

## **Sistem Pemeriksaan Jumlah Fisik Botol Kaca Berbasis Citra pada Proses Pengemasan Manufaktur Industri**

Susetyo Bagas Bhaskoro, Hadi Supriyanto, Syamsul Falah

Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung

Email: bagas@polman-bandung.ac.id

---

<b>Informasi Artikel:</b>	<b>ABSTRAK</b>
<p><i>Received:</i> 30 Januari 2023</p> <p><i>Accepted:</i> 01 April 2024</p> <p><i>Available:</i> 28 Juni 2024</p>	<p><i>Machine vision</i> merupakan teknologi yang biasa digunakan pada industri modern untuk analisis dan inspeksi otonom berbasis citra. <i>Machine vision</i> membantu proses analisis dan inspeksi produk di industri lebih cepat dibandingkan dengan analisis dan inspeksi manual. Penelitian ini menerapkan <i>Machine vision</i> pada sistem identifikasi jumlah produk berbasis pengolahan citra pada proses pengemasan industri manufaktur dengan menggunakan algoritma YOLOv4 dan evaluasi sistem confusion matrix. Hasil dari identifikasi disimpan di database dan ditampilkan pada website agar memudahkan proses monitoring. Sistem ini telah melakukan beberapa pengujian terutama pengujian fungsi utama sistem yaitu perhitungan produk, dilakukan 10x percobaan. Kemudian, pengujian variasi intensitas cahaya dengan range 20 – 225lux dan variasi ketinggian dengan range 48 – 68 cm dengan masing – masing pengujian sebanyak 10x percobaan. Dari pengujian yang telah dilakukan diterapkan evaluasi confusion matrix dan menghasilkan akurasi dan presisi sebesar 100% dan error sebesar 0%. Kecepatan komputasi rata – rata dari sistem ini sebesar 6.95 FPS dengan bantuan CUDA.</p>

---

<b>Kata Kunci:</b>	<b>ABSTRACT</b>
<p><i>Machine vision</i> Object Counter Object Detection YOLOv4</p>	<p><i>Machine vision is a technology commonly used in modern industry for image-based autonomous analysis and inspection. Machine vision helps the process of product analysis and inspection in the industry faster than manual analysis and inspection. This study applies Machine vision to an image processing-based product number identification system in the manufacturing industry packing process using the YOLOv4 algorithm and evaluation of the confusion matrix system. The results of the identification are stored in a database and displayed on the website to facilitate the monitoring process. This system has carried out several tests, especially testing the main function of the system, namely product calculations, carried out 10x experiments. Then, testing variations in light intensity with a range of 20 – 225 lux and variations in height with a range of 48 – 68 cm with 10 trials each. From the tests that have been carried out, an evaluation of the confusion matrix is applied and produces an accuracy and precision of 100% and an error of 0%. The average computing speed of this system is 6.95 FPS with the help of CUDA.</i></p>

## 1 PENDAHULUAN

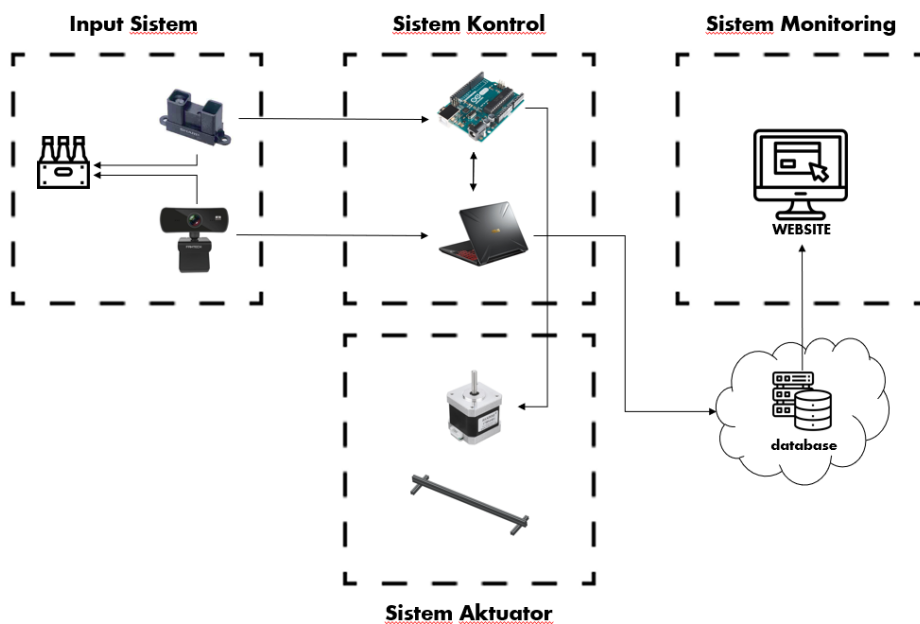
*Machine vision* merupakan teknologi yang biasa digunakan pada industri modern untuk analisis dan inspeksi otonom berbasis citra [1]. Menurut Harry pada artikel [2] menyatakan *Machine vision* dapat diterapkan untuk deteksi kehadiran, pengukuran otomatis, pembacaan barcode, membandingkan warna, deteksi cacat produk, perhitungan produk otomatis dan lain – lain. *Machine vision* sendiri sering disamakan dengan computer vision [3] dan memiliki sistem yang terdiri dari beberapa proses secara berurutan, diantaranya; akuisisi citra, digitalisasi, filtering, pengumpulan informasi, dan eksekusi keputusan [4].

Pada perkembangannya, *Machine vision* telah diterapkan di berbagai bidang industri seperti industri makanan, industri tekstil, dan industri keramik [5] dikarenakan penggunaan *Machine vision* membuat analisis dan inspeksi lebih cepat dibandingkan dengan analisis dan inspeksi manual. Menurut M. A. Khalim dkk. pada penelitian [6] menjelaskan bahwa pada salah satu perusahaan, perhitungan produk masih dilakukan secara manual yang tidak berbanding sama dengan hasil produksi yang lebih besar sehingga perhitungan produk tertumpuk. Oleh karena itu, teknologi menjadi solusi untuk menyelesaikan masalah tersebut dan salah satu teknologi yang cocok adalah *Machine vision*.

Dengan penjelasan di atas, pada penelitian ini penulis telah membuat sistem identifikasi jumlah produk berbasis pengolahan citra pada proses pengepakan industri manufaktur dengan algoritma pendeteksian objek YOLOv4 dan evaluasi confusion matrix. Sistem ini diharapkan dapat membantu sebuah industri dalam identifikasi jumlah fisik sebuah produk dengan cepat dibandingkan identifikasi manual.

## 2 METODOLOGI PENELITIAN

### 2.1 Gambaran Umum Sistem

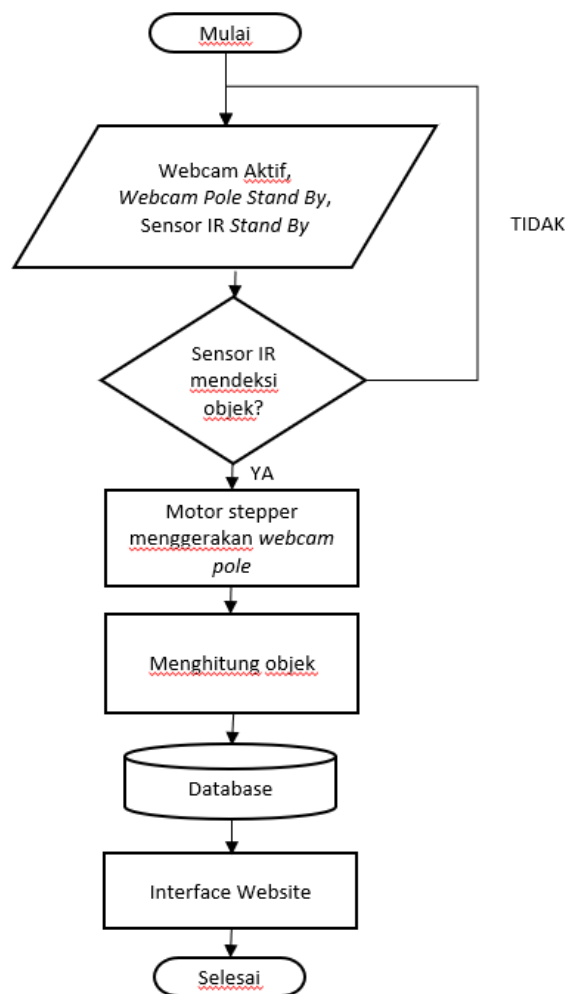


Gambar 1. Gambaran umum sistem

Pada Gambar 1. menunjukkan gambaran umum dari sistem yang akan dibuat, yang terdiri dari 4 bagian sistem; input sistem, sistem kontrol, sistem aktuator, dan sistem *monitoring*. Langkah awal sistem yang akan dibuat yaitu sensor jarak mendeteksi adanya objek pada zona pendeteksian dalam jarak sekian sentimeter apabila objek berada pada jarak tertentu akan menjalankan motor stepper yang akan menggerakkan *webcam pole* pada lintasan. Ketika kamera *webcam* melewati objek yang berada pada zona pendeteksian, proses deteksi objek berlangsung dan perhitungan jumlah botol kaca dalam *crate*. Kemudian, dari hasil deteksi dan perhitungan jumlah botol kaca dalam *crate* akan disimpan dalam *database* yang selanjutnya ditampilkan dalam *website* secara *real-time*.

### 2.1.1 Diagram Alir Sistem

Adapun diagram alir dari sistem yang telah dibuat adalah sebagai berikut:



Gambar 2. Diagram alir sistem

Pada Gambar 2. menampilkan diagram alir dari sistem yang akan dibuat. Sistem diawali dengan kondisi *webcam* aktif, *pole* tempat mendudukan *webcam* dan sensor jarak dalam kondisi *stand by*. Kemudian, apabila sensor jarak mendeteksi objek pada zona pendeteksian maka sensor akan memberi sinyal pada mikrokontroler dan mengaktifkan motor stepper, apabila sensor jarak tidak mendeteksi maka periksa kembali kondisi awal. Motor stepper yang

telah diaktifkan akan menggerakkan *webcam pole* dan *webcam* akan mendeteksi dan menghitung objek. Setelah objek telah dideteksi dan dihitung jumlahnya, data akan disimpan dalam *database* yang kemudian ditampilkan pada halaman *website*.

## 2.2 Perancangan Mekanikal Sistem

Perancangan mekanikal sistem dibutuhkan agar keseluruhan sistem bekerja, pada perancangan ini penulis ingin menampilkan spesifikasi dari rancangan mekanikal sistem.



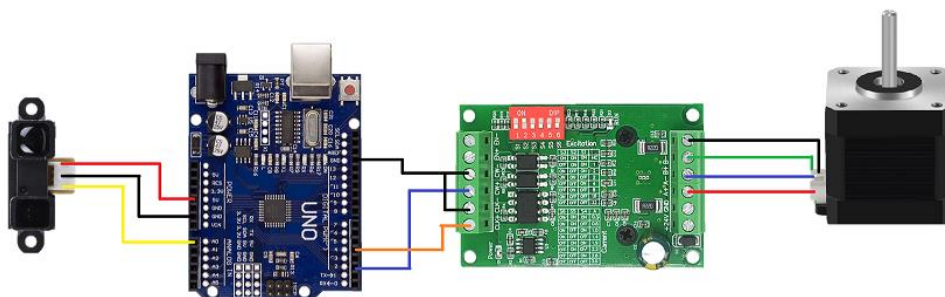
Gambar 3. Rancangan mekanikal sistem

Spesifikasi dari rancangan mekanikal sistem ditampilkan pada Tabel 1.

Tabel 1. Spesifikasi rancangan mekanikal

Jenis	Keterangan
Dimensi Lintasan Pole Webcam	750 x 20 x 40 mm
Dimensi Pole Webcam	Vertically 750x20x20 mm
	Horizontally 200x20x20mm
Dimensi Penampang Pole Webcam	75 x 75 x 3 mm
Penggerak Penampang Pole Webcam	<i>Roller-Bearing</i> , dan <i>Timing Belt</i>
Aktuator	Motor Stepper NEMA-17 HS4401

## 2.3 Perancangan Elektrikal Sitem



Gambar 4. Rancangan elektrikal sistem

Setelah perancangan mekanikal sistem dibuat, penulis membuat rancangan elektrikal sistem dari sistem yang akan dibuat seperti pada Gambar 4. yang menampilkan motor stepper diberi sinyal dari driver TB6560, sumber driver dari power supply 24V dan GND ke ground, CW+ driver terhubung ke pin D2 dan CLK+ driver terhubung ke pin D4 dari Arduino sedangkan CW- dan CLK- terhubung ke GND Arduino. Untuk sensor Sharp GP2Y0A02YK0F sinyal ke pin AO, GND ke GND, dan Vcc ke 5V pada Arduino.

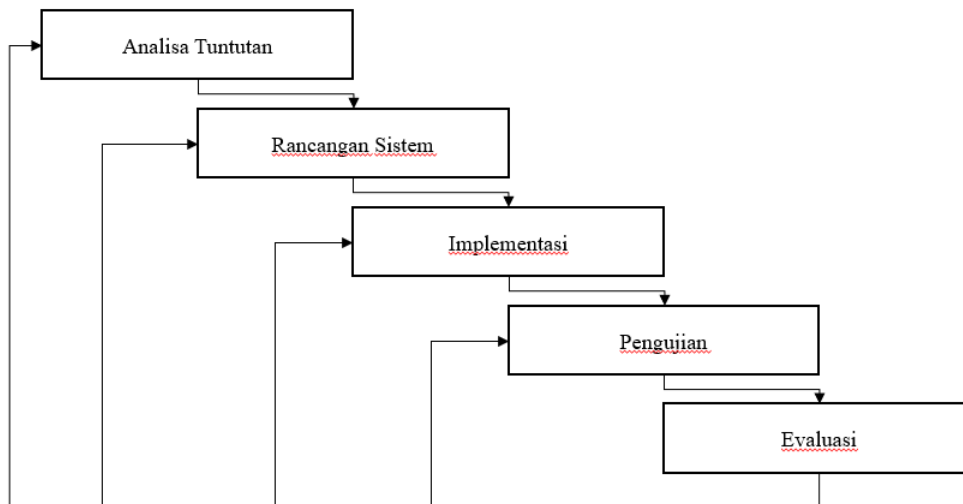
Untuk spesifikasi rancangan elektrikal terlampir pada Tabel 2. sebagai berikut:

Tabel 2. Spesifikasi rancangan elektrikal sistem

Jenis	Keterangan
Mikrokontroler	Arduino Uno R3
Driver	TB6560 3A Upgrade
Komunikasi Data	Serial
Jumlah Input I/O Digital	2 Buah
Jumlah Input I/O Analog	1 Buah

#### 2.4 Penerapan Metode *Waterfall* untuk Perancangan Program

Metode *waterfall* [7] diterapkan untuk perancangan *program*. Pada Gambar 3. menunjukkan tahapan – tahapan dari penerapan metode *waterfall*. Penulis memilih metode *waterfall* dikarenakan metode ini memudahkan untuk mengontrol pengembangan model dikarenakan proses dari pengembangan modelnya *one by one* seperti air terjun yang dimana proses satu harus lengkap dan selesai terlebih dahulu sebelum lanjut ke proses selanjutnya sehingga dapat meminimalisir kesalahan yang mungkin terjadi.



Gambar 5. Tahapan – tahapan penerapan metode penelitian waterfall [8]

Pada Gambar 5. telah ditunjukkan tahapan – tahapan dari penerapan metode penelitian waterfall sebanyak lima tahapan. Pertama yaitu tahapan analisis tuntutan, pada tahap ini penulis melakukan pengumpulan data – data yang dibutuhkan untuk memenuhi tuntutan sistem yang akan dibuat. Kedua yaitu tahapan rancangan sistem, setelah mengumpulkan data – data yang dibutuhkan untuk membuat sistem, pada tahapan ini penulis membuat rancangan sistem sesuai dengan tuntutan sistem yang sudah dikumpulkan untuk memenuhi tujuan yang diinginkan. Ketiga yaitu tahapan implementasi, rancangan yang sudah terbentuk kemudian dibuat dalam bentuk program agar dapat diimplementasikan pada sistem. Keempat yaitu pengujian, setelah program dibuat, penulis melakukan percobaan dari program tersebut dan melihat output dari sistem apakah sudah sesuai dengan tujuan yang diinginkan atau belum. Terakhir yaitu evaluasi untuk memastikan hasil dari pengujian agar sistem berjalan sesuai dengan tuntutan dan tujuan yang diinginkan [8].

### 2.4.1 Analisa Tuntutan Sistem

Tahapan awal dari metode *waterfall* yaitu tahapan analisa tuntutan. Pada tahap ini, penulis melakukan pengumpulan data – data yang dibutuhkan untuk memenuhi tuntutan sistem yang akan dibuat.

Tuntutan tersebut salah satunya *output* seperti apa yang diinginkan dari sistem tersebut. Dari sistem yang penulis buat dan teliti, *output* utama yang diinginkan adalah mendeteksi tutup botol kaca dan menghitung jumlah botol kaca dalam satu *crate* botol berukuran 6 x 4 (isi 24 botol). Untuk mencapai *output* tersebut, penulis membutuhkan data pendukung seperti algoritma deteksi objek, logika penghitungan objek, ukuran dimensi *crate*, jarak *webcam* dengan *crate*, jarak alat dengan *crate*.

Setelah sistem dapat mendeteksi dan menghitung jumlah botol kaca sesuai dengan tujuan yang diinginkan, kemudian data *output* tersebut disimpan dalam *database*. Untuk mencapainya, penulis membutuhkan data pendukung seperti jumlah botol kaca yang terdapat dalam *crate*, jumlah botol kaca yang tidak terdapat dalam *crate*, tabel *database*.

Terakhir, setelah data *output* tersimpan dalam *database*, data *output* yang tersimpan ditampilkan pada halaman *website* agar memudahkan proses *monitoring*. Untuk mencapai tujuan tersebut, penulis membutuhkan data pendukung seperti *framework website* yang cocok dan ringan, program *export* data dari dalam *database* ke dalam *framework database* yang telah dibuat.

Tabel 3. berisikan kumpulan data yang dibutuhkan untuk membuat sistem penghitung botol kaca dalam satu *crate* berbasis citra agar memudahkan pelaku industri dalam proses perhitungan botol kaca dalam satu *crate* sehingga mengurangi *cost* dan waktu yang tidak perlu.

Tabel 3. Tabel data – data tuntutan sistem

No	Tuntutan Sistem	Output Sistem
1	Model algoritma deteksi objek	Klasifikasi objek <i>crate</i> botol dalam <i>crate</i>
2	Logika penghitungan objek	Menghitung jumlah objek botol kaca dalam <i>crate</i>
3	Data jumlah botol kaca dalam <i>crate</i>	Objek yang telah terdeteksi dan terhitung disimpan dalam <i>database</i>
4	Tabel <i>database</i> , <i>framework website</i>	Menampilkan data yang tersimpan dalam <i>database</i> pada halaman <i>website</i>

### 2.4.2 Perancangan Sistem

Perancangan sistem dilakukan dengan merencanakan model algoritma deteksi objek, algoritma penghitungan objek, tabel *database*, *framework website* dan komponen lain yang dibutuhkan untuk menjalankan sistem sesuai dengan tujuan yang diinginkan. Pada Tabel 4. Menampilkan analisis dari rancangan sistem yang akan dibuat.

Tabel 4. Perancangan sistem

No	Perancangan Sistem	Metode
1	Rancangan Model Algoritma Deteksi Objek	Menggunakan YOLO ( <i>You Only Look Once</i> )
2	Rancangan Logika Penghitungan Objek	Menggunakan fungsi <code>len()</code> dari Python
3	Rancangan Tabel Database Manager	Menggunakan MySQL

4	Rancangan <i>website</i>	Menggunakan framework CodeIgniter
---	--------------------------	-----------------------------------

### 2.4.3 Pengujian Rancangan Sistem

Setelah melakukan analisis tuntutan, perancangan, selanjutnya penulis ingin melakukan pengujian pada rancangan sistem yang telah dibuat untuk mengetahui kemampuan dari sistem apakah sudah dapat memenuhi tuntutan dari tujuan yang ingin dicapai. Pada tahapan pengujian rancangan sistem, penulis membuat skema pengujian agar pengujian terstruktur.

Tabel 5. Skema pengujian rancangan sistem

No	Pengujian Sistem	Fungsi
1	Pengujian Rancangan Model Algoritma Deteksi Objek	Mengamati hasil deteksi dari model algoritma YOLOv4 mengikuti dataset yang sudah di- <i>training</i>
2	Pengujian Rancangan Logika Penghitung Objek	Mengamati hasil dari program Python penghitung objek mengikuti jumlah objek yang terdeteksi.
3	Pengujian Rancangan Tabel Database Manager	Mengamati hasil dari perhitungan objek apakah sudah bisa input ke dalam <i>database</i> atau belum.
4	Pengujian Rancangan <i>website</i>	Mengamati hasil dari perngujian integrasi tabel <i>database</i> dengan <i>website</i> , dengan menampilkan data dalam <i>database</i> pada <i>website</i>

### 2.4.4 Evaluasi Sistem

Evaluasi sistem dilakukan untuk melihat hasil dari pengujian apakah sudah memenuhi tuntutan sistem yang diinginkan agar alat dapat bekerja dengan baik dan optimal. Untuk mengevaluasi sistem penulis menggunakan metode *Confusion Matrix* untuk menghitung error dan mengamati hasil presisi dan akurasi.

*Confusion matrix* adalah metode evaluasi dengan cara membuat tabel yang menyatakan klasifikasi jumlah data nilai aktual dan jumlah data nilai prediksi [9]. Contoh penerapan metode *confusion matrix* pada klasifikasi bilangan biner ditunjukkan pada Tabel 6.

Tabel 6. Contoh *confusion matrix*

		Nilai Prediksi	
		1	0
Nilai Aktual	1	TP	FN
	0	FP	TN

keterangan:

TP (True Positive) = Jumlah hasil prediksi angka 1 yang aktual angka 1.

TN (True Negative) = Jumlah hasil prediksi angka 0 yang aktual angka 0.

FP (False Positive) = Jumlah hasil prediksi angka 1 yang aktual angka 0. error I

FN (False Negative) = Jumlah hasil prediksi angka 0 yang aktual angka 1. error II (berbahaya)

Berdasarkan hasil dari klasifikasi *confusion matrix* dapat menghasilkan perhitungan *accuracy*, *precision*, dan *error*.

Dengan persamaan secara berurutan sebagai berikut:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$
- $Precision = \frac{TP}{TP+FP} \times 100\%$
- $Error = \frac{FP+FN}{TP+TN+FP+FN} \times 100\%$

Penjelasan:

1. Nilai *accuracy* adalah rasio prediksi benar baik positif maupun negative terhadap keseluruhan data.
2. Nilai *precision* adalah rasio nilai TP prediksi terhadap keseluruhan nilai prediksi positif.
3. Nilai *error* adalah nilai kesalahan prediksi terhadap nilai aktual.

### 3 HASIL

#### 3.1 Pengujian Rancangan Elektrikal dan Mekanikal Sistem

Pada pengujian rancangan elektrikal dan mekanikal, penulis ingin mengamati hasil integrasi dari pembacaan sensor dengan mikrokontroller dan motor stepper, dikarenakan pengujian ini merupakan dasar dari sistem alat, apabila rancangan elektrikal dan mekanikal tidak memenuhi tuntutan sistem yang dibutuhkan maka keseluruhan sistem alat tidak akan berjalan maksimal. Tabel 7. merupakan tabel data dari hasil pengujian rancangan elektrikal dan mekanikal.

Tabel 7. Hasil pengujian rancangan elektrikal dan mekanikal

No	Jarak Pembacaan Sensor	Mikrokontroller	Motor Stepper
1	5cm	'No object'	Tidak Bergerak
2	10cm	'No object'	Tidak Bergerak
3	15cm	'No object'	Tidak Bergerak
4	20cm	'Object inside counting zone'	Bergerak
5	25cm	'No object'	Tidak Bergerak

Mengacu pada Tabel 7. hasil pengujian elektrikal dan mekanikal menampilkan pembacaan sensor jarak pada lima titik jarak; 5, 10, 15, 20, dan 25cm. Pada jarak 5, 10, dan 15cm di depan sensor jarak, mikrokontroller pada Arduino IDE memberikan *output* 'No object' mengikuti hasil program sehingga motor stepper tidak bergerak karena pembacaan objek belum pada zona perhitungan. Pada jarak 20cm di depan sensor jarak, mikrokontroller pada Arduino IDE menampilkan *output* 'Object inside counting zone' karena pembacaan objek berada pada range zona perhitungan sehingga membuat motor stepper bergerak sejauh ±30cm. Sedangkan pada jarak 25 cm di depan sensor jarak, mikrokontroller pada Arduino IDE menampilkan *output* 'No object' lagi karena pembacaan objek sudah di melewati range zona perhitungan dan motor stepper tidak bergerak.



### 3.2 Pengujian Model Algoritma Deteksi Objek

Pengujian ini diawali dengan pembuatan program. Langkah awal dalam pembuatan program deteksi objek adalah dengan memuat konfigurasi, weights, dan list kelas untuk kebutuhan model algoritma YOLOv4. Selanjutnya, merubah setelan komputasi program menggunakan CUDA menjadi GPU agar proses pendeteksian objek *frame by frame* lebih cepat dibanding dengan CPU. Lalu, membuat looping agar *frame* selalu terbuka dikarenakan video merupakan kumpulan dari banyak *frame* sehingga pendeteksian terjadi setiap *frame*. Berikutnya, pembuatan *bounding boxes* pada setiap deteksi objek dengan kelas yang diperkirakan, apabila kelas dari deteksi objek merupakan '0' yang berarti 'bottle crate' maka *bounding box* berwarna hijau, apabila kelas dari deteksi objek merupakan '1' yang berarti 'bottlecap' maka *bounding box*-nya berwarna biru. Setelah program dijalankan akan membuka jendela 'Output'. Pengujian model algoritma deteksi objek dibagi menjadi beberapa bagian, diantaranya; pengujian deteksi dan klasifikasi objek, pengujian nilai confidence deteksi objek, pengujian deteksi objek dengan kondisi khusus, dan pengujian kecepatan komputasi deteksi objek.

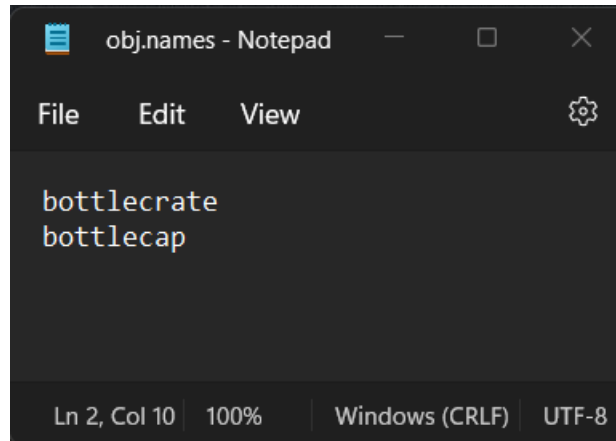
#### 3.2.1 Pengujian Deteksi dan Klasifikasi Objek

Pada Tabel 8. menampilkan hasil dari pengujian deteksi dan klasifikasi objek.

Tabel 8. Hasil pengujian deteksi dan klasifikasi

No	Hasil Deteksi	Keterangan
1		Deteksi Objek Tutup Botol
2		Deteksi Objek <i>Crate</i>
3		Klasifikasi Objek

Hasil dari pengujian deteksi dan klasifikasi objek dapat dinyatakan berhasil. Mengacu pada file *obj.names*, kelas '0' berarti 'bottlecrate' dengan *bounding box* hijau dan kelas '1' berarti 'bottlecap' dengan *bounding box* biru. File *obj.names* merupakan list dari kelas yang sudah di-*training*.



Gambar 6. Isi file obj.names

### 3.2.2 Pengujian Nilai Confidence Deteksi Objek

Deteksi objek yang dihitung nilai confidence-nya hanya pada deteksi objek kelas 'bottlecap'. Nilai confidence pada program harus dikali 100 untuk mendapatkan nilai persentase.

Tabel 9. Hasil pengujian deteksi dan klasifikasi

No.	Output	Rata – Rata Confidence(%)	No.	Output	Rata – Rata Confidence(%)
1		99.99	6		100
2		100	7		99.98
3		99.98	8		99.98
4		99.99	9		100
5		100	10		100

Total rata – rata	99.93
-------------------	-------

Dengan 10x percobaan didapatkan total rata – rata nilai confidence adalah 99.93%. Dengan begitu, pengujian nilai confidence pada deteksi objek tutup botol dapat dikatakan sangat baik.

### 3.2.3 Pengujian Kecepatan Komputasi Deteksi Objek

FPS adalah jumlah banyaknya *frame* yang dimuat dalam satu detik. Proses deteksi objek pada video memperhatikan banyaknya FPS dikarenakan video merupakan kumpulan *frame* yang disatukan, sehingga apabila FPS dari video tinggi maka hasil pemutaran video akan berjalan lancar dan tidak mengganggu proses perhitungan objek karena posisi dari titik x pada *bounding box* per *frame* sangat mempengaruhi hasil perhitungan pada sistem alat ini.

Tabel 10. Hasil pengujian kecepatan komputasi deteksi objek

Detik	FPS GPU	FPS CPU	Detik	FPS GPU	FPS CPU
1	5.06	1.15	51	6.76	1.11
2	7.13	1.24	52	7.25	1.12
3	6.87	1.4	53	6.82	1.12
4	7.28	1.47	54	7.14	1.12
5	5.2	1.38	55	7.04	1.09
6	6.13	1.32	56	6.99	1.08
7	6.6	1.39	57	7.13	1.11
8	7.09	1.31	58	6.94	1.08
9	7.13	1.31	59	6.89	1.09
10	7.19	1.34	60	6.9	1.07
11	7.25	1.31	61	7.28	1.09
12	7.17	1.28	62	6.8	1.08
13	7.54	1.33	63	7.2	1.09
14	6.81	1.35	64	6.83	1.09
15	7.14	1.36	65	7.15	1.08
16	6.89	1.39	66	7.06	1.07
17	7.13	1.36	67	6.94	1.07
18	7.57	1.34	68	6.98	1.14
19	6.97	1.33	69	7.18	1.07
20	7.04	1.32	70	7.08	1.07
21	6.87	1.35	71	6.83	1.08
22	7.35	1.31	72	6.94	1.08
23	6.8	1.33	73	6.8	1.08
24	6.89	1.25	74	6.94	1.08
25	7.39	1.29	75	7.06	1.08
26	6.77	1.29	76	6.92	1.05
27	7.18	1.24	77	7.12	1.07
28	6.99	1.2	78	6.79	1.06
29	6.99	1.2	79	7.12	1.07
30	6.94	1.16	80	7.04	1.09
Average FPS GPU				6.95	
Average FPS CPU				1.19	

Pada Tabel 10. menampilkan perbandingan banyaknya FPS dan rata – rata FPS pada proses komputasi deteksi objek antara menggunakan GPU dengan bantuan CUDA dan hanya menggunakan CPU dari 80 detik. Rata – rata FPS proses komputasi dengan GPU sebanyak 6.95 FPS dan dengan CPU sebanyak 1.19 FPS.

### 3.3 Pengujian Logika Penghitung Objek

Pada pengujian logika penghitung objek terdapat delapan sub-pengujian, diantaranya pengujian perhitungan produk, pengujian perhitungan produk beda posisi, pengujian perhitungan produk dengan tutup botol silver, pengujian perhitungan produk dengan tutup botol gold, pengujian perhitungan produk dengan tutup botol merah, pengujian perhitungan produk dengan tutup botol hijau, pengujian perhitungan produk dengan tanpa tutup botol, dan pengujian perhitungan produk dengan botol terbalik.

#### 3.3.1 Pengujian Perhitungan Produk

Pengujian ini merupakan fungsi utama dari sistem alat. Perhitungan objek tutup botol dilakukan sebagai ganti perhitungan jumlah botol dalam *crate* maka apabila ingin menghitung jumlah botol kaca diharapkan selalu menempatkan tutup botolnya, pada hasil pengujian ini juga menerapkan evaluasi *confusion matrix* untuk melihat seberapa besar *accuracy*, *precision*, dan *error*.

Tabel 11. Hasil pengujian perhitungan objek tutup botol

No.	Manual		Otomatis		Keterangan			
	Isi	Kurang	Isi	Kurang	TP	TN	FP	FN
1	2	22	2	22	2	22	0	0
2	9	15	9	15	9	15	0	0
3	12	12	12	12	12	12	0	0
4	13	11	13	11	13	11	0	0
5	14	10	14	10	14	10	0	0
6	17	7	17	7	17	7	0	0
7	19	5	19	5	19	5	0	0
8	20	4	20	4	20	4	0	0
9	22	2	22	2	22	2	0	0
10	24	0	24	0	24	0	0	0
Total					152	88	0	0

Pada Tabel 11. telah ditampilkan hasil perhitungan objek tutup botol dalam *crate* sebanyak 10x percobaan. Setelah data pengujian terkumpul, selanjutnya menerapkan evaluasi sistem *confusion matrix*. Untuk evaluasi *confusion matrix*, data yang sudah terkumpul dimasukkan ke dalam persamaan di bawah ini:

$$\text{Accuracy} = \frac{152 + 88}{152 + 88 + 0 + 0} \times 100 = 100\% \quad (1)$$

$$\text{Precision} = \frac{152}{152 + 0} \times 100\% = 100\% \quad (2)$$

$$\text{Error} = \frac{0 + 0}{152 + 88 + 0 + 0} \times 100\% = 0\% \quad (3)$$

Dari hasil perhitungan di atas dapat diambil informasi nilai *accuracy* sebesar 100%, *precision* sebesar 100%, dan error sebesar 0%. Dengan begitu, hasil pengujian untuk perhitungan objek tutup botol sudah sangat memuaskan mengingat pengujian ini untuk penerapan fungsi utama sistem alat.

### 3.3.2 Pengujian Perhitungan Produk Beda Posisi

Pada pengujian ini sama seperti pengujian sebelumnya namun dengan posisi yang berbeda dan isi yang sama. Pengujian ini dilakukan untuk membuktikan perhitungan produk tidak hanya pada posisi tertentu namun bisa pada setiap posisi. Hasil dari pengujian ini juga menerapkan evaluasi *confusion matrix* untuk melihat seberapa besar *accuracy*, *precision*, dan *error*.

Tabel 12. Hasil pengujian perhitungan objek tutup botol

No.	Manual		Otomatis		Keterangan			
	Isi	Kurang	Isi	Kurang	TP	TN	FP	FN
1	2	22	2	22	2	22	0	0
2	9	15	9	15	9	15	0	0
3	12	12	12	12	12	12	0	0
4	13	11	13	11	13	11	0	0
5	14	10	14	10	14	10	0	0
6	17	7	17	7	17	7	0	0
7	19	5	19	5	19	5	0	0
8	20	4	20	4	20	4	0	0
9	22	2	22	2	22	2	0	0
10	24	0	24	0	24	0	0	0
Total					152	88	0	0

Pada Tabel 12. telah ditampilkan hasil perhitungan objek tutup botol dengan posisi yang berbeda sebanyak 10x percobaan. Setelah data pengujian terkumpul, selanjutnya menerapkan evaluasi sistem *confusion matrix*. Untuk evaluasi *confusion matrix*, data yang sudah terkumpul dimasukkan ke dalam persamaan di bawah ini:

$$\text{Accuracy} = \frac{152 + 88}{152 + 88 + 0 + 0} \times 100 = 100\% \quad (4)$$

$$\text{Precision} = \frac{152}{152 + 0} \times 100\% = 100\% \quad (5)$$

$$\text{Error} = \frac{0 + 0}{152 + 88 + 0 + 0} \times 100\% = 0\% \quad (6)$$

Dari hasil perhitungan di atas dapat diambil informasi nilai *accuracy* sebesar 100%, *precision* sebesar 100%, dan error sebesar 0%. Dengan begitu, hasil pengujian untuk perhitungan objek tutup botol dengan posisi yang berbeda sudah dapat disebut sangat baik.

### 3.3.3 Pengujian Perhitungan Produk dengan Tanpa Tutup Botol

Pada pengujian ini diterapkan kondisi khusus dimana sebagian botol kaca dipasangkan tutup botol dan sebagian lainnya tidak dipasangkan tutup botol. Tujuan dari pengujian ini agar mengetahui seberapa besar persentase error dari sistem alat dengan adanya ambiguitas dalam rentang dataset. Untuk evaluasi pada hasil pengujian ini penulis menerapkan metode evaluasi *confusion matrix* untuk melihat kinerja sistem alat pada perhitungan objek tutup botol dengan kondisi khusus.

Tabel 13. Hasil pengujian perhitungan objek tutup botol

No.	Manual		Otomatis		Keterangan			
	DTB*	TTB*	DTB	TTB	TP	TN	FP	FN
1	2	22	2	22	2	22	0	0
2	4	20	4	20	4	20	0	0
3	6	18	6	18	6	18	0	0
4	8	16	8	16	8	16	0	0
5	10	14	10	14	10	14	0	0
6	11	13	11	13	11	13	0	0
7	13	11	13	11	13	11	0	0
8	17	7	17	7	17	7	0	0
9	19	5	19	5	19	5	0	0
10	21	3	21	3	21	3	0	0
Total					111	129	0	0

\*DTB berarti dengan tutup botol, TTB berarti tanpa tutup botol.

Dengan data yang sudah terlampir pada Tabel IV.13. dapat diketahui nilai *accuracy*, *precision*, dan *error* dari model algoritma deteksi objek YOLOv4 pada kondisi khusus dengan evaluasi *confusion matrix*. Untuk evaluasi *confusion matrix*, data – data yang sudah terkumpul dimasukkan ke dalam persamaan:

$$\text{Accuracy} = \frac{111 + 129}{111 + 129 + 0 + 0} \times 100 = 100\% \quad (7)$$

$$\text{Precision} = \frac{111}{111 + 0} \times 100\% = 100\% \quad (8)$$

$$\text{Error} = \frac{0 + 0}{111 + 129 + 0 + 0} \times 100\% = 0\% \quad (9)$$

Dari hasil perhitungan di atas, didapatkan nilai *accuracy* sebesar 100%, *precision* 100%, dan *error* sebesar 0%. Hasil pengujian ini dapat dikatakan sangat baik.

### 3.3.4 Pengujian Perhitungan Produk dengan Botol Terbalik

Pada pengujian ini, sama seperti pengujian sebelumnya, diterapkan kondisi khusus dimana sebagian botol kaca diletakkan terbalik. Tujuan dari pengujian ini agar mengetahui seberapa

besar persentase error dari sistem alat dengan adanya ambigu dalam rentang dataset. Untuk evaluasi pada hasil pengujian ini penulis menerapkan metode evaluasi *confusion matrix* untuk melihat kinerja sistem alat pada perhitungan objek tutup botol dengan kondisi khusus.

Tabel 14. Hasil pengujian perhitungan objek tutup botol

No.	Manual		Otomatis		Keterangan			
	TT*	T*	TT	T	TP	TN	FP	FN
1	4	20	4	20	4	20	0	0
2	6	18	6	18	6	18	0	0
3	8	16	8	16	8	16	0	0
4	15	9	15	9	15	9	0	0
5	16	8	16	8	16	8	0	0
6	18	6	18	6	18	6	0	0
7	20	4	20	4	20	4	0	0
8	21	3	21	3	21	3	0	0
9	22	2	22	2	22	2	0	0
10	23	1	23	1	23	1	0	0
Total					153	87	0	0

\*TT berarti tidak terbalik, T berarti terbalik.

Dengan data yang sudah terlampir pada Tabel 14. dapat diketahui nilai *accuracy*, *precision*, dan *error* dari model algoritma deteksi objek YOLOv4 pada kondisi khusus dengan evaluasi *confusion matrix*. Untuk evaluasi *confusion matrix*, data – data yang sudah terkumpul dimasukkan ke dalam persamaan:

$$\text{Accuracy} = \frac{153 + 87}{153 + 87 + 0 + 0} \times 100 = 100\% \quad (10)$$

$$\text{Precision} = \frac{153}{153 + 0} \times 100\% = 100\% \quad (11)$$

$$\text{Error} = \frac{0 + 0}{153 + 87 + 0 + 0} \times 100\% = 0\% \quad (12)$$

Dari hasil perhitungan di atas, didapatkan nilai *accuracy* sebesar 100%, *precision* 100%, dan *error* sebesar 0%. Hasil pengujian ini dapat dikatakan sangat baik.

#### 4 KESIMPULAN

Dari hasil pengujian, dapat disimpulkan bahwa Sistem alat yang diuji telah dapat mendeteksi dan mengklasifikasikan objek antara kelas 'bottle crate' dan 'bottlecap' dengan nilai confidence di atas 99%. Rata – rata kecepatan komputasi dari sistem alat dengan bantuan CUDA sebanyak 6.95 FPS. Fungsi utama sistem alat yang diuji telah dapat menghitung dengan akurasi dan presisi 100% serta error dari hasil perhitungan sebesar 0%. Juga dari kondisi –

kondisi khusus dari pengujian alat telah dilakukan dan mendapatkan presisi dan akurasi 100% serta error 0%.

## 5 REFERENSI

- [1] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, "Exploring impact and features of *Machine vision* for progressive industry 4.0 culture," *Sensors Int.*, vol. 3, no. October 2021, p. 100132, 2022, doi: 10.1016/j.sintl.2021.100132.
- [2] "Machine vision Systems Capabilities, Uses, And Applications." <https://www.lacconveyors.co.uk/machine-vision-systems-provided/> (accessed Jan. 10, 2023).
- [3] "What Is *Machine vision* and How Does it Work?" <https://www.techtarget.com/searchenterpriseai/definition/machine-vision-computer-vision> (accessed Jan. 10, 2023).
- [4] J. Beyerer, F. P. León, and C. Frese, *Machine vision: Automated visual inspection: Theory, practice and applications*. 2015. doi: 10.1007/978-3-662-47794-6.
- [5] V. Nandini, R. Deepak Vishal, C. Arun Prakash, and S. Aishwarya, "A Review on Applications of *Machine vision* Systems in Industries," *Indian J. Sci. Technol.*, vol. 9, no. 48, 2016, doi: 10.17485/ijst/2016/v9i48/108433.
- [6] M. A. Khalim, A. H. Kurniawan, and C. Supriadi, "Sistem informasi penghitungan hasil produk berbasis internet of things," vol. 2, no. 1, 2021.
- [7] M. Waterfall, "Penerapan Metodologi Waterfall Pada Rancangan," vol. 11, no. 1, pp. 1–12.
- [8] F. Mahardika, K. A. Purwanto, and D. I. Surya Saputra, "Implementasi Metode Waterfall pada Proses Digitalisasi Citra Analog," *VOLT J. Ilm. Pendidik. Tek. Elektro*, vol. 2, no. 1, p. 63, 2017, doi: 10.30870/volt.v2i1.948.
- [9] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 5, no. 2, pp. 697–711, 2021.