

## APLIKASI PENDETEKSI KOTAK UNTUK MENENTUKAN UKURAN TUBUH MENGGUNAKAN ALGORITMA HSV

M. Alfian Qadar <sup>\*)</sup>, Achmad Hidayatno dan Yosua Alvin Adi S.

Departemen Teknik Elektro, Universitas Diponegoro  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)</sup>E-mail : [alfan.mq@outlook.com](mailto:alfan.mq@outlook.com)

### Abstrak

Belakangan ini teknologi pendeteksian objek banyak dikembangkan, mengingat pentingnya akan kebutuhan hal tersebut dalam kehidupan untuk menganalisa dan mengamati lingkungan sekitar. Berbagai cara digunakan untuk melakukan pendeteksian, salah satunya adalah pengolahan citra digital. Citra digital mengandung sejumlah elemen yang dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam computer vision. Salah satu elemen dasar yang paling penting dalam citra digital ialah warna. Menurut beberapa penelitian model warna HSV adalah model warna yang lebih baik digunakan untuk keperluan pengolahan citra dan computer vision. Pada penelitian Tugas Akhir ini, pengolahan citra digunakan untuk mendeteksi dan komputasi dimensi objek kotak menggunakan algoritma HSV sehingga didapatkan parameter keluaran yang sudah sesuai dengan dimensi asli objek. Berdasarkan hasil tersebut, parameter keluaran dari proses deteksi objek kotak layak digunakan sebagai identifikasi dan perhitungan ukuran tubuh manusia sehingga pada pengklasifikasian ukuran baju pengguna didapatkan hasil yang sesuai dengan standar SNI.

**Kata kunci :** *Pengolahan Citra Digital, HSV, Warna, Deteksi Objek, Ukuran Tubuh.*

### Abstract

*Lately, object detection technology has been developed, given the importance of the need for this in life to analyze and observe the surrounding environment. Various methods are used to detect, one of which is digital image processing. Digital images contain a number of elements that are manipulated in image processing and further exploited in computer vision. One of the most important basic elements in digital images is color. According to some studies the HSV color model is a better color model used for image processing and computer vision. In this final project, image processing is used to detect and compute the dimensions of the box object using the HSV algorithm so that the output parameters that are in accordance with the original dimensions of the object are obtained. Based on these results, the output parameters of the box object detection process are feasible to be used as identification and calculation of the size of the human body so that the classification of the size of the user's clothes is obtained in accordance with SNI standards.*

**Keywords:** *Digital Image Processing, HSV, Color, Object Detection, Body Size.*

### 1. Pendahuluan

Penerapan teknik pengolah citra digital pada rangkaian evaluasi dan identifikasi telah banyak diteliti oleh berbagai peneliti. Penerapan teknik tersebut dilakukan dalam berbagai bidang yang memungkinkan kita untuk dapat menilai, menyimpulkan dan mendapatkan suatu informasi pada suatu citra digital [1]. Teknologi dengan menggunakan metode pengolahan citra digital (digital image processing) banyak dikembangkan belakangan ini, hal ini disebabkan kemudahan dalam penggunaannya serta didukung dengan kemampuan komputer yang memadai untuk melakukan komputasi. Saat ini teknologi pendeteksian objek banyak dikembangkan, mengingat pentingnya akan kebutuhan hal tersebut dalam kehidupan

untuk menganalisa dan mengamati lingkungan sekitar. Berbagai cara digunakan untuk melakukan pendeteksian, di antaranya adalah dengan sensor magnetik, gelombang, dan pengolahan citra digital [2].

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam computer vision. Salah satu elemen dasar yang paling penting ialah warna. Citra berwarna adalah citra yang memiliki 3 buah kanal warna di dalamnya. Pada umumnya jenis citra ini terbentuk dari komponen merah/red (R), hijau/green (G), dan biru/blue (B) yang dimodelkan kedalam ruang warna RGB. RGB adalah standar yang digunakan untuk menampilkan citra berwarna pada layar televisi maupun layar komputer.

Namun juga diperlukan citra yang menggunakan ruang warna berbeda, seperti CMYK (Cyan, Magenta, Yellow, Black), HSV (Hue, Saturation, Value), YCbCr (Luma, Chroma blue, Chroma red), dan Lab ( $L^*a^*b^*$ ). Namun ada banyak model warna lain yang dapat digunakan untuk keperluan seperti pengolahan citra karna memiliki hasil yang lebih baik dari model warna RGB. Menurut beberapa penelitian model warna HSV adalah model warna yang lebih baik digunakan untuk keperluan pengolahan citra dan computer vision [3].

Ada banyak aplikasi yang telah dirancang menggunakan model HSV diantaranya "Pendeteksi Objek menggunakan Algoritma HSV berbasis Raspberry PI 3b" yang digunakan pada deteksi bola berwarna [2], "Aplikasi Pendeteksi Tangan dalam Ruang Warna YCbCr dan HSV pada Virtual Mouse" yang mendeteksi telapak tangan untuk gerakan mouse [4], "Pendeteksi Objek Bola dengan Metode Color Filtering HSV pada Robot Soccer Humanoid" yang digunakan pada deteksi bola pada mata robot [5]. Pada Tugas Akhir ini yang menjadi pembeda dalam penelitian sebelumnya adalah memanfaatkan hasil deteksi dan komputasi objek deteksi kotak menggunakan algoritma HSV yang digunakan untuk mendeteksi objek kotak serta mendapatkan nilai dimensi dari objek tersebut.

Pengolahan citra digital merupakan bidang yang dicirikan akan kebutuhan eksperimen kerja yang ekstensif untuk menghasilkan solusi layak pada masalah yang diberikan [8]. Seperti pepatah "A picture is more than thousand words" dengan sebuah gambar akan bercerita lebih banyak dibandingkan dengan seribu kata. Namun sayangnya gambar juga seringkali mengalami penurunan kualitas, baik disebabkan karena pengaruh cahaya, terdapat derau (noise), terlalu kontras, kurang tajam (blur), dan lain sebagainya yang menyebabkan hilangnya informasi yang ingin disampaikan [9].

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode [10]. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas [11] dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

Python mendukung pemrograman multi paradigma yakni pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada Python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, Python umumnya digunakan untuk pemrograman dasar atau tingkat lanjut. Pada praktiknya, penggunaan bahasa ini diimplementasikan lebih luas bahkan mencakup konteks

pemanfaatan yang umumnya tidak dilakukan dengan menggunakan pemrograman lainnya [12].

Python sangat ideal digunakan dalam proyek yang membutuhkan pengembangan yang cepat. Python mendukung banyak filosofi pemrograman, sehingga lebih baik digunakan dalam pembuatan program yang membutuhkan fleksibilitas. Banyak paket dan modul yang sudah ditulis untuk Python sehingga memberikan fleksibilitas dan menghemat waktu dalam pengerjaannya [14].

Tkinter, atau "Tk interface", adalah modul python yang menyediakan antarmuka pada toolkit GUI Tk yang dikembangkan pada Tool Command Language (TCL) dan merupakan modul multiplatform yang mendukung untuk Linux, MAC OS dan MS Windows. Tk awalnya tersedia di Linux dan MAC OS, dan dapat dengan mudah diinstal pada MS Windows, namun modul ini bukan bagian langsung dari Python. Namun setelah ada perubahan, Tkinter menjadi bagian dari Python dalam penggunaannya disebut dengan "Tkinter" pada versi sebelum 3, dan "tkinter" pada versi saat ini. Widget, Geometry Management, dan Event Handling adalah tiga konsep utama Tk yang juga berlaku untuk versi Tkinter [15].

Open Source Computer Vision Library (OpenCV) adalah aplikasi open source untuk computer vision dan library perangkat lunak dalam pembelajaran mesin. OpenCV dirancang untuk menyediakan infrastruktur umum untuk aplikasi computer vision dan untuk penggunaan machine perception pada produk komersial. OpenCV merupakan produk berlisensi Berkeley Software Distribution (BSD), yang merupakan lisensi perangkat lunak bebas permisif dengan memberlakukan pembatasan minimal pada penggunaan dan redistribusi perangkat lunak tertutup. OpenCV dapat digunakan dalam bidang apapun dengan memanfaatkan dan memodifikasi kode yang sesuai dalam penggunaannya [16].

Library OpenCV memiliki lebih dari 2500 algoritma yang dioptimalkan, yang mencakup serangkaian computer vision, algoritma mesin dan pembelajaran mesin yang mutakhir dan komprehensif. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia di video, melacak pergerakan menggunakan kamera, melacak objek yang bergerak, mengekstraksi model objek ke 3D, menghasilkan geometri titik 3D dari kamera stereo, menemukan gambar serupa dari basis data gambar, menghapus red eye dari gambar yang diambil menggunakan flash, aplikasi penggerak benda dengan mengikuti gerakan mata, mengenali objek pemandangan dan membuat mark untuk menambahkannya dengan augmented reality dan lain sebagainya [8]. Penggunaan OpenCV pada penelitian kali yakni menggunakan beberapa algoritma dalam pengolahan citra seperti read, resize, color conversion, thresholding, contour dan bounding.

OpenCV memiliki lebih dari 47 ribu dari komunitas pengguna dan dengan perkiraan jumlah unduhan melebihi 18 juta. Library OpenCV telah digunakan secara luas dan tersebar pada perusahaan, kelompok penelitian dan badan pemerintah [16].

Aplikasi pendeteksi kotak untuk menentukan ukuran tubuh menggunakan algoritma HSV sebagai thresholding objek referensi. Pada pengolahan objek referensi digunakan fitur contouring serta bounding untuk mendapatkan dimensi dari objek referensi kotak. Setelah dimensi didapatkan dilakukan perhitungan dalam mencari parameter konversi panjang serta tinggi dari objek referensi. Parameter digunakan pada sistem identifikasi ukuran tubuh untuk menentukan ukuran baju terkhusus pada sistem perhitungan dari ukuran tubuh sebenarnya.

Bagian 1 menjelaskan tentang latar belakang serta dasar teori yang digunakan pada Tugas Akhir ini. Bagian 2 menjelaskan metode yang digunakan serta perancangan terhusus pada perangkat keras serta perangkat lunak Tugas Akhir ini. Bagian 3 menjelaskan pengujian beserta analisis dari beberapa variasi pengujian. Bagian 4 menjelaskan kesimpulan serta saran yang memungkinkan untuk Penelitian atau Tugas Akhir berikutnya.

## 2. Metode

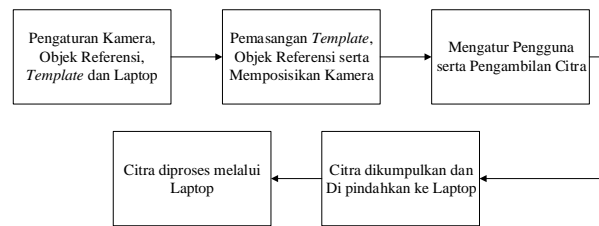
Perancangan sistem alat terdiri atas perancangan hardware dan software.

### 2.1. Perancangan Perangkat Keras

Perangkat keras dalam Tugas Akhir ini terdiri dari 4 bagian utama, yaitu :

1. Handphone digunakan sebagai pengambilan citra, media untuk pengiriman file hasil penangkapan citra serta pengaturan berdasarkan ketentuan yang digunakan dalam pengolahan citra pada perangkat lunak.
2. Template berfungsi sebagai pengatur objek manusia, dan sebagai komponen pendukung objek referensi.
3. Laptop digunakan sebagai media pengolahan citra dengan dilengkapi dengan perangkat lunak utama yang digunakan diantaranya Python beserta library yang telah disiapkan untuk pengolahan citra.
4. Objek Referensi sebagai komponen utama dalam proses objek deteksi kotak. Nilai dimensi dari deteksi tersebut digunakan sebagai referensi pada perhitungan ukuran tubuh.

Untuk perancangan perangkat keras dapat dilihat melalui diagram alir pada Gambar 1.



Gambar 1. Gambaran perancangan perangkat keras

Semua komponen hardware yang digunakan dalam Tugas Akhir ini dirancang dan disiapkan. Perencanaan penempatannya berada di Laboratorium KPS (Komunikasi dan Pengolahan Sinyal) di Jurusan Teknik Elektro Universitas Diponegoro. Gambar 2. menunjukkan tampilan pada perancangan dan penempatan perangkat keras pada Tugas Akhir ini.



Gambar 2. Tampilan fisik perancangan perangkat keras

### 2.2. Perancangan Perangkat Lunak

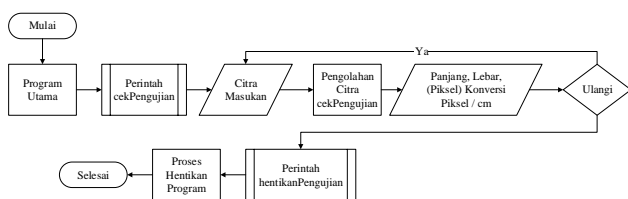
Pada perancangan perangkat lunak Tugas Akhir ini, ada beberapa program yang akan digunakan dengan tujuan untuk pengolahan citra masukan yang didapat pada handphone adalah sebagai berikut

- Python 3.6.8 sebagai aplikasi utama perancang perangkat lunak.
- OpenCV sebagai library untuk pengolahan citra objek secara umum baik objek manusia maupun objek kotak.
- Tkinter sebagai library untuk perancangan GUI perangkat lunak.
- NumPy sebagai library untuk komputasi dan perhitungan.
- Imutils sebagai library untuk pengambilan nilai kontur objek referensi.

Perancangan perangkat lunak dalam Tugas Akhir ini terdiri dari 2 bagian utama, yaitu :

1. Perancangan desain GUI program utama.
2. Perancangan algoritma perintah.

Berikut proses perancangan algoritma dan diagram alir pada Gambar 3.



Gambar 3. Diagram alir program secara menyeluruh

Perancangan program dilakukan menggunakan IDLE Python v3.6.8 dengan nama program Aplikasi\_TA.py. Berikut merupakan senarai dari library yang dipakai dalam pemograman

```
import tkinter
import cv2
import imutils
import tkinter as tk
import numpy as np
from tkinter import filedialog
from tkinter import *
from PIL import ImageTk, Image
```

Seluruh libray diatas mendukung proses perancangan source code pada aplikasi deteksi kotak. Berikut penjelasan lengkap mengenai perancangan perangkat lunak pada Tugas Akhir ini

- Perancangan GUI

Perancangan GUI terdiri dari pembuatan main windows dengan geometri GUI adalah 460x600 piksel, pembuatan nama judul jendela utama, pembuatan beberapa string judul umum, pembuatan label logo undip dan logo himpunan mahasiswa elektro, pembuatan tombol perintah untuk melakukan atau mengesekusi pengolahan yang dirancang, serta pembuatan frame pendukung untuk mengelompokkan hasil keluaran proses deteksi serta hasil keluaran berupa gambar. Berikut senarai dari GUI yang telah dirancang dengan menggunakan library tkinter: Berikut tampilan GUI yang dirancang menggunakan bahasa pemograman Python pada Gambar 4.



Gambar 4. Tampilan GUI

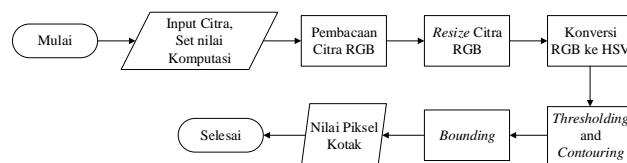
Persiapan penguajian dilakukan pada sisi software. Pengaturan program dilakukan dengan mengatur parameter `known_width`, `known_height`, `upper_red` dan `lower_red`. Parameter ini digunakan untuk mendeklarasi ukuran kotak asli dengan `known_width` (panjang) 21 cm dan `known_height` (tinggi) 15 cm. Parameter `upper_red` dan `lower_red` digunakan sebagai rentang untuk thresholding warna kotak yang telah dirancang sebelumnya. Rentang tersebut diklasifikasi berdasarkan standar range OpenCV yakni 0 – 255 dengan rentang Hue yakni 160 – 190, Saturation dengan rentang 20 – 255 serta nilai Value yakni 70 – 255.

- Perancangan Algoritma Perintah

Perancangann algoritma perintah memiliki 2 fungsi utama yakni perintah deteksi kotak “lakukanPenguajian” serta perintah berhenti “hentikanPenguajian”. Fungsi “lakukanPenguajian” merupakan perintah untuk menjalankan proses deteksi kotak dan menampilkan keluaran hasil proses deteksi serta komputasi. Fungsi berhenti atau “hentikanPeguajian” digunakan untuk menghentikan seluruh aktifitas program dengan menghentikan (destroy) main windows Python.

• Perintah Deteksi Kotak

Algoritma program ini terintegrasi dengan program utama yang dijalankan melalui perintah “lakukanPenguajian” dengan menggunakan tombol Deteksi yang ada pada GUI. Perintah akan menjalankan senarai yang telah dirancang berdasarkan Gambar 5.



Gambar 5. Rangkaian keseluruhan perintah deteksi kotak

Berikut adalah senarai skrip untuk perintah “lakukanPenguajian”:

```
#Pemilihan File citra pada laptop
def open_img():
    x = filedialog.askopenfilename(title='open')
    return x
def lakukanPenguajian():
    imagereal = cv2.imread(open_img())
    #membaca image masukan
```

Pertama program akan melakukan pemilihan file citra untuk dideteksi. Setelah melakukan pemilihan file citra. Citra yang digunakan memiliki format jpeg dengan kanal citra berwarna RGB. Citra tersebut dibaca dan dianalisis menggunakan perintah `cv2.imread`. Setelah proses pembacaan citra, dilakukan pengaturan pada parameter yang dibutuhkan untuk komputasi program. Parameter yang dibutuhkan yakni jarak kamera, panjang objek, tinggi objek serta parameter rentang warna HSV merah

yang digunakan pada proses thresholding yakni menggunakan nilai upper and lower warna HSV tersebut.

```
#Pemilihan File citra pada laptop
KNOWN_DISTANCE = 180.0
#jarak kamera ke referensi objek
KNOWN_WIDTH = 21
#panjang referensi objek yang telah
diketahui

KNOWN_HEIGHT = 15
#tinggi referensi objek yang telah
diketahui

#menentukan batas bwh hsv merah
lower_red = np.array([160, 20, 70])
#menentukan batas bwh hsv merah
upper_red = np.array([190, 255, 255])
#membaca image masukan
```

Selanjutnya citra di resize menggunakan fungsi cv2.resize dengan resolusi yang ditetapkan yakni 680x450 piksel. Metode yang digunakan dalam resize citra masukan menggunakan metode bilinear interpolation. Pada senarai tidak terdapat metode dalam resize karna pada kondisi default fungsi ini menetapkan metode ini yang digunakan dalam pengolahannya. Berikut senarai dari resize citra masukan.

```
image = cv2.resize(imagereal, (450, 680))
#mengatur ukuran image
```

Citra hasil proses resize dikonversikan ke model warna HSV dengan menggunakan perintah cv2.cvtColor.

```
def find_marker(image):
#mengubah gambar dari rgb ke hsv
hsv = cv2.cvtColor(image,
cv2.COLOR_BGR2HSV)
```

Setelah konversi ke kanal warna HSV, program akan melakukan proses thresholding atau mencari dimensi berdasarkan rentang warna HSV (lower dan upper) dari objek referensi menggunakan fungsi cv2.inRange. Program akan mencari dimensi kontur hasil proses thresholding dan dilakukannya proses contouring sehingga permukaan bidang objek deteksi dapat dibedakan menggunakan cv2.findContours. Kemudian nilai array kontur disimpan agar dapat dipisahkan serta dapat ditandai menggunakan fungsi imutils.grab.contours. Setelah proses contouring selesai, dilakukan proses bounding untuk membentuk dimensi serta memperhalus dimensi pada perintah cv2.approxPolyDP dan cv2.boundingRect.

```
#proses thresholding
mask = cv2.inRange(hsv, lower_red,
upper_red)
#proses contouring
cnts = cv2.findContours(mask.copy(),
cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
#menemukan contour pada masukan image
cnts = imutils.grab_contours(cnts)
c = sorted(cnts, key=lambda x:
cv2.contourArea(x),
```

```
reverse=True)
for cnt in c:
approx = cv2.approxPolyDP
(cnt, 0.04*cv2.arcLength
(cnt, True), True)
if len(approx)==4:
#proses bounding
persegi = cv2.boundingRect(cnt)
return persegi
```

Setelah proses bounding, program melakukan pengambilan nilai parameter panjang, tinggi, panjang konversi dan tinggi konversi pada marker hasil proses bounding serta perhitungan nilai konversi pada panjang2 dan tinggi2.

```
#deteksi benda dan ketahu panjangnya
(pixel)
marker = find_marker(image)
panjang = marker[2]
#mengeluarkan nilai dari panjang
piksel
tinggi = marker[3]
#mengeluarkan nilai dari tinggi
piksel
#membagi panjang piksel dengan pjg
sesungguhnya
panjang2=(panjang / KNOWN_WIDTH)
#membagi tinggi piksel dengan pjg
sesungguhnya
tinggi2=(tinggi / KNOWN_HEIGHT)
```

Untuk mendapatkan keluaran dari perintah “lakukanPengujian” pada program utama ditambahkan widget label untuk menampilkan gambar hasil deteksi serta masukan aplikasi. Selain itu widget string yang digunakan untuk menampilkan keluaran panjang dan lebar kotak referensi dalam satuan piksel serta konversi panjang dan lebar dalam bentuk piksel/cm.

```
#menampilkan lokasi path dalam variabel
widget string lokasiPath
path = StringVar()
Lokasipath = Label(JendelaMain,
textvariable=path)
Lokasipath.config(font=("Helvetica",
10),bg='white')
path.set("Path :" + str(x))
Lokasipath.pack()
Lokasipath.place(x=10, y=365)
#Menampilkan hasil panjang kotak referensi
dalam piksel
path6 = StringVar()
Lokasipath6 = Label(JendelaMain,
textvariable=path6)
Lokasipath6.config(font=("Helvetica",
10),bg='white')
path6.set("Panjang :" + str(panjang) + "
pixel")
Lokasipath6.pack()
Lokasipath6.place(x=140, y=495)
#Menampilkan konversi panjang kotak dalam
piksel/cm
path7 = StringVar()
Lokasipath7 = Label(JendelaMain,
textvariable=path7)
Lokasipath7.config(font=("Helvetica",
10),bg='white')
```

```

path7.set("P. Pixel :" + "%.2f" % (panjang2)
+ " pixel/cm")
Lokasipath7.pack()
Lokasipath7.place(x=140, y=535)
#Menampilkan hasil tinggi kotak referensi
dalam piksel
path8 = StringVar()
Lokasipath8 = Label(JendelaMain,
textvariable=path8)
Lokasipath8.config(font=("Helvetica",
10),bg='white')
path8.set("Tinggi :" + str(tinggi)+ "
pixel")
Lokasipath8.pack()
Lokasipath8.place(x=140, y=515)
#Menampilkan konversi tinggi kotak dalam
piksel/cm
path9 = StringVar()
Lokasipath9 = Label(JendelaMain,
textvariable=path9)
Lokasipath9.config(font=("Helvetica",
10),bg='white')
path9.set("T. Pixel :" + "%.2f" % (tinggi2)
+
" pixel/cm")
Lokasipath9.pack()
Lokasipath9.place(x=140, y=555)
#Menampilkan citra hasil deteksi kotak
image7 = cv2.resize(image, (202,270),
interpolation=cv2.INTER_AREA)
cv_img4 = cv2.cvtColor(image7,
cv2.COLOR_BGR2RGB)
height, width, no_channels = cv_img4.shape
photo4 = ImageTk.PhotoImage(image =
Image.fromarray(cv_img4))
label4 = Label(image=photo4,
height=270,width=202,anchor=E)
label4.image=photo4
label4.pack()
label4.place(x=230, y=80)

```

Berikut merupakan keluaran hasil proses perintah “lakukanPenguujian” pada Gambar 6.



Gambar 6. Keluaran hasil proses deteksi kotak

- Perintah Berhenti

Algoritma program ini terintegrasi dengan program utama yang dijalankan melalui perintah “hentikanPenguujian” dengan menggunakan tombol “Berhenti” yang ada pada

GUI. Perintah akan menghentikan seluruh aktifitas dari program baik saat memanggil library OpenCV dan lainnya. Berikut senarai dari program “hentikanPenguujian”.

```

def hentikanPenguujian() :
    JendelaMain.destroy()

```

Algoritma diatas akan memerintahkan program akan menutup seluruh proses aktifitas jendela main.

### 3. Pengujian dan Analisis

Pada pengujian ini akan dilakukan pengaturan aplikasi kamera handphone agar masukan citra sesuai dengan ketentuan dari aplikasi. Pada pengujian ini objek kotak akan dideteksi berdasarkan aspek kondisi lingkungan yang memungkinkan dapat mempengaruhi proses pendeteksian. Sebelum pengujian dilakukan, file masukan citra dipindahkan lalu diproses menggunakan aplikasi yang telah dijalankan pada laptop. Berikut adalah kondisi lingkungan objek yang akan diujikan.

- 1 Variasi Pencahayaan.
- 2 Variasi Jarak.
- 3 Implementasi dalam Program Utama

Pengujian menggunakan sebuah objek deteksi berupa objek kotak berwarna merah. Penetapan jumlah kotak saat pengujian ditentukan agar pengujian lebih terfokus pada dampak pengaruh lingkungannya. Objek dan lingkungan akan diatur berdasarkan variasi kondisi yang sudah direncanakan. Pengambilan citra pengujian dilakukan sebanyak 5 variasi pada setiap kondisinya. Parameter yang akan diambil dari pengujian ini adalah:

- 1 Terdeteksinya objek referensi.
- 2 Panjang objek.
- 3 Tinggi objek.
- 4 Nilai konversi panjang serta tinggi objek deteksi kotak dari ukuran piksel ke ukuran sebenarnya.

#### 3.1. Pengujian variasi pencahayaan

Tabel 1. Hasil pengujian menggunakan variasi cahaya

Kondisi	Panjang (piksel)	Tinggi (piksel)	Nilai Konversi Panjang (piksel/cm)	Nilai Konversi Tinggi (piksel/cm)	Terdeteksi
Gelap	25	29	1.19	1.93	Terdeteksi
Hampir Gelap	64	52	3.05	3.47	Terdeteksi
Normal	60	51	2.95	3.4	Terdeteksi
Hampir Terang	62	51	2.95	3.4	Terdeteksi
Terang	61	50	2.90	3.33	Terdeteksi

Pengujian ini dilakukan untuk mengetahui pengaruh pencahayaan terhadap proses deteksi kotak. Perancangan pengujian dilakukan dengan variasi pencahayaan yakni mulai dari gelap, hampir gelap, sedang (normal), hampir

terang, terang, lalu data yang didapatkan diolah didalam Tabel 1. Berikut hasil yang didapatkan saat pengujian pada Tabel 1.

Dari Tabel 1 dapat disimpulkan bahwa adanya pengaruh pencahayaan dalam proses deteksi kotak. Berdasarkan percobaan didapatkan bahwa dalam variasi cahaya hampir gelap, normal, hampir terang dan terang memiliki hasil yang hampir menyerupai sedangkan untuk variasi gelap jauh berbeda dengan variasi lainnya. Pada hasil terlihat bahwa keluaran parameter panjang didapatkan nilai keluaran dengan rentang 60 – 64 piksel. Pada tinggi didapatkan nilai keluaran dengan rentang 50 – 52 piksel. Pada nilai konversi panjang didapatkan nilai keluaran dengan rentang 2.90 – 3.05 piksel/cm. Pada nilai konversi tinggi didapatkan nilai keluaran dengan rentang 2.90 – 3.05 piksel/cm. Namun pada pencahayaan gelap didapatkan nilai panjang 25 piksel, tinggi 29 piksel, konversi panjang 1.93 piksel/cm serta tinggi 1.19 piksel/cm yang menunjukkan perbedaan dengan variasi pencahayaan lainnya.

Dari uraian sebelumnya disimpulkan bahwa deteksi kotak dengan pengujian variasi pencahayaan memiliki hasil deteksi bagus, namun terdapat sebuah variasi yang memiliki parameter yang jauh berbeda dengan variasi lainnya dikarenakan pencahayaan yang terlalu gelap sehingga mempengaruhi kualitas warna HSV. Hal ini menyebabkan tidak terdeteksinya kotak secara menyeluruh karena rentang warna HSV yang digunakan saat proses deteksi kotak tidak menangkap citra yang berwarna gelap sehingga dalam prosesnya hanya sebagian kotak yang terdeteksi.

### 3.2. Pengujian variasi jarak

Pengujian ini dilakukan untuk mengetahui pengaruh jarak terhadap proses deteksi kotak. Perancangan pengujian dilakukan dengan variasi jarak 50cm, 100cm, 200cm, 300cm dan 400cm dalam keadaan kondisi pencahayaan yang terang. Setelah dilakukan pengujian terhadap 5 variasi jarak, data yang didapatkan diolah didalam Tabel 2. Berikut Tabel 2 merupakan hasil yang didapatkan saat pengujian

Tabel 2 Hasil Pengujian menggunakan variasi jarak

Variasi Jarak (cm)	Panjang (piksel)	Tinggi (piksel)	Nilai Konversi Panjang (piksel/cm)	Nilai Konversi Tinggi (piksel/cm)	Terdeteksi
50	197	156	9.39	10.4	Terdeteksi
100	110	89	5.24	5.93	Terdeteksi
200	52	44	2.48	2.93	Terdeteksi
300	34	27	1.62	1.80	Terdeteksi
400	27	23	1.29	1.53	Terdeteksi

Pada Tabel 2 dapat disimpulkan bahwa adanya pengaruh jarak dalam proses deteksi kotak. Berdasarkan percobaan didapatkan bahwa variasi jarak tidak mempengaruhi dari

proses pencarian dimensi objek. Namun mempengaruhi dalam menentukan nilai konversi panjang sebenarnya dalam deteksi kotak. Untuk menentukan perbandingan nilai konversi dari hasil pengujian variasi jarak, nilai panjang serta tinggi yang didapat pada pengolahan dibandingkan dengan dimensi panjang dan tinggi sebenarnya. Diketahui bahwa rasio dimensi dari ukuran kotak sebenarnya ialah 7 : 5. Berdasarkan rasio tersebut nilai panjang dan tinggi pada citra dirubah menjadi ukuran rasio yang mendekati dengan dimensi kotak sebenarnya baik dalam sisi panjang maupun sisi tinggi. Berikut merupakan perbandingan hasil percobaan pada Tabel 3.

Tabel 3 Perbandingan hasil pengujian variasi jarak

Variasi Jarak (cm)	Panjang (piksel)	Tinggi (piksel)	Perbandingan Rasio berdasarkan panjang	Perbandingan Rasio berdasarkan tinggi
50	197	156	7 : 5.54	6.31 : 5
100	110	89	7 : 5.66	6.17 : 5
200	52	44	7 : 5.92	5.71 : 5
300	34	27	7 : 5.55	6.29 : 5
400	27	23	7 : 5.96	5.86 : 5

Berdasarkan Tabel 3 merupakan perbandingan dan rasio dimensi kotak hasil proses deteksi kotak. Disimpulkan bahwa jarak dapat mempengaruhi deteksi kotak terkhusus pada nilai panjang dan tinggi dari dimensi kotak. Nilai dimensi kotak yang didapat setelah proses deteksi variatif sehingga persebaran nilai rasio yang didapatkan seharusnya cukup mendekati nilai rasio dari dimensi kotak sebenarnya. Namun pada pengujian didapatkan parameter rasio yang memiliki error sekitar 0-2 untuk panjang dan tinggi. Error terjadi dikarenakan faktor dimana semakin jauh objek yang dideteksi semakin sulit untuk Selain itu terdapat faktor lain seperti peletakkan objek referensi yang kurang rapi sehingga menyebabkan dalam proses bounding juga mengambil kontur kotak selain objek kotak tersebut k menentukan panjang dan tinggi dari citra tersebut.

### 3.3. Pengujian pada program utama

Setelah dilakukannya pengujian terhadap lingkungan, perlu adanya uji coba pengaruh proses deteksi kotak terhadap identifikasi dan perhitungan ukuran tubuh serta klasifikasi ukuran baju. Pengujian diperlukan agar mengetahui kelebihan dan kekurangan dari aplikasi yang terintegrasi dengan program utama. Pengujian yang dilakukan yakni dengan meneliti 25 citra tubuh dan menganalisa setiap proses program utama yang berkaitan dengan deteksi kotak. Pada pengujian ini dilakukan pengambilan data terkhusus data yang berhubungan dengan proses deteksi kotak pada aplikasi. Seluruh keluaran proses deteksi kotak dapat dilihat pada Tabel 5.



Tabel 4. Hasil pengujian terhadap program utama

Nama	Panjang Kotak	Tinggi Kotak	Panjang Piksel/cm	Tinggi Piksel/cm
Rizqi	65	53	3.10	3.53
Zaki	64	52	3.05	3.47
Irfan	62	51	2.95	3.40
Yere	63	51	3.00	3.40
Faiz	62	52	2.95	3.47
Frenky	63	51	3.00	3.40
Cristian	63	52	3.00	3.47
Gabriel	63	52	3.00	3.47
Arif	62	50	2.95	3.33
Arinal	62	51	2.95	3.40
Trio	62	51	2.95	3.40
Amri	62	52	2.95	3.40
Rico	63	50	3.00	3.33
Hafidz	62	52	2.95	3.47
Ridho	62	50	2.95	3.33
David	64	52	3.05	3.47
Aldo	65	53	3.10	3.53
Absor	64	52	3.05	3.47
Fahmi	63	52	3.05	3.47
Batul	65	52	3.10	3.47
Bondan	62	51	2.95	3.40
Avi	63	51	3.00	3.40
Rizal	61	51	2.90	3.40
Ibek	63	52	3.00	3.47
Fauzan	62	52	2.95	3.47

Dari Tabel 4 disimpulkan bahwa adanya pengaruh proses deteksi kotak terhadap hasil pengujian menggunakan program utama. Dapat diketahui bahwa dari 25 objek pengujian, panjang kotak referensi memiliki nilai yang tidak tetap yakni dalam rentang 61 – 65 piksel dengan rata-rata 62.88 piksel serta tinggi kotak yang memiliki rentang 50 – 53 piksel dengan rata-rata 51.52 piksel. Nilai panjang dan tinggi tersebut mempengaruhi parameter konversi nilai sebenarnya. Parameter konversi yang didapatkan berdasarkan dimensi kotak yakni dengan nilai konversi panjang sekitar 2.90 – 3.10 piksel/cm dan nilai konversi tinggi sekitar 3.33 – 3.53 piksel/cm. Berdasarkan analisa pengujian, keluaran dari proses deteksi kotak tidak terlalu berpengaruh terhadap proses perhitungan dan klasifikasi ukuran baju dikarenakan error yang terdapat dalam proses deteksi berkisar 0.20 piksel/cm untuk panjang dan 0.20 piksel/cm untuk tinggi, sehingga menjadi bukti bahwa parameter keluaran proses deteksi kotak sudah sesuai dengan dimensi asli dari objek kotak.

#### 4. Kesimpulan

Pada proses pengujian berdasarkan variasi pencahayaan, variasi jarak dan implementasi program utama, objek kotak berhasil untuk terdeteksi dan terkomputasi dengan baik. Pencahayaan yang terlalu gelap mempengaruhi kualitas warna HSV sehingga menyebabkan tidak terdeteksinya kotak secara menyeluruh karena rentang warna HSV yang digunakan saat proses deteksi kotak tidak menangkap citra yang berwarna gelap sehingga dalam prosesnya hanya sebagian kotak yang terdeteksi. Deteksi kotak dengan pengujian variasi jarak

memiliki hasil deteksi objek yang tepat. Namun nilai parameter yang dihasilkan setelah proses deteksi variatif pada pengujian didapatkan parameter rasio yang memiliki error sekitar 0-2 untuk panjang dan lebar. Error terjadi dikarenakan semakin jauh objek yang dideteksi semakin sulit untuk menentukan panjang dan lebar dari citra dan faktor peletakkan objek referensi yang kurang rapi sehingga menyebabkan dalam proses bounding juga mengambil kontur kotak selain objek kotak tersebut. Pada implementasi dengan program utama keluaran dari proses deteksi kotak tidak terlalu berpengaruh terhadap proses perhitungan dan klasifikasi ukuran baju dikarenakan error yang terdapat dalam proses deteksi berkisar 0.20 piksel/cm untuk panjang dan 0.20 piksel/cm untuk tinggi sehingga menjadi bukti bahwa parameter keluaran proses deteksi kotak sudah sesuai dengan dimensi asli dari objek kotak.

Dari hasil Tugas Akhir ini, disarankan untuk dapat dilakukan penelitian lebih dalam lagi berdasarkan saran yang dapat dipertimbangkan untuk melanjutkan atau mengembangkan Tugas Akhir ini seperti:

1. Penggunaan resolusi yang lebih besar sehingga dalam pengolahan citra dan pengaplikasiannya mendapati hasil yang lebih baik untuk digunakan dalam perhitungan atau komputasi citra.
2. Pengujian dalam deteksi objek dilakukan dengan variasi pengujian lainnya seperti pengaruh posisi pengambilan citra, pengaruh variasi warna dan lain sebagainya.

#### Referensi

- [1]. A. P. Wijayanto, G. V. Valentinus, “Penerapan Teknologi Pengolahan Citra Digital dan Komputasi pada Pengukuran dan Pengujian berbagai Parameter Benang”, Laporan Tugas Akhir, Departemen Teknik Tekstil, Bandung, Indonesia: Politeknik STTT Bandung, 2012.
- [2]. Taufik Ramadhani dkk, “Pendeteksi Objek Menggunakan Algoritma HSV Berbasis Raspberry PI 3B”, Departemen Teknik Elektro, Semarang, Indonesia: Universitas Diponegoro, 2018.
- [3]. Priyanto Hidayatullah, “Pengolahan Citra Digital Teori dan Aplikasi Nyata”, Bandung, Indonesia: Informatika Bandung, 2017.
- [4]. A. T. Nugraha, Sopian Soim, Irawan Hadi, “Aplikasi Pendeteksi Tangan Dalam Ruang Warna YCbCr dan HSV pada Virtual Mouse”, Palembang, Indonesia : Politeknik Negeri Sriwijaya, 2016.
- [5]. Nur Khamdi, Muhammad Susantok, Piter Leopard, “Pendeteksi Objek Bola dengan Metode Color Filtering HSV pada Robot Soccer Humanoid”, Riau, Indonesia: Politeknik Caltex Riau, 2017.
- [6]. R.C. Gonzales, R. E. Woods, S. L. Eddins, “Digital Image Processing Using MATLAB”, New Jersey, USA: Pearson Prentice-Hall, 2004, ch. 1, sec. 1.2, hal. 2.
- [7]. Reinaldi Munir, “Pengolahan Citra Digital dengan Pendekatan Algoritmik”, Bandung, Indonesia: Informatika, 2004, ch 2, sec 2.3, hal. 25-26.



- [8]. R.C. Gonzales, R. E. Woods, S. L. Eddins, “Digital Image Processing Using MATLAB”, New Jersey, USA: Pearson Prentice-Hall, 2004, ch. 1, sec. 1.1, hal. 1.
- [9]. Reinaldi Munir, “Pengolahan Citra Digital dengan Pendekatan Algoritmik”, Bandung, Indonesia: Informatika, 2004, ch. 1, sec. 1.2, hal. 3.
- [10]. Python, "Python Tutorial", Python Software Foundation. [Online]. Available: <https://docs.python.org/3/download.html> [Accessed: 2-Agustus-2019].
- [11]. J. VanderPlas, A Whirlwind Tour of Python, O’Reilly Media, Inc., United States of America, 2016.
- [12]. M. Banzi, Getting Started with Arduino, O’Reilly Media, Inc., United States of America, 2009.
- [13]. OpenCV, “Miscellaneous Image Transformation”, OpenCv Team. [Online] Available: [https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous\\_transformations.html](https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html) [Accessed: 19-September-2019].
- [14]. Stef Maruch and Aazh Maruch, “Python for Dummies”, John Wiley & Sons 2006, ch. 1, sec. “Good uses of Python”, hal. 7-8.
- [15]. D. B. Beniz, A.M. Espindola, “Using Tkinter op Python to Create Graphical User Interface (GUI) for Scripts in LNLS” Proceedings of PCaPAC2016, Campina, Brazil: Brazilian Synchrotron Light Laboratory, August, 2016.
- [16]. OpenCV, “About”, OpenCv Team. [Online] Available: <https://opencv.org/about/> [Accessed: 2-Agustus-2019].