

SISTEM PENDETEKSI DAN PENGHITUNG OBJEK BERBASIS IMAGE PROCESSING

Detection System and Counting Object Based On Image Processing

Farid Sihabudin ¹, Suci Aulia, S.T, M.T. ², Dadan Nur Ramadan, S.Pd., M.T ^{3 1,2,3}

Prodi D3 Teknologi Telekomunikasi, Universitas Telkom

[1 faridshb17@gmail.com](mailto:faridshb17@gmail.com), [2 suciaulia@tass.telkomuniversity.ac.id](mailto:suciaulia@tass.telkomuniversity.ac.id), [3 dadan.nr@gmail.com](mailto:dadan.nr@gmail.com)

Abstract— This paper aims to develop an image processing system, based on computer vision technology. This application was developed to obtain information contained in images and project object detection and counters using a webcam camera. This system is able to detect and count object contained in the image with an optimal distance of 25cm. this application can simplify the process of counting and object automatically.

Intisari— Makalah ini bertujuan untuk mengembangkan suatu sistem pengolahan citra, berdasarkan teknologi *Computer Vision*. Aplikasi ini dikembangkan untuk memperoleh informasi yang terdapat pada citra dan memproyeksikan sistem pendeteksi dan penghitung objek menggunakan kamera *webcam*. Sistem ini mampu mendeteksi dan menghitung objek yang terdapat pada citra dengan jarak optimal 25cm. aplikasi ini dapat mempermudah proses penghitungan suatu objek secara otomatis.

Kata Kunci— *Background Substraction, Segmentation, Digital Image*

I. PENDAHULUAN

Computer Vision merupakan salah satu cabang ilmu yang ditawarkan untuk dapat mengatasi berbagai masalah dengan cara mengekstrak informasi dari citra yang disediakan dalam menyelesaikan suatu tugas. Dan dengan adanya kamera di jaman yang semakin maju ini memudahkan pengamatan suatu objek[1]. *Computer vision* berperan penting di jaman yang semakin maju ini karena bisa mendapatkan suatu informasi dari citra dan cara mendapatkan informasi tersebut digunakan metode *image processing*[7]. Pengolahan citra *digital (Digital Image Processing)* adalah sebuah ilmu yang mempelajari teknik mengolah sebuah citra.. Dan sebuah teknik yang dapat mempermudah aktivitas manusia di jaman sekarang ini, Contohnya beberapa perusahaan banyak yang menggunakan tenaga kerja manusia, contohnya penghitungan jumlah produk mereka yang masih dihitung secara manual dan cara tersebut terkadang dapat merugikan pada saat pabrikasi dimana bisa terjadi kesalahan pada saat menghitung jumlah produk karena kesalahan manusia.

Oleh karena itu pada Proyek Akhir ini akan dirancang sebuah sistem pendeteksi dan penghitungan secara otomatis agar dapat mempermudah dan meminimalisir setiap kesalahan pada saat pabrikasi.

Dan sistem ini menggunakan metode *image processing* serta prosesnya menggunakan *background subtraction*[2][9], dimana citra akan diubah kedalam *Black & White* untuk pemisah antara *background* dengan objek agar mudah menghitung jumlah objek pada citra, dengan bantuan implementasi menggunakan *software MATLAB (R2018a)*[5].

Dan harapan pada proyek akhir ini sistem memiliki performansi yang lebih baik, diantaranya tingkat akurasi, waktu proses dan kehandalan citra terhadap noise. Dan sistem yang akan saya buat dapat mempermudah penghitungan objek secara otomatis dalam pabrikasi khususnya pabrikasi yang masih menghitung benda secara manual.

II. DASAR TEORI

2.1 Pengolahan Citra

Citra adalah gambar bidang dua dimensi[9]. Dan citra memiliki fungsi yang berkesinambungan dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari sumber cahaya. Pantulan dari sumber cahaya ditangkap oleh alat optic, seperti pada kamera, mata manusia, dan lain-lain sehingga bayangan objek dalam citra dapat terekam. Citra sebagai keluaran dari suatu sistem perekam data bersifat :

1. Optik berupa foto
2. Analog berupa sinyal video seperti pada citra pada monitor televisi
3. Digital yang dapat langsung disimpan pada media penyimpanan magnetic

diproyeksikan secara tegak lurus pada bidang dua dimensi atau kertas gambar sesuai dengan ketentuan dari jenis proyeksi yang digunakan[4].

2.2 Pengolahan Citra dalam *Computer Vision*

Computer vision merupakan proses otomatis mengintegrasikan sebagian besar proses untuk persepsi visual, seperti akuisisi pengolahan, pengenalan dan membuat keputusan dalam citra, *Computer vision* mencoba untuk meniru cara kerja sistem visual manusia (*human vision*) yang sebenarnya sangat kompleks[7]. Untuk itu *computer vision* diharapkan dapat memiliki kemampuan tingkat tinggi sebagaimana *human visual*. Kemampuan itu diantaranya adalah:

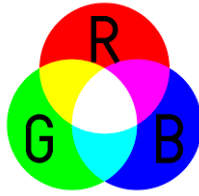
1. *Object Detection* - Apakah sebuah objek ada pada scene
2. *Recognition* - Menempatkan label pada objek
3. *Description* - Menugaskan property kepada objek
4. *3D Inference* - Menafsirkan adegan 3D dari 2D yang dilihat
5. *Interpreting motion* - Menafsirkan gerakan

Computer vision sering didefinisikan sebagai cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati. Cabang ilmu ini bersama dengan inteligensial semu (*Artificial intelligence*) akan mampu menghasilkan sistem intelijen visual (*Visual Intelligence System*).

2.3 RGB (*Red Green Blue*)

RGB sering disebut sebagai warna *additive*. Karena warna dihasilkan oleh cahaya, ada beberapa alat yang menggunakan warna RGB antara lain, *projector, TV, human vision*, kamera video dan kamera digital serta alat-alat yang menghasilkan cahaya. Dan proses pembentukan cahaya adalah dengan mencampur ketiga warna, dan skala intensitas warnanya dinyatakan dalam rentang nilai 0 sampai 255[4].

Ketika warna RGB memiliki intensitas sebanyak 255 maka akan menjadi warna putih, dan sebaliknya jika nilai intensitas 0 maka warna akan menjadi hitam. Sama halnya ketika didalam ruangan gelap tanpa cahaya maka disekitar kita hanya dapat melihat warna hitam.



Gambar 2 1 RGB



Gambar 2 5 Proses segmentasi

2.4 Grayscale

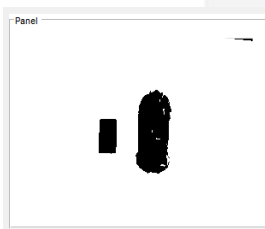
Grayscale adalah salah satu citra dimana nilai dari setiap pixel merupakan sampel tunggal. Citra yang ditampilkan adalah citra keabuan dimana intensitasnya berada pada interval 0 – 255, dan warna hitam (0), dan warna putih (255) [4][5].



Gambar 2 2 Proses Greyscale

2.5 Thresholding

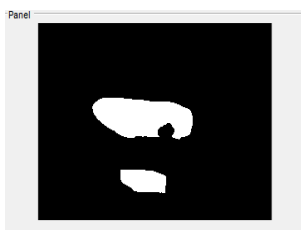
Prosesnya adalah mengubah citra grayscale menjadi citra biner. Secara matematis, dan thresholding dibagi menjadi dua yaitu thresholding global dan thresholding local[3], dimana jika global menggunakan metode dengan seluruh pixel pada citra dikonversi menjadi hitam putih dengan suatu nilai thresholding. Dan thresholding local nilai tidak bergantung pada citra saja, tetapi bergantung juga pada ciri khusus pada tiap pixel.



Gambar 2 3 Proses Thresholding

2.6 BW (Black and White)

Citra BW adalah keadaan sebuah citra yang hanya terdapat warna hitam dan putih, dimana citra BW digunakan untuk mempermudah sebuah tahap proses dalam pengolahan citra digital[6].



Gambar 2 4 Citra BW

2.7 Segmentation

Segmentasi merupakan tahapan pada proses analisis pada sebuah citra yang bertujuan untuk memperoleh informasi yang ada dalam citra tersebut dengan membagi ke dalam daerah terpisah dimana setiap daerah adalah homogen dan mengacu pada sebuah kriteria keseragaman yang jelas dan harus tepat agar informasi yang terkandung dalam citra dapat diterjemahkan dengan baik[10].

2.8 Metode Sobel

Metode Sobel merupakan metode yang menggunakan filter HPF (High Pass Filter) yang diberi satu angka nol penyangga, kelebihan dalam menggunakan metode ini adalah dapat mengurangi noise sebelum terjadinya perhitungan thresholding[8].

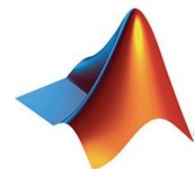
2.9 WEBCAM

Kamera Web atau yang lebih dikenal istilah Webcam adalah sebutan bagi kamera waktu nyata yang gambarnya bias dilihat melalui www (World Wide Web)[1]. Kamera web dapat diartikan sebagai kamera video kecil yang dihubungkan ke computer melalui port USB, port COM atau dengan jaringan Ethernet atau Wi-Fi. Adapun kamera webcam yang digunakan adalah Logitech c270 dengan spesifikasi kamera dibawah ini.

1. HD video calling (1280 X 720)
2. Video capture : Up to 1280 X 720 pixels
3. Photo : Up to 3.0 megapixels
4. Hi-Speed USB 2.0
5. Universal Clip fits leptop, LCD or CRT monitor
6. Price < Rp.300.000,-

2.10 MATLAB

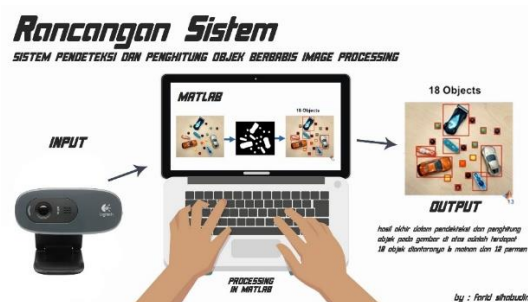
Sebuah software pemrograman untuk analisa matrix, desain control, identifikasi system dengan Teknik penggambaran grafik. Matlab merupakan singkatan dari Matrix Laboratory. Matlab dapat mengtegrasikan komputasi, pemrograman dan visualisasi untuk dapat digunakan secara mudah.



Gambar 2 6 Matlab

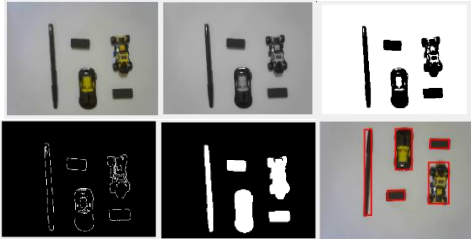
III. PERANCANGAN DAN IMPLEMENTASI SISTEM

Perancangan sistem pada penelitian ini menggunakan perangkat lunak MATLAB untuk analisis citra sehingga dapat mendeteksi dan menghitung jumlah objek pada citra. Dan konsep yang digunakan pada penelitian ini adalah morphological image processing. Dimana citra akan diambil secara real-time menggunakan webcam dan dikirimkan ke MATLAB untuk analisis agar dapat mendeteksi dan menghitung jumlah objek pada citra. Secara keseluruhan perancangan sistem penghitung objek diperlihatkan pada gambar.7

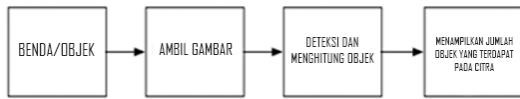


Gambar 3 1 Model perancangan sistem pendeteksi dan penghitung objek

Seperti pada Gambar 3 1 dijelaskan bahwa sistem mendeteksi objek dari citra yang didapatkan dari kamera webcam secara real time. Setelah itu sistem akan mendeteksi dan menghitung jumlah objek yang terdapat pada citra. Dan hasil citra yang masih RGB diubah kedalam *BW* (*Black and White*) untuk mempermudah dalam tahap proses lalu citra dikembalikan menjadi RGB serta hasil akhir akan menunjukkan *bounding box* dan jumlah objek yang terdapat pada citra, proses yang terjadi pada sistem dapat dilihat pada Gambar 3 2 sampai Gambar 3 4.



Gambar 3 2 Proses pendeteksi dan penghitung pada citra



Gambar 3 3 Blok diagram sistem penghitung objek

3.1 Analisis Kebutuhan Sistem

Spesifikasi perangkat keras (hardware) yang digunakan dalam pengerjaan proyek akhir ini adalah sebagai berikut:

1. Prosesor. Intel® Core™ i7 7700HQ Processor.
2. Sistem Operasi. Endless OS.
3. Chipset. Integrated Intel® CPU.
4. Memori. DDR4 2133 MHz SDRAM, OnBoard Memory 8 GB , 1x DIMM socket, DIMM Up to 16 G.
5. Display. 15.6" Auto Full HD (1920x1080)
6. Grafis. NVIDIA® GeForce® GTX 950M 2GB GDDR5.
7. Storage. - 1TB HDD 5400.
8. Drive Optik. Super-Multi DVD.
9. Kamera VGA Web Camera

3.1.1 Spesifikasi perangkat lunak

- 3.1.2 Windows 10
- 3.1.3 Matlab 2018a

3.2 Tahapan Perancangan

3.2.1 Monitoring objek dan mengambil citra

Merupakan proses memonitoring objek, setelah itu tekan tombol *capture* untuk mengambil sampel citra yang akan di analisis.



Gambar 3 4 Proses *monitoring* objek dan *capture*

3.2.2 Preprocessing

Pada tahap ini data citra masukan yang telah diambil sebelumnya akan memasuki tahap *preprocessing* dimana citra di ubah menjadi *Grayscale* agar bisa diproses ke tahap selanjutnya, lalu citra *Grayscale* di ubah menjadi citra biner (*Thresholding*).

1. RGB

Citra RGB merupakan citra yang tersusun dari tiga kanal warna yaitu merah, hijau, dan biru. Citra RGB terdiri dari 24 bit di

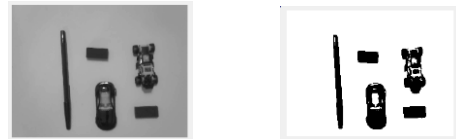
masing – masing kanal. Warna memiliki intensitas pixel 8 bit yang artinya memiliki variasi warna $2^8 = 256$ derajat warna (0 s.d 255). Setiap pixel pada citra RGB merupakan kombinasi dari nilai R, G dan, B.



Gambar 3 5 Citra RGB

2. Grayscale & Threshold

Gambar 3 6 Kiri merupakan hasil citra RGB yang diubah menjadi warna abu-abu agar dapat mempermudah untuk ketahap selanjutnya, karena citra grayscale hanya memiliki satu kanal warna yaitu abu-abu dengan nilai 256, setelah itu citra grayscale akan diubah menjadi citra biner agar dapat mempermudah proses pemisahan antara objek dengan latar, citra biner dibagi menjadi $2^1 = 2$ warna yaitu warna hitam yang dinyatakan 0 dan warna putih yang dinyatakan 1. Dan Gambar 3 6 Kanan merupakan hasil proses *background subtraction* yang telah di *threshold* bertujuan untuk menemukan objek pada latar dengan membandingkan sebuah citra yang memiliki nilai grayscale yang menunjukkan area berwarna dari input citra RGB.



Gambar 3 6 Grayscale(kiri), Threshold(kanan)

3.2.3 Processing

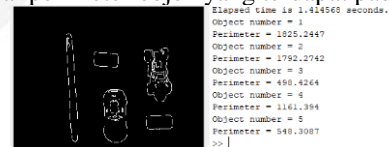
Dan pada tahap *processing* citra biner akan diubah menjadi citra BW agar mempermudah proses analisis sistem pada citra.



Gambar 3 7 Tahap proses dan hasil akhir

1. Deteksi Garis Tepi

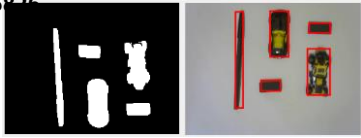
Pada tahap ini dilakukan operasi morfologi untuk mendapatkan boundaries (batas) dari citra biner. Kemudian dilakukan penghitungan keliling objek dengan cara menjumlahkan seluruh angka 1 (pixel warna putih) pada citra biner yang telah diproses dan keliling objek dihitung dalam satuan pixel. Gambar 3 8 kanan menampilkan nilai perimeter objek yang terdapat pada citra.



Gambar 3 6 Deteksi Garis Tepi (kiri), Keliling Objek (kanan)

2. Segmentation & Bounding Box

Segmentation adalah proses untuk dapat memperoleh hasil citra biner yang mempresentasikan luasan dari objek. Luas objek kemudian dihitung dengan cara menjumlahkan seluruh angka 1 (pixel warna putih) pada citra biner yang diperoleh, dan gambar 3 10 menampilkan hasil luas objek yang terdapat pada citra. Luas objek dihitung dalam satuan pixel. Setelah selesai mendapatkan nilai luas objek tahap selanjutnya diberikan bounding box untuk menandakan objek yang terdapat pada citra.



Gambar 3 9 Segmentation(kiri), Bounding Box(kanan)

```

Elapsed time is 6.729705 seconds.
Object number = 1
Area = 14014
Object number = 2
Area = 9307
Object number = 3
Area = 39536
Object number = 4
Area = 32895
Object number = 5
Area = 9991
>> |
    
```

Gambar 3 7 Luas Objek

3.3 Skenario Pengujian

Gambar 3 11 Menunjukkan skenario pengujian pada sistem, dimana objek di-monitoring menggunakan webcam secara real-time lalu diambil citra diam (capture) dan sistem akan memproses citra, setelah selesai sistem akan menampilkan bounding box yang menunjukkan jumlah objek yang terdapat pada citra dan hasil perhitungannya ditampilkan pada kotak edit text.



Gambar 3 11 Skenario Pengujian

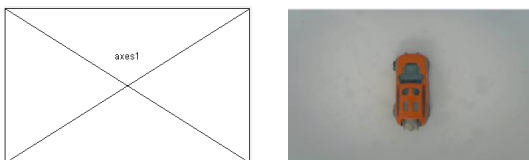
3.4 Realisasi Sistem

Dalam pengerjaan proyek akhir ini dilakukan realisasi terhadap sistem pendeteksi dan penghitung objek. Dibawah ini merupakan tampilan GUI dan beberapa fungsi kegunaannya sebagai berikut:



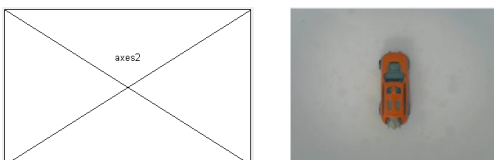
Gambar 3 12 Tampilan desain sistem pada GUI

Gambar 3 12 merupakan tampilan menu utama pada sistem pendeteksi dan penghitung objek. Pada GUI yang telah di buat terdapat 4 axes, 6 push button, 10 static text, 5 edit text.



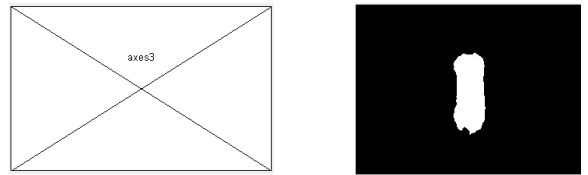
Gambar 3 13 Tampilan GUI Kamera Secara Real Time

Gambar 3 13 merupakan tampilan untuk menampilkan proses kamera secara real time, pada proses ini kamera melakukan pemantauan terhadap objek yang akan diujikan.



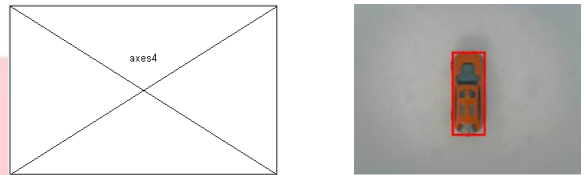
Gambar 3 14 Tampilan GUI Hasil Capture

Gambar 3 14 merupakan tampilan hasil capture dari pemantauan terhadap objek.



Gambar 3 15 Tampilan GUI Proses Grayscale, Threshold, dan Segmentation

Gambar 3 15 merupakan tampilan proses terhadap hasil citra yang telah di capture untuk di proses. Pertama citra RGB diubah menjadi ke abu-abuan (grayscale), lalu diubah menjadi citra biner (thresholding) dan terakhir di ubah menjadi citra hitam putih (BW), dan citra akan diperhalus sehingga noise yang ikut terdeteksi akan dihilangkan.



Gambar 3 16 Tampilan GUI Proses Bounding Box

Gambar 3 16 merupakan hasil citra yang telah di analisis dan tahap selanjutnya adalah pemberian bounding box untuk mendeteksi dan menghitung jumlah objek yang terdapat pada citra.



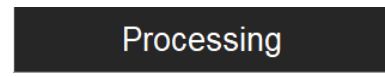
Gambar 3 8 Tampilan Push Button 1

Gambar 3 17 merupakan tampilan push button 1 berfungsi menghidupkan kamera untuk melakukan pemantauan terhadap objek yang akan diujikan.



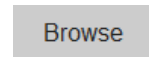
Gambar 3 9 Tampilan Push Button 2

Gambar 3 18 merupakan tampilan push button 2 berfungsi menangkap citra yang telah dipantau oleh kamera.



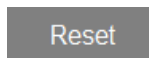
Gambar 3 19 Tampilan Push Button 3

Gambar 3 19 merupakan tampilan push button 3 berfungsi untuk memproses hasil capture atau browse. Pada proses ini akan menampilkan hasil segmentasi pada axes 3, dan akan menampilkan hasil akhir citra yang telah diubah kembali menjadi citra RGB lalu diberi bounding box pada axes 4.



Gambar 3 10 Tampilan Push Button 4

Gambar 3 20 merupakan tampilan push button 4 berfungsi untuk membuka file objek yang akan dianalisis.



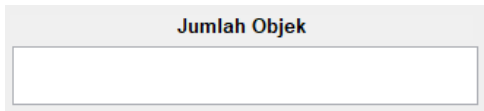
Gambar 3 11 Tampilan Push Button 5

Gambar 3 21 merupakan tampilan push button 5 berfungsi me-reset semua axes dan edit text.



Gambar 3 12 Tampilan Push Button 6

Gambar 3 22 merupakan tampilan *push button* berfungsi untuk keluar dari tampilan GUI.



Gambar 3 13 Tampilan *Edit Text*

Gambar 3 23 merupakan tampilan *edit text* berfungsi untuk menampilkan hasil jumlah objek(keseluruhan).



Gambar 3 14 Tampilan *Static Text*

Gambar 3 24 merupakan tampilan *static text* berfungsi untuk memberikan nama atau keterangan.

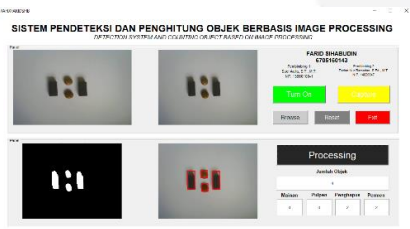
IV. PENGUJIAN DAN ANALISA SISTEM

4.1 Pengujian Terhadap Latar

Pada proyek akhir ini dilakukan beberapa pengujian dalam latar untuk mendapatkan hasil akhir yang diinginkan. Dan saya telah mencoba menggunakan 2 latar untuk pengujian pada sistem ini yaitu latar hitam dan latar putih, dibawah ini hasil yang telah dilakukan pengujian pada tingkat keberhasilan mendeteksi dan menghitung objek yang terdapat pada citra. Berikut merupakan contoh pengujian terhadap latar hitam dan latar putih:



Gambar 4 1 Contoh pengujian terhadap latar hitam



Gambar 4 2 Contoh pengujian terhadap latar putih



Gambar 4 3 pengujian terhadap latar merah



Gambar 4 4 pengujian terhadap latar hijau



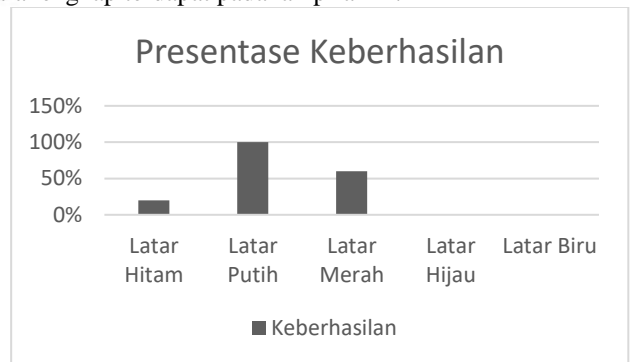
Gambar 4 5 pengujian terhadap latar biru

Gambar 4 1 – 4 5 merupakan hasil pengujian terhadap latar menggunakan objek dan terdapat kesalahan pada proses deteksi dimana beberapa objek yang tidak dapat dideteksi pada latar hitam, merah, hijau, dan biru sedangkan latar putih sistem dapat mendeteksi dengan baik objek yang terdapat pada citra, dan bukti pengujian terhadap latar secara lengkap terdapat pada lampiran A.

Tabel 1 Pengujian Terhadap Latar

No	Latar	Rata – Rata Delay (detik)	Presentase Keberhasilan
1	Latar Hitam	1.16	20%
2	Latar Putih	1.22	100%
3	Latar Merah	1.23	60%
4	Latar Hijau	1.14	0%
5	Latar Biru	1.14	0%

Tabel 1 menunjukan data pengujian berdasarkan latar hitam dan latar putih. Latar hitam mendapatkan presentase keberhasilan sebesar 20% dari 5 kali percobaan dengan rata – rata delay 1.16 detik, latar merah mendapatkan presentase keberhasilan 60% dari 5 kali percobaan dengan rata – rata delay 1.22 detik lalu latar hijau dan latar biru mendapatkan presentasi keberhasilan 0% dari 5 kali percobaan dengan rata – rata delay 1.14 detik. Sedangkan untuk latar putih mendapatkan presentase keberhasilan 100% dari 5 kali percobaan dengan rata – rata delay 1.22 detik. Karena warna objek sangat berpengaruh dengan latar, jika warna latar dengan objek sama maka objek tidak dapat terdeteksi, dan bukti pengujian terhadap latar secara lengkap terdapat pada lampiran A.



Gambar 4 6 Grafik Pengujian Terhadap Latar

4.2 Pengujian Terhadap Jarak

Pada proyek akhir ini telah dilakukan pengujian terhadap jarak antara kamera dengan objek, jarak yang digunakan pada pengujian ini bervariasi mulai dari 15cm, 20cm, 25cm, 30cm, dan 35cm. Parameter yang dianalisis adalah delay dan tingkat akurasi. Berikut merupakan contoh pengujian terhadap jarak:



Gambar 4 7 Contoh pengujian terhadap jarak 20cm



Gambar 4 8 Contoh pengujian terhadap jarak 25cm



Gambar 4 9 Contoh pengujian terhadap jarak 30cm



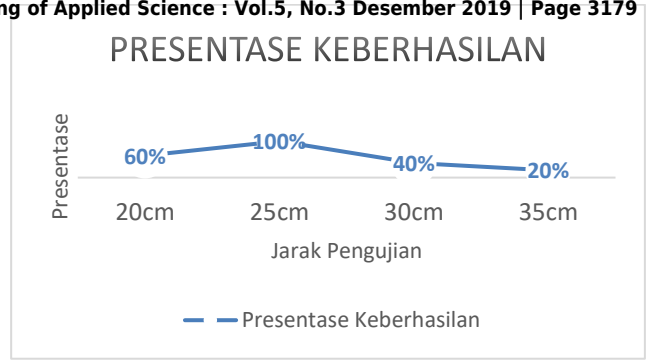
Gambar 4 10 Contoh pengujian terhadap jarak 35cm

Gambar 4 4 – 4 7 merupakan hasil pengujian perhitungan terhadap jarak 20cm, 30cm dan 35cm terdapat kesalahan dimana objek tidak dapat terdeteksi, sedangkan jarak 25cm semua objek dapat terdeteksi dengan baik, dan bukti pengujian terhadap jarak secara lengkap terdapat pada lampiran B.

Tabel 2 Pengujian Terhadap Jarak dan Delay

No	Jarak	Presentase Keberhasilan	Delay (detik)
1	20cm	60%	1.38
2	25cm	100%	1.31
3	30cm	40%	1.29
4	35cm	20%	1.13
Rata - Rata		50%	1.28

Setelah dilakukan pengujian pada sistem mendapatkan presentase hasil pengujian 55% dari 20 kali pengujian dan rata – rata delay 1.28 second. Dari pengujian tersebut didapatkan hasil terbaik pada jarak 25cm dengan presentase keberhasilan 100% dan rata – rata delay yang didapatkan adalah 1.31 second.



Gambar 4 11 Grafik performansi sistem pendeteksi terhadap jarak

4.3 Pengujian Terhadap Sudut

Pada proyek akhir ini telah dilakukan pengujian terhadap sudut kamera dengan objek, sudut yang digunakan pada pengujian ini bervariasi mulai dari 70°, 90°, dan 120°, pengukuran berdasarkan sudut diukur dari bidang horizontal terhadap kamera sebagai sudut 0°. Parameter yang dianalisis adalah delay dan tingkat akurasi. Berikut merupakan contoh pengujian terhadap sudut:



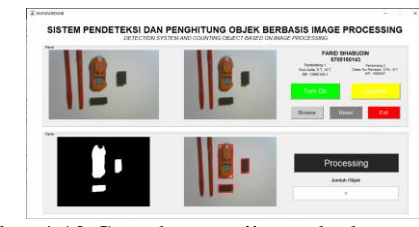
Gambar 4 9 Contoh pengujian terhadap sudut 40°



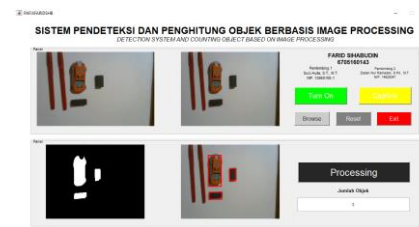
Gambar 4 10 Contoh pengujian terhadap sudut 60°



Gambar 4 11 Contoh pengujian terhadap sudut 90°



Gambar 4 12 Contoh pengujian terhadap sudut 120°



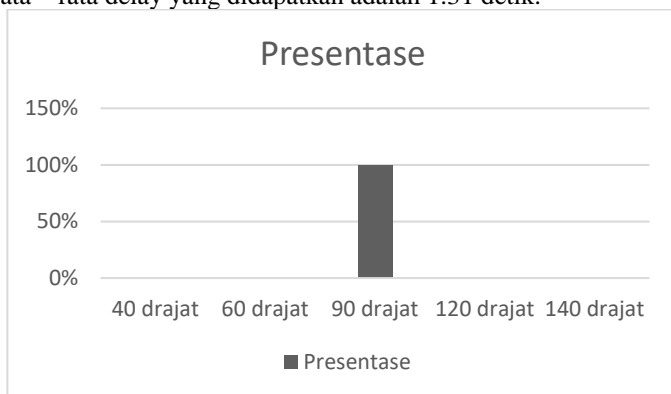
Gambar 4 13 Contoh pengujian terhadap sudut 140°

Gambar 4 9 – 4 13 merupakan hasil pengujian sudut kamera terhadap objek pada sudut 40°, 60°, 90°, 120° dan 140° terdapat kesalahan dimana objek tidak dapat terdeteksi dan sudut 90° dapat mendeteksi objek dengan baik tanpa kesalahan, bukti pengujian terhadap sudut secara lengka terdapat pada lampiran C.

Tabel 3 Pengujian Terhadap Sudut

No	Sudut	Citra	Hasil Pengujian	Delay (detik)
1	60°	Data 1	Gagal	1.22
2	60°	Data 2	Gagal	1.34
3	70°	Data 1	Gagal	1.43
4	70°	Data 2	Gagal	1.31
5	90°	Data 1	Berhasil	1.28
6	90°	Data 2	Berhasil	1.34
7	120°	Data 1	Gagal	1.15
8	120°	Data 2	Gagal	1.16
9	140°	Data 1	Gagal	1.15
10	140°	Data 2	Gagal	1.15
Rata - Rata			20%	1.27

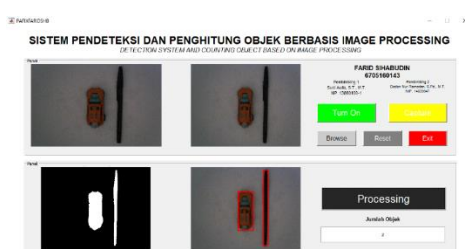
Setelah dilakukan pengujian pada sistem mendapatkan presentase hasil pengujian 20% dari 10 kali pengujian dan rata – rata delay 1.27 detik. Dari pengujian tersebut didapatkan hasil terbaik pada sudut 90° dengan presentase keberhasilan 100% dan rata – rata delay yang didapatkan adalah 1.31 detik.



Gambar 4 14 Grafik performansi sistem pendeteksi terhadap jarak

4.4 Pengujian Terhadap Intensitas Cahaya

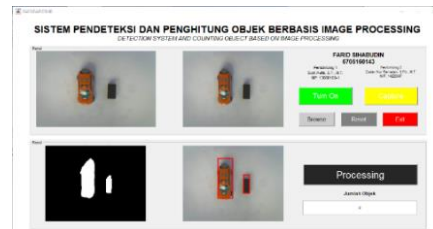
Pada proyek akhir ini telah dilakukan pengujian terhadap intensitas cahaya terhadap objek, intensitas cahaya yang digunakan pada pengujian ini bervariasi mulai dari 0lx, 25lx, 92lx, 150lx, dan 255lx. Parameter yang dianalisis adalah delay dan tingkat akurasi. Berikut merupakan contoh pengujian terhadap intensitas cahaya:



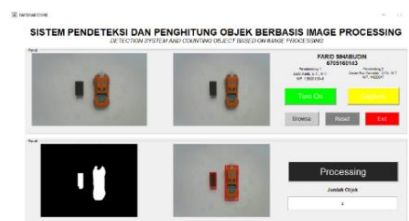
Gambar 4 15 Contoh pengujian terhadap lux 0



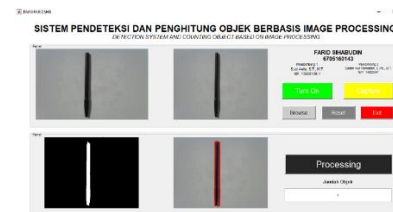
Gambar 4 16 Contoh pengujian terhadap lux 25



Gambar 4 17 Contoh pengujian terhadap lux 92



Gambar 4 18 Contoh pengujian terhadap lux 255



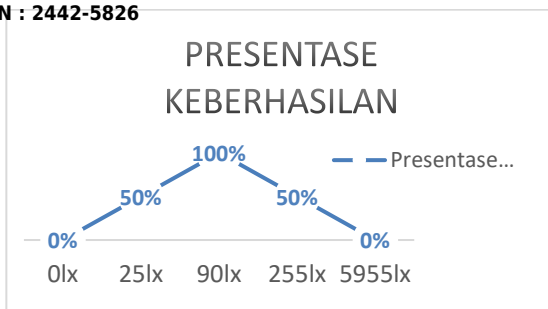
Gambar 4 19 Contoh pengujian terhadap lux 5955

Gambar 4 15 – 4 19 merupakan contoh hasil pengujian terhadap intensitas cahaya(lux).

Tabel 4 Pengujian Terhadap Intensitas Cahaya

No	Intensitas cahaya(lx)	Citra	Hasil Pengujian	Delay (detik)
1	0	Data 1	Gagal	1.28
2	0	Data 2	Gagal	1.29
3	25	Data 1	Gagal	1.25
4	25	Data 2	Berhasil	1.33
5	92	Data 1	Berhasil	1.33
6	92	Data 2	Berhasil	1.31
7	255	Data 1	Berhasil	1.29
8	255	Data 2	Gagal	1.23
9	5955	Data 1	Gagal	1.24
10	5955	Data 2	Gagal	1.21
Rata – Rata			40%	1.27

Setelah dilakukan pengujian pada sistem mendapatkan presentase hasil pengujian 40% dari 10 kali pengujian dan rata – rata delay 1.27 detik. Dari pengujian tersebut didapatkan hasil terbaik pada lux 92 dengan presentase keberhasilan 100% , dan rata – rata delay yang didapatkan adalah 1.32 detik.



Gambar 4.12 Grafik performansi sistem pendeteksi terhadap intensitas cahaya

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan maka didapati kesimpulan yaitu:

1. Keakuratan sistem dalam mendeteksi dan menghitung objek mendapatkan hasil presentase 55% dari 20 kali percobaan pada jarak yang bervariasi dari 20-35cm dimana rata – rata delay 1.28 detik.
2. Keakuratan sistem dalam mendeteksi dan menghitung objek mendapatkan hasil presentase 30% dari 10 kali percobaan pada latar hitam dan rata – rata delay 1.21 detik.
3. Keakuratan sistem dalam mendeteksi dan menghitung objek mendapatkan hasil presentase 100% dari 10 kali percobaan pada latar putih dan rata – rata delay 1.31 detik.
4. Sistem mendapatkan hasil terbaik pada jarak 25cm dimana presentase keberhasilan 100% dari 5 kali percobaan dan rata – rata delay 1.31 detik.
5. Sistem mendapatkan hasil terbaik pada sudut 90° dimana presentase keberhasilan 100% dari 3 kali percobaan dan rata – rata delay 1.32 detik.
6. Sistem mendapatkan hasil terbaik pada lux 92 dimana presentase keberhasilan 100% dari 2 kali percobaan dan rata – rata delay 1.36 detik.

5.2 Saran

Pada proyek akhir ini terdapat beberapa kekurangan pada aplikasi sehingga dapat dilakukan pengembangan untuk proyek berikutnya, berikut adalah saran pengembangan dari aplikasi pada proyek akhir ini:

- a. Aplikasi dapat memonitoring menggunakan kamera handphone.
- b. Dapat mendeteksi dan menghitung objek yang bergerak.

DAFTAR PUSTAKA

- [1] Andre, J. A. (2016). Visual Studio 2012 Source Code. *Jurnal Teknologi Dan Sistem Informasi UNIVRAB*. 1(2), 46–58. <https://doi.org/10.1039/c2cc38049h>
- [2] Ardianto, E., & Hadikurniawati, W. (2013). *Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video*. *Jurnal Teknologi Informasi DINAMIK*. 18(2), 91–100.
- [3] Aynurrohmah, M., Sunyoto, A., & Kunci, K. (2011). *PENGHITUNG JUMLAH MOBIL MENGGUNAKAN PENGOLAHAN CITRA DIGITAL DENGAN INPUT VIDEO DIGITAL*. *Jurnal Dasi*. 12(3), 2–6.
- [4] Indriyani, L., Susanto, W., & Riana, D. (2017). Aplikasi Matlab Pada Pengukuran Diameter buah Jeruk Keprok. *IJCIT (Indonesian Journal on Computer and Information Technology)*. 2(1), 46–52.
- [5] Kusumanto, R. D., & Tomponu, A. N. (2011).

- [6] Rijal, Y., & Ariefianto, R. D. (2008). Deteksi Wajah Berbasis Segmentasi Model Warna Menggunakan Template Matching pada Objek Bergerak. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2008*. 2008(January 2008), 35–42.
- [7] Sukatmi, S. (2017). Perbandingan Deteksi Tepi Citra Digital dengan Menggunakan Metode Prewitt, Sobel dan Canny. *KOPERTIP : Jurnal Ilmiah Manajemen Informatika Dan Komputer*. 1(1), 1–4. <https://doi.org/10.32485/kopertip.v1i1.3>
- [8] Umam, K., & Negara, B. S. (2016). *Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi*. *Jurnal CoreIT*. 2(2), 31–40.
- [9] Sindar, A., & Sinaga, R. M. (2017). IMPLEMENTASI THRESHODING PADA SEGMENTASI CITRA DIGITAL. *Jurnal Mantik Penusa*. 1(2), 48–51.

