titik-titik cahaya yang membentuk sebuah objek. Semakin banyak jumlah *pixel* dalam sebuah gambar, maka semakin besar juga resolusi spasial gambar tersebut sehingga gambar terlihat semakin tajam. Setiap *pixel* mewakili tidak hanya satu titik dalam sebuah gambar melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil. Seperti yang ditunjukkan pada Gambar 2.5



Gambar 2.5 *Pixel* berbentuk kotak-kotak kecil [10]

2.9 Citra Biner

Citra biner merupakan citra digital yang hanya mempunyai dua kemungkinan pada nilai *pixel* yaitu hitam dan putih. Citra biner juga dapat dikatakan sebagai citra black & white (BW) atau bisa disebut juga dengan citra monokrom. Di dalam citra biner setiap titik (*pixel*) bernilai 0 atau 1, masing-masing mempresentasikan warna tertentu. Contoh yang paling umum digunakan adalah warna hitam bernilai 0 dan warna putih bernilai 1. Setiap titik *pixel* pada citra hanya membutuhkan media penyimpanan 1 bit saja, sehingga setiap byte dapat menampung informasi 8 titik *pixel*. Gambar 2.6 menunjukkan contoh citra biner dan representasinya dalam data digital.



Gambar 2.6 Citra biner data digital dan penyimpanan pada memori [10]

Citra biner juga sering kali dimunculkan sebagai hasil dari proses pengolahan yang meliputi segmentasi, pengembangan, maupun morfologi pada gambar atau citra.



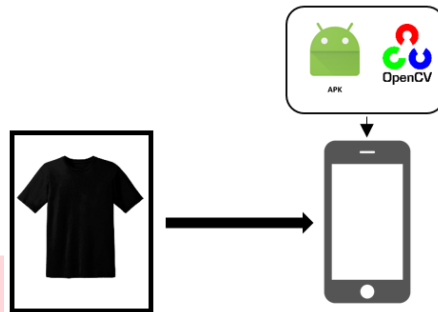
Gambar 2.7 Citra Biner [15]

2.10 Bitmap

Bitmap merupakan sebuah format gambar asli yang tidak mengalami proses kompresi gambar. Ukuran memori gambar dengan format ini sangat besar dan dapat memperoleh 24 bit kedalaman warna. Sebab gambar berformat BMP belum mengalami proses kompresi sehingga program aplikasi pengolahan citra yang dirancang ini menggunakan gambar input berformat BMP. Bitmap (BMP) yaitu suatu format gambar yang paling umum dan merupakan format gambar standar windows. Kelebihan dari tipe Bitmap ini adalah dapat dibuka hampir di semua program pengolah gambar, selain itu gambar yang disimpan dengan tipe data BMP tidak akan mengalami penurunan mutu gambar. Gambar dalam format BMP umumnya tidak dikompres sehingga tidak ada informasi yang hilang. File ini adalah format yang belum terkompresi dan menggunakan sistem warna Red, Green Blue (RGB) yang masing-masing warna pixelnya terdiri atas 3 komponen utama yang digabung menjadi satu. Bitmap merupakan suatu representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan pada memori komputer. Dikembangkan oleh Microsoft dan nilai setiap titik diawali dengan satu bit data untuk gambar hitam putih atau dapat lebih lagi untuk gambar yang mempunyai warna. Kerapatan titik-titik tersebut disebut resolusi yang menyatakan seberapa tajam gambar yang ditampilkan tersebut diperoleh dari jumlah baris dan kolom. Gambar dalam format BMP terdiri dari 3 macam yaitu citra biner, citra berwarna dan citra hitam-putih (*grayscale*). Citra biner hanya memiliki dua nilai keabuan 0 dan 1. Oleh sebab itu, 1 bit sudah cukup untuk merepresentasikan nilai *pixel*. Citra berwarna merupakan citra yang lebih umum. 17 Warna yang terlihat pada citra bitmap adalah kombinasi dari tiga warna dasar yaitu merah, hijau dan biru. Setiap pixelnya disusun dari 3 komponen warna yakni Red, Green dan Blue (RGB). Kombinasi dari 3 warna RGB tersebut dapat menghasilkan warna yang khas terhadap *pixel* yang diperoleh dari sebuah gambar atau citra tersebut.

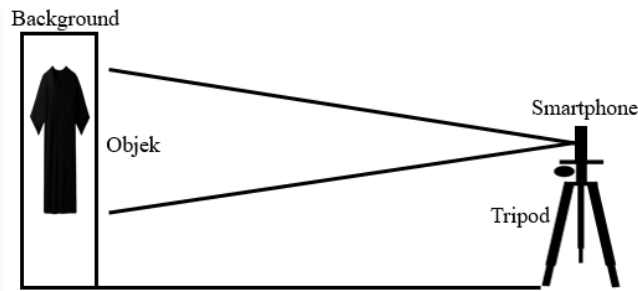
3. Perancangan

3.1 Pemodelan Sistem Aplikasi Pendeteksi Ukuran Kemeja



Gambar 3.1 Sistem Aplikasi Pendeteksi Ukuran Kemeja Pada *Smartphone*

Pemodelan sistem pada proyek akhir ini dapat dilihat pada Gambar 3.1. *Smartphone* dipasang pada tripod dengan jarak tertentu agar gambar yang diambil dari kamera *smartphone* nilai pikselnya tidak berubah dan mudah untuk dideteksi. Setelah itu, akan dilakukan proses pemotongan gambar oleh aplikasi yang sudah terinstal pada *smartphone*. Kemudian *output* atau nilai hasil perhitungan panjang dan lebar kemeja akan ditampilkan pada layar aplikasi.



Gambar 3.2 Posisi kamera *smartphone* terhadap objek

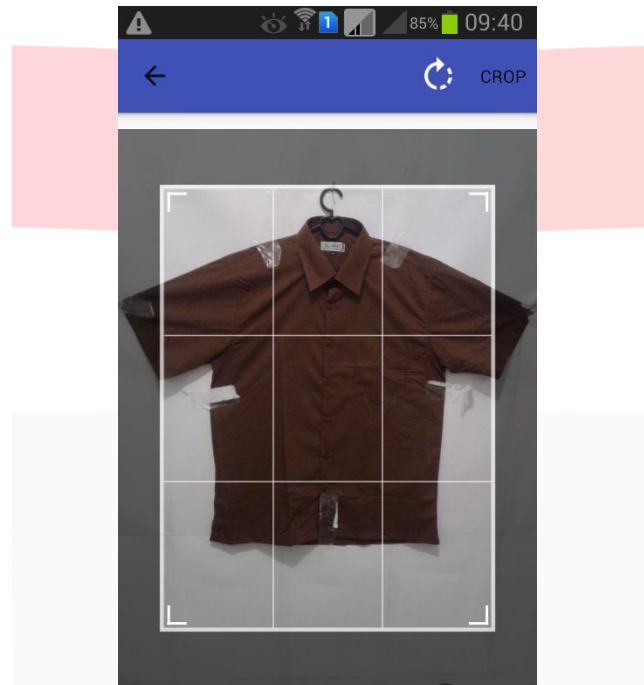
Pada Gambar 3.2 menunjukkan posisi *smartphone* terhadap objek kemeja. Posisi *smartphone* harus lurus dengan sisi kanan dan kiri dari *background* dan posisi objek juga harus berada tepat jika dilihat dari tengah layar *smartphone*. Posisi seperti ini digunakan untuk mengetahui berapa jarak ideal antara objek dengan *smartphone* dan posisi ideal objek rata tengah pada kamera *smartphone* untuk memudahkan perhitungan nilai ukuran kemeja.

3.2 Blok Diagram Sistem



Gambar 3.3 Sistem Deteksi Gambar

Gambar 3.3 merupakan diagram sistem gambar mengenai proses pengolahan gambar. Akuisisi gambar akan ditangkap oleh kamera dalam bentuk foto dengan objek kemeja. Foto kemeja tersebut kemudian akan dilakukan proses pemotongan gambar sesuai dengan ukuran kemeja pada foto dengan tujuan agar objek yang diamati lebih mudah untuk diproses selanjutnya hanya diambil gambar panjang dan lebar kemeja. Selanjutnya gambar hanya terdiri dari kumpulan *pixel-pixel* sehingga hasil dari proses deteksi *pixel* tersebut akan dikonversikan dari bentuk *pixel* ke dalam satuan Centimeter untuk menentukan berapa Centimeter panjang dan lebar kemeja yang kemudian memunculkan *output* data berupa ukuran kemeja sesuai dengan ukuran sebenarnya.

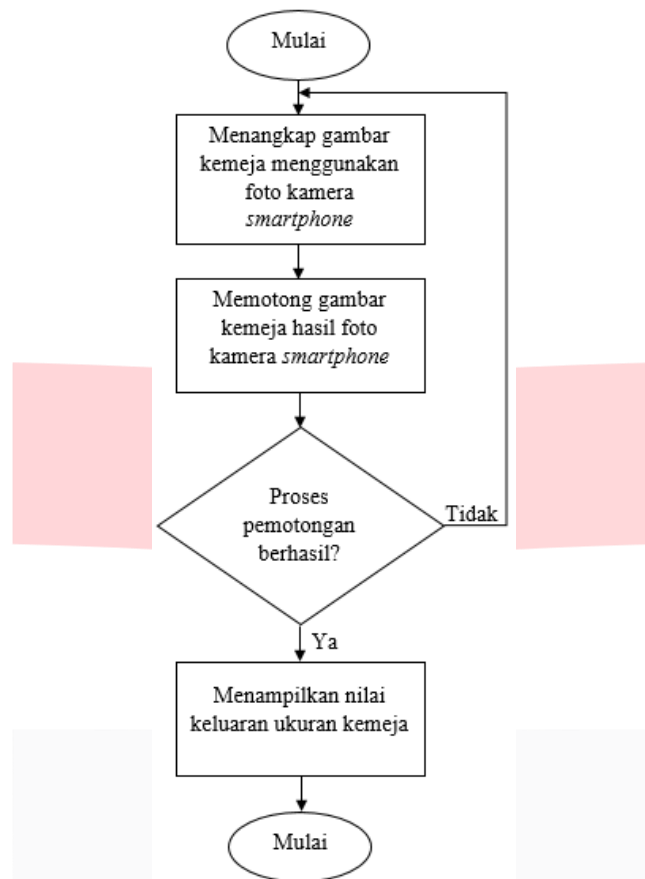


(a) Proses *Image Cropping*

Gambar 3.4 Cropping Image

Proses *image cropping* ini dilakukan karena sistem dibuat untuk mendeteksi sekumpulan *sample pixel* berdasarkan *range* yang ditentukan. Dengan melakukan proses pemotongan gambar yang dideteksi hanya *pixel* yang memenuhi layar pada aplikasi *smartphone* sesuai dengan ukuran objek. Selanjutnya gambar akan diolah oleh aplikasi untuk mendapatkan nilai ukuran panjang dan lebar kemeja menggunakan algoritma deteksi.

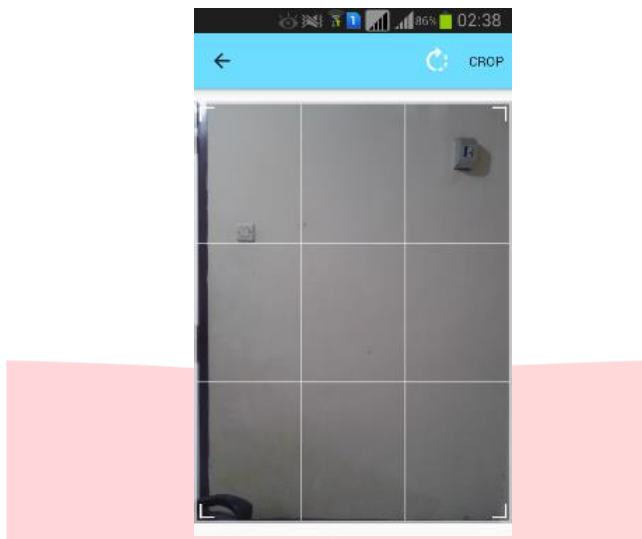
3.3 Diagram Alir Sistem Aplikasi



Gambar 3.5 Perancangan Program Aplikasi

Ketika gambar berupa objek kemeja telah diambil menggunakan foto pada kamera *smartphone* dan menghasilkan *output* berupa gambar. Gambar kemeja hasil foto tersebut akan dilakukan proses pemotongan agar dapat menghasilkan nilai resolusi *pixel* dari kemeja yang akan diukur. Kemudian setelah di potong gambar kemeja yang terdiri dari *pixel* akan dikonversikan ke dalam centimeter menggunakan algoritma pada software android studio. Setelah dikonversi maka hasil keluaran konversi dari *pixel* ke centimeter akan ditampilkan pada layar aplikasi langsung sesuai dengan ukuran objek kemeja sebenarnya.

3.4 Algoritma Deteksi



Gambar 3.6 Full screen image

Pada Gambar 3.6 adalah proses pengambilan gambar secara *full screen* sesuai dengan ukuran layar *smartphone* untuk memudahkan dalam proses pencarian banyaknya *pixel* yang dapat ditangkap oleh *smartphone* tersebut. Setelah dilakukan pengambilan gambar secara *full screen* maka diperoleh data ukuran *pixel* panjang dan lebar layar yang ditampilkan pada *Logcat software* Android Studio menggunakan perintah *bitmapRect*,

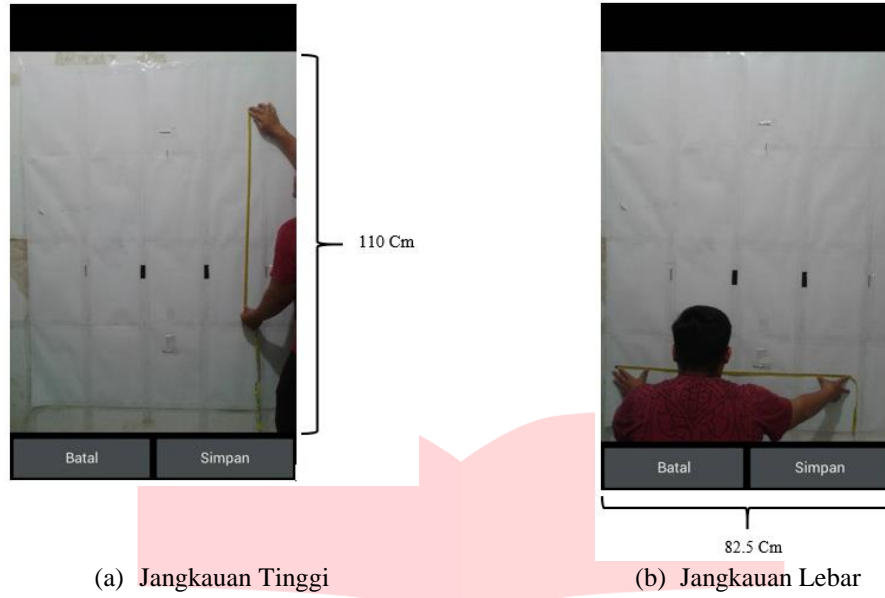
```
Log.d("tes width", "tes width: " + width);
Log.d("tes height", "tes height: " + height);
```

Hasil dari perhitungan *full screen* pada *Logcat* dilakukan dengan menghubungkan *smartphone* dan Android Studio. Untuk hasil pengukurannya dapat dilihat seperti pada Gambar 3.7 berikut ini,

```
01-04 02:53:17.930 21274-21274/com.example.test D/tes width: tes width: 640.0
01-04 02:53:17.930 21274-21274/com.example.test D/tes height: tes height: 853.0
```

Gambar 3.7 Logcat full screen

Pada Gambar 3.7 dapat diketahui bahwa ukuran *pixel* layar yang dapat ditangkap oleh aplikasi adalah memiliki panjang 853 dan lebar 640. Nilai panjang dan lebar ini nantinya akan dijadikan sebagai salah satu parameter algoritma. Setelah menemukan nilai *full screen*, selanjutnya adalah kalibrasi jangkauan kamera *smartphone* dengan tujuan mencari nilai jangkauan tinggi dan lebar sesuai dengan luas layar *smartphone*. Tahapan kalibrasi jangkauan kamera dapat dilihat dari Gambar 3.8 berikut



Gambar 3.8 Kalibrasi jangkauan kamera

Pada Gambar 3.8 dengan melakukan proses foto kemudian *background* diukur secara manual sesuai dengan jangkauan kamera dapat diperoleh nilai tinggi jangkauan kamera *smartphone* adalah 110 Cm dan lebar jangkauan kamera yaitu 82.5 Cm dengan jarak kamera *smartphone* dengan *background* adalah 2 Meter. Nilai jangkauan tinggi dan lebar kamera ini akan menjadi nilai algoritma dalam penentuan perbandingan nilai. Kemudian setelah diketahui nilai *pixel* foto kamera dan jangkauan kamera, selanjutnya memasuki tahap penentuan perbandingan *pixel* dengan centimeter untuk mengetahui berapakah nilai 1 *pixel* dari 1 centimeter pada objek sesungguhnya. Untuk menentukan nilai centimeter terhadap *pixel* maka digunakan rumus sebagai berikut :

1. Dari hasil pengukuran ukuran jangkauan kamera pada layar *smartphone* dan pengujian pada *software* Android Studio maka diperoleh nilai hasil pengujian aplikasi seperti ditunjukkan pada Tabel 3.1

Tabel 3.1 Nilai Hasil Pengujian Aplikasi

Layar Smartphone		Jangkauan Kamera Smartphone	
Tinggi	Lebar	Tinggi	Lebar
853 Pixel	640 Pixel	110 Cm	82.5 Cm

Pada Tabel 3.1 didapat nilai masing-masing ukuran tinggi dan lebar pada layar *smartphone* dan jangkauan kamera yang dapat diperoleh dari kamera *smartphone*. Dalam hal penulisan rumus algoritma secara matematis ini dianggap panjang adalah sama dengan tinggi yang menunjukkan sisi panjang pada kemeja. Setelah diketahui nilainya maka dapat menuliskan algoritma secara matematis dengan menggunakan rumus sebagai berikut :

Rumus yang digunakan untuk mencari nilai panjang dan lebar objek menggunakan rumus yang sama.

Rumus :

$$\begin{aligned}
 2. \text{ Mencari nilai Tinggi} &= \frac{(\text{Tinggi Jangkauan Kamera Smartphone})}{(\text{Tinggi Layar Smartphone})} \\
 &= \frac{110 \text{ Cm}}{853 \text{ Pixel}} \\
 &= 0.125 \text{ Cm/Pixel}
 \end{aligned}$$

$$\begin{aligned}
 3. \text{ Mencari nilai Lebar} &= \frac{(\text{Lebar Jangkauan Kamera Smartphone})}{(\text{Lebar Layar Smartphone})} \\
 &= \frac{82.5 \text{ Cm}}{640 \text{ Pixel}} \\
 &= 0.127 \text{ Cm/Pixel}
 \end{aligned}$$

Dengan hasil perhitungan rumus tersebut dapat diketahui bahwa 1 *pixel* gambar pada kamera aplikasi *smartphone* sama dengan panjang 0,125 Cm dan lebar 0.127 Cm pada objek kemeja sesungguhnya. Jika perbandingan nilai *pixel* dengan centimeter telah ditemukan, maka nilai perbandingan tersebut akan dimasukkan perintah pada software Android Studio sebagai parameter nilai yang digunakan untuk menentukan ukuran kemeja pada tahap perhitungan selanjutnya menggunakan perintah *double* untuk memasukkan nilai seperti berikut,

```
double pixelRatioLebar = 0.127;
double getPixelRatioPanjang = 0.125;
```

Parameter nilai ini diambil dari beberapa kali pengujian terhadap aplikasi, setelah dilakukan pengujian nilai yang paling akurat untuk ukuran panjang objek yang dideteksi adalah 0.125 dan lebar 0.127.

Kemudian setelah diketahui nilai perbandingannya, untuk menentukan ukuran kemeja selanjutnya nilai 0.125 dan 0.127 akan dikalikan dengan nilai panjang dan lebar objek kemeja yang telah di *crop* sesuai dengan ukuran kemeja yang ditampilkan pada aplikasi *smartphone*.

Mode penentuan nilai ukuran kemeja ini menggunakan perintah-perintah sebagai berikut :

```
RectF bitmapRect = new RectF();
bitmapRect.right = imageView.getDrawable().getIntrinsicWidth();
bitmapRect.bottom = imageView.getDrawable().getIntrinsicHeight();
```

Untuk memasukkan nilai panjang dan lebar *pixel* menggunakan perintah *bitmapRect* sehingga nilai dapat diproses pada program aplikasi. Kemudian mengambil nilai ukuran *pixel image* hasil dari *crop* di *bitmapRect* yang akan digunakan pada perhitungan rumus selanjutnya yaitu dengan melakukan perintah *double* seperti berikut,

```
double width = (double) bitmapRect.width();
double height = (double) bitmapRect.height();
```

Setelah nilai ukuran *pixel image* telah diambil, selanjutnya memasuki proses perhitungan rumus untuk menentukan ukuran kemeja berdasarkan satuan centimeter sesuai dengan ukuran kemeja sesungguhnya sebagai berikut,

```
double ukuranHeight = height * getPixelRatioPanjang;
double ukuranWidth = width * pixelRatioLebar;
```

Jika dalam perhitungan matematisnya dapat dituliskan sebagai berikut :

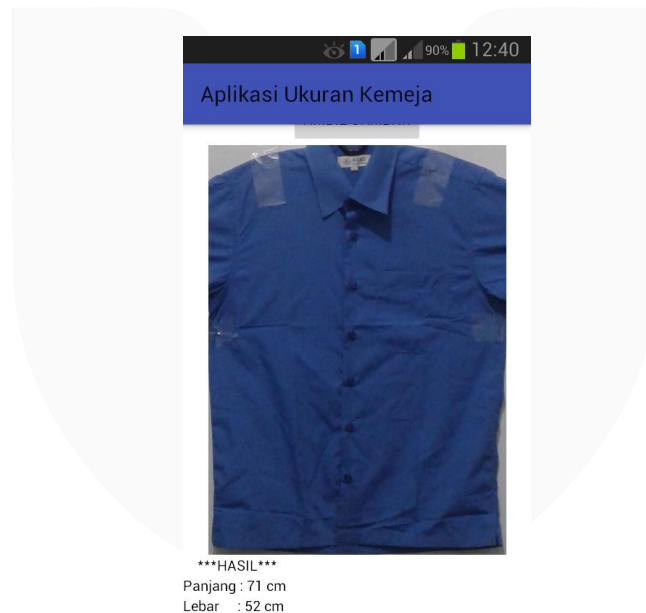
Misal, jika nilai *pixel* hasil *image cropping* diketahui panjang 560 dan lebar 409 dibagi dengan 0.125 dan 0.127,

$$\text{Panjang : } 560 \times 0.125 = 70 \text{ Cm} \qquad \text{Lebar : } 409 \times 0.127 = 52 \text{ Cm}$$

Dengan hasil perhitungan tersebut dapat dikonversikan dalam bentuk satuan centimeter untuk mengetahui ukuran objek kemeja yang diamati sesungguhnya yaitu mempunyai panjang 70 Cm dan lebar kemeja 52 Cm. Kemudian nilai hasil konversi tersebut ditampilkan pada layar aplikasi *smartphone* menggunakan perintah *TextView* dan *setText*,

```
hasil.setText ("    ***HASIL***" );
tinggi.setText ("Panjang : " + Math.round(ukuranHeight) + " cm" );
lebar.setText ("Lebar      : " + Math.round(ukuranWidth) + " cm" );
```

Setelah semua perhitungan dan konversi *pixel* ke centimeter telah dilakukan sesuai dengan percobaan pengujian aplikasi, selanjutnya melakukan *build application* program Android Studio untuk dibangun sebuah aplikasi Android pada *smartphone* yang telah ditentukan. Tampilan hasil konversi *pixel* terhadap centimeter dapat dilihat pada Gambar 3.9 berikut



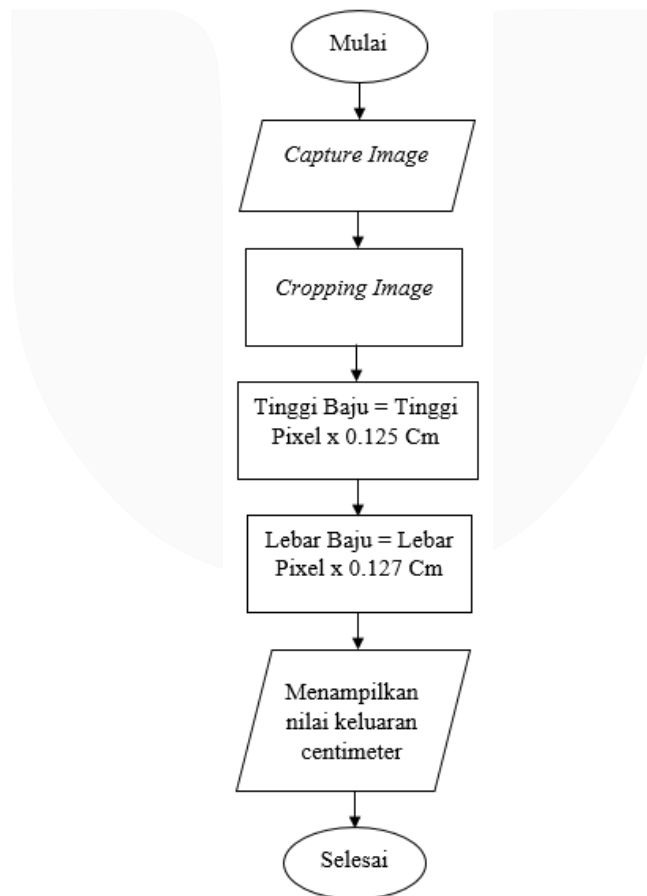
Gambar 3 .9 Tampilan nilai keluaran ukuran kemeja

Pada Gambar 3.9 menunjukkan nilai keluaran ukuran kemeja hasil dari proses *image cropping* sesuai dengan ukuran objek kemeja yang dideteksi. Untuk daftar ukuran kemeja akan ditampilkan di bawah dari nilai keluaran objek kemeja seperti terlihat pada Gambar 3.10



Gambar 3 .10 Tampilan daftar ukuran kemeja

Dari beberapa tahapan aplikasi dalam pemrosesan gambar atau objek kemeja yang diamati dari awal tahap hingga dapat menampilkan nilai keluaran ukuran kemeja berikut merupakan *flowchart* deteksi *pixel* seperti yang ditunjukkan pada Gambar 3.11



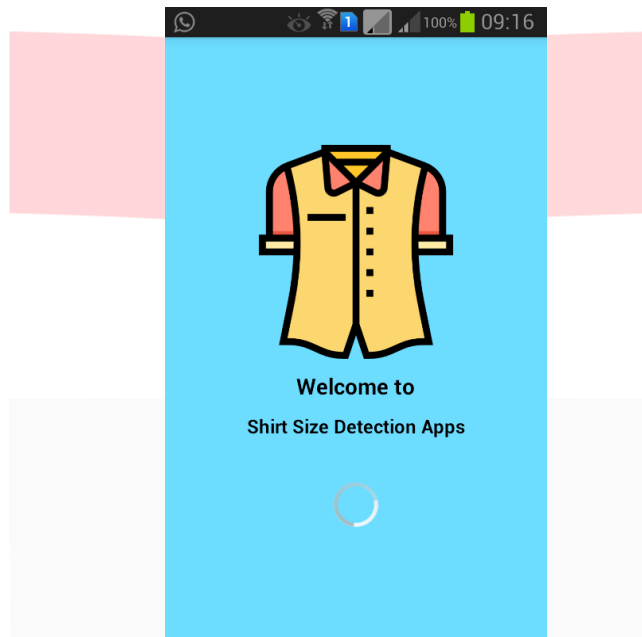
Gambar 3 .11 Flowchart Deteksi Pixel

Pada Gambar 3.11 dapat dilihat bahwa sebelum dilakukan proses deteksi *pixel* dengan cara *crop image*, dilakukan terlebih dahulu kalibrasi ukuran *pixel* jika gambar diambil dengan *full screen* pada aplikasi yang digunakan pada *smartphone* agar dapat mengestimasi ukuran *pixel* yang mampu ditangkap oleh layar. Mode kalibrasi dan deteksi yang diatur menggunakan perintah bitmap.

```
CropImage.toOvalBitmap(result.getBitmap())
```

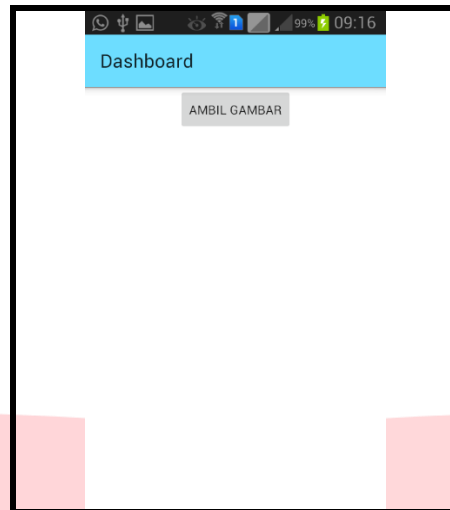
3.5 Perancangan Aplikasi

3.5.1 Mockup Aplikasi



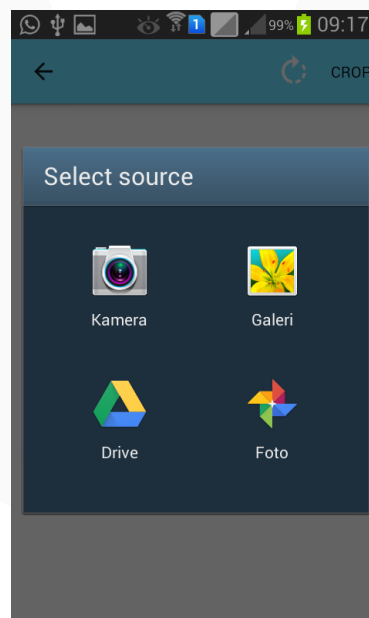
Gambar 3.12 *Splash Screen*

Pada Gambar 3.12 menunjukkan *Splash Screen* diawal pada saat aplikasi dibuka. Didalam *Splash Screen* terdapat gambar kemeja dan terdapat *ProgressBar*, jika *ProgressBar* selesai maka aplikasi akan dialihkan ke mode selanjutnya. Untuk durasi *Splash Screen* adalah pada *ProgressBar* 5 detik sebelum masuk ke mode selanjutnya. Setelah *Splash Screen* selesai maka akan langsung masuk ke mode *Dashboard* aplikasi. Mode *Dashboard* dapat dilihat pada Gambar 3.13



Gambar 3.13 *Dashboard*

Pada Gambar 3.13 menunjukkan dashboard dari aplikasi pendeteksi ukuran meja. Terdapat Button AMBIL GAMBAR untuk mulai mengambil gambar yang selanjutnya akan diproses. Kemudian tampilan akan dialihkan pada pilihan untuk metode mengambil gambarnya seperti Gambar 3.14



Gambar 3.14 *Pilihan Metode Pengambilan*

Pada Gambar 3.14 adalah pilihan metode pengambilan gambar mempunyai pilihan yaitu dapat melalui kamera secara langsung, galeri, *Google Drive* atau fitur foto. Kemudian user memilih kamera dan aplikasi akan dialihkan pada kamera belakang *smartphone* seperti pada Gambar 3.15



Gambar 3.15 Kamera Smartphone



Gambar 3.16 Hasil Pengambilan Kamera

Pada Gambar 3.15 adalah tampilan kamera aplikasi pada *smartphone* yang digunakan untuk mengambil gambar dan Gambar 3.16 adalah hasil dari pengambilan gambar yang digunakan untuk pendeteksian gambar. Selanjutnya aplikasi akan dialihkan pada mode *image cropping* untuk menyeleksi bagian dari objek kemeja yang diambil seperti terlihat pada Gambar 3.17



Gambar 3.17 Cropping Image



Panjang: 85 cm
Lebar: 62 cm

Gambar 3.18 Hasil Cropping Image

Pada Gambar 3.17 adalah proses *cropping image* yang dilakukan dengan pemotongan gambar sesuai dengan panjang dan lebar ukuran kemeja yang diamati. Untuk Gambar 3.18 adalah hasil akhir dari proses pendeteksian

ukuran kemeja dan dapat ditampilkan secara langsung nilai keluaran ukuran kemeja yang berada dibawah gambar kemeja.

3.6 Spesifikasi Sistem

3.6.1 Spesifikasi Software

3.6.1.1 OpenCV

OpenCV adalah singkatan dari *Open Computer Vision*, yaitu *library open source* yang dikhususkan untuk melakukan pengolahan citra. *OpenCV* dirancang untuk efisiensi komputasional dan dengan fokus yang kuat pada aplikasi real-time. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. *OpenCV* memiliki *Application Programming Interface (API)* untuk pengolahan tingkat tinggi maupun tingkat rendah. Pada *OpenCV* juga terdapat fungsi-fungsi siap pakai untuk me-load, menyimpan, serta mengakuisisi gambar dan video. *Library OpenCV* memiliki fitur-fitur sebagai berikut [3] :

- a. Manipulasi data gambar (mengalokasi memori, melepaskan memori, menduplikasi gambar, mengatur serta mengkonversi gambar).
- b. Manipulasi matriks dan vector
- c. Pengolahan citra dasar (penapisan, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi).
- d. Kalibrasi kamera.
- e. Pendeteksian gerakan.
- f. Pengenalan objek.
- g. GUI dasar (menampilkan gambar/video, mengontrol mouse/keyboard, scrollbar).

3.6.1.2 Android Studio

Android studio adalah *Integrated Development Environment (IDE)* resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event *Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan plugin *Android Development Tools (ADT)*. Android studio memiliki fitur [1]:

1. Projek berbasis pada *Gradle Build*
2. *Refactory* dan pembenahan bug yang cepat Tools baru yang bernama “*Lint*” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
3. Mendukung *Proguard And App-signing* untuk keamanan.
4. Memiliki GUI aplikasi android lebih mudah.
5. Didukung oleh *Google Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

Android Studio dapat digunakan untuk mengembangkan aplikasi dalam bahasa Java, sesuai dengan Bahasa yang digunakan pada sistem yang dibuat pada proyek akhir ini.

3.6.2 Spesifikasi Hardware

3.5.2.1 Smartphone Samsung Galaxy Core Duos I8262

Tabel 3.2 Spesifikasi *Smartphone*

Ukuran Layar	480 x 800 pixels, 4.3 inches (~217 ppi pixel density) Multitouch
Kamera	5 MP, 2592 x 1944 pixels
Fitur OS	Android OS v4.1.2 (Jelly Bean)
CPU	Snapdragon MSM8225 S4 Play CPU Dual-core 1.2 GHz Cortex-A5GPU Adreno 203 Sensors Accelerometer, Proximity, Compass

4. Pengujian

4.1 Pengujian Sistem Aplikasi Berdasarkan Jarak

4.1.1 Jarak 1 Meter

Pengujian ini dilakukan dengan cara menemukan ukuran kemeja pada jarak 1 meter dengan tujuan untuk mengetahui jarak dan jangkauan yang paling tepat dari data yang diambil sesuai dengan perhitungan yang telah dilakukan. Pengujian ini dilakukan dengan ukuran kemeja S, M, L, XL dan XXL.

Berikut adalah tabel hasil pengujian :

Tabel 4.1 Pengujian Jarak 1 Meter

Jarak	Ukuran	Sebenarnya		Hasil	
		Panjang	Lebar	Panjang	Lebar
1 meter	S	70	52	155	116
	M	73	53	151	103
	L	74	56	146	111
	XL	78	60	155	116
	XXL	79	56	158	116

Dari hasil Tabel 4.1 dilakukan pengujian aplikasi pendeteksi ukuran kemeja pada 5 *sample* data dari ukuran kemeja yang berbeda dengan pengambilan jarak 1 Meter. Pengujian dengan jarak 1 Meter dapat disimpulkan bahwa tingkat keakuratan konversi *pixel* ke centimeter sangat jauh diluar estimasi, ini menandakan bahwa pendeteksian objek kemeja dengan jarak 1 Meter tidak tepat untuk menentukan nilai ukuran kemeja.

4.1.2 Jarak 2 Meter

Pengujian ini dilakukan dengan pengambilan jarak 2 Meter untuk mengukur keakuratan deteksi *pixel* dan konversi ke centimeter untuk menentukan keakuratan konversi *pixel* dengan parameter jarak kamera *smartphone* dengan objek kemeja yang diamati sebanyak 2 kali percobaan.

1. Percobaan Pertama

Tabel 4.2 Pengujian Jarak 2 Meter Percobaan Pertama

Jarak	Ukuran	Sebenarnya		Hasil		Akurasi
		Panjang	Lebar	Panjang	Lebar	
2 meter	S	70	52	73	53	92.12%
	M	73	53	77	53	94.96%
	L	74	56	77	60	90.9%
	XL	78	60	81	62	93.1%
	XXL	79	56	81	57	95.95%

Dari hasil Tabel 4.2 dilakukan pengujian aplikasi pendeteksi ukuran kemeja pada 5 *sample* data dari ukuran kemeja yang berbeda dengan pengambilan jarak 2 Meter. Analisa data keakuratan dari pengambilan jarak 2 Meter rata-rata adalah 93.4% hasil pengambilan data *sample* sebanyak 5 macam ukuran kemeja.

Data dari percobaan pertama masih terdapat selisih antara 2 sampai 3 centimeter yang masih memungkinkan untuk lebih akurat lagi. Oleh karena itu dilakukan perbaikan pada parameter pikselnya. Setelah dilakukan perbaikan maka didapat data pada percobaan kedua.

2. Percobaan Kedua

Tabel 4.3 Pengujian Jarak 2 Meter Percobaan Kedua

Jarak	Ukuran	Sebenarnya		Hasil		Akurasi
		Panjang	Lebar	Panjang	Lebar	
2 meter	S	70	52	71	52	98.78%
	M	73	53	74	53	98.74%
	L	74	56	74	57	98.7%
	XL	78	60	79	60	98.62%
	XXL	79	56	80	57	97.3%

Dari hasil Tabel 4.3 dilakukan pengujian aplikasi pendeteksi ukuran kemeja pada 5 *sample* data dari ukuran kemeja yang berbeda dengan pengambilan jarak 2 Meter. Analisa data keakuratan dari pengambilan jarak 2 Meter rata-rata adalah 98.4% hasil pengambilan data *sample* sebanyak 5 macam ukuran kemeja.

Data dari hasil perbaikan pada percobaan kedua lebih akurat 5% dari percobaan pertama. Dapat disimpulkan bahwa pengambilan dengan jarak 2 Meter cukup baik untuk melakukan pendeteksian ukuran kemeja menggunakan *OpenCV* pada *Smartphone*.

5. Kesimpulan dan Saran

Kesimpulan

Kesimpulan pada Proyek Akhir ini adalah :

1. Aplikasi dapat berjalan dengan baik menggunakan resolusi kamera *smartphone* yang digunakan.
2. Aplikasi ini hanya digunakan menggunakan *smartphone* yang telah ditentukan. Jika ingin menggunakan pada *smartphone* lainnya harus di atur ulang sesuai dengan resolusi *smartphone*.
3. Ketepatan jarak dan jangkauan kamera terhadap objek jauh lebih akurat pada pengambilan jarak 2 meter sedangkan pada pengambilan jarak 1 meter nilai keluarannya cukup jauh dari ukuran kemeja sebenarnya.

Saran

Adapun saran yang diberikan berdasarkan hasil analisis yang telah dilakukan pada Proyek Akhir ini adalah :

1. Menggunakan metode lain agar aplikasi dapat digunakan pada berbagai macam *smartphone* yang berbeda
2. Untuk selanjutnya terkait pengamiblan jarak lebih dikembangkan lagi sehingga dapat menentukan ukuran kemeja dengan berbagai jarak

DAFTAR PUSTAKA

- [1] Abdul Fadlil, Aditiya Dwi Candra. “*Sistem Penentuan Sudut Diam Granul Menggunakan Metode Pengolahan Citra Berbasis Android*”. Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan. 2017
- [2] Andi Juansyah. “*Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System (A-GPS) Dengan Platform Android*”. Teknik Informatika, Universitas Komputer Indonesia. 2015
- [3] Bradski, G. Kaehler, A. Learning OpenCV: *Computer Vision with the OpenCV Library*. California: O’Reilly Media Inc. 2008
- [4] Gonzalez, C. Rafael. dan Woods, E. Richard. *Digital Image Processing, 3rd Ed.* New Jersey: Prentice Hall. 2008
- [5] Khairul Umam, Benny Sukma Negara. *Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Substraction Dan Operasi Morfologi*. Teknik Informatika, UIN Sultan Syarif Kasim Riau. 2016
- [6] M. Soleh Gangsarestu. *Rancang Bangun Sistem Pengukuran Badan Pria Untuk Menentukan Ukuran Baju Berbasis Kamera Kinect*. Surabaya: Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. 2015
- [7] M. Stefani Putri Rizky. *Pakaian Sebagai Komunikasi (Pemakaian Baju Bekas Impor Sebagai Media Untuk Mengkomunikasikan Identitas Sosial)*. Salatiga: Program Studi Komunikasi. Universitas Kristen Satya Wacana. 2012
- [8] Munir, Rinaldi. “*Pengolahan Citra Digital dengan Pendekatan Algoritmik*”. Bandung: Penerbit Informatika. 2004
- [9] Rio Pradhitya. “*Pembangunan Aplikasi deteksi dan Tracking Warna Virtual Drawing Menggunakan Algoritma Color Filtering*”. Teknik Informatika, Universitas Komputer Indonesia. 2015
- [10] Sutoyo, T, dkk. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi. 2009