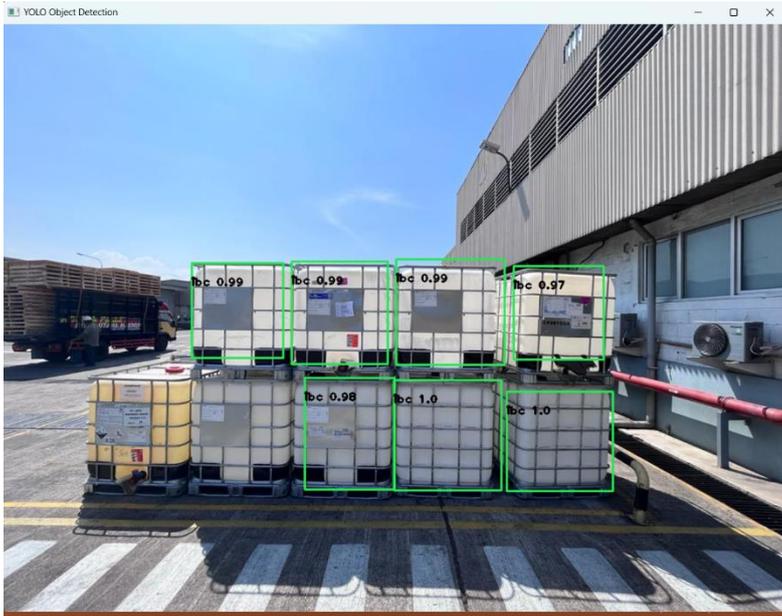


## 5. ANALISA PENGUJIAN

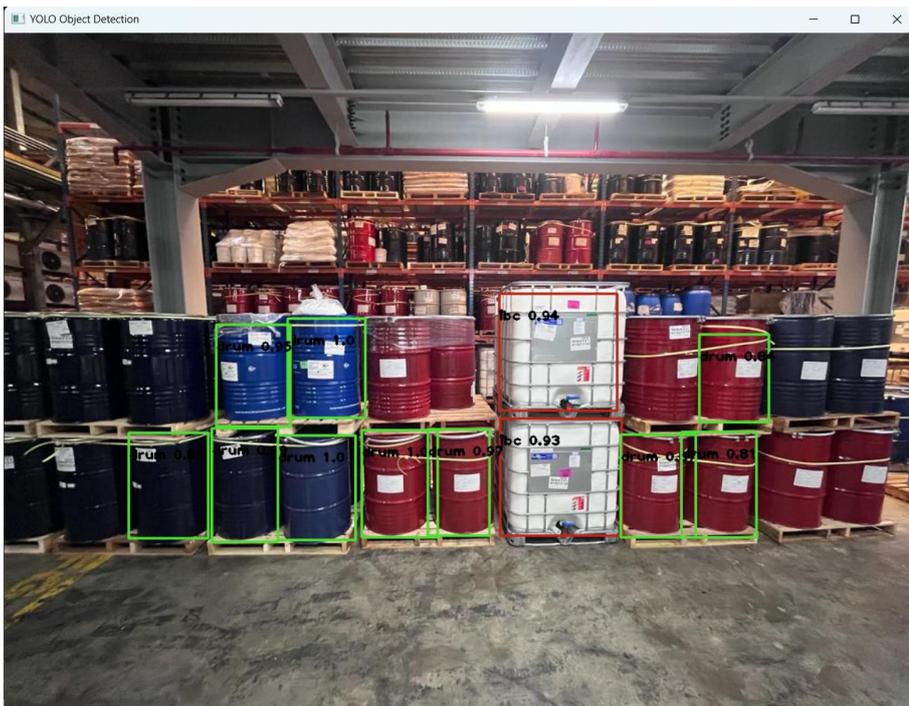
Bab ini membahas mengenai hasil dari pengujian sistem yang telah dilakukan dalam skripsi ini. Beberapa aspek yang akan dibahas meliputi evaluasi hasil program dalam mendeteksi objek-objek yang ada dalam pabrik melalui kamera yang terpasang pada sisi-sisi forklift, output notifikasi yang akan dihasilkan oleh sistem berdasarkan estimasi jarak yang telah dihitung, dan juga FPS yang dihasilkan oleh program saat berjalan. Pada pengujian kali ini, spesifikasi kamera yang digunakan adalah 720p dengan 30fps

### 5.1 Hasil Program dalam Mendeteksi Objek

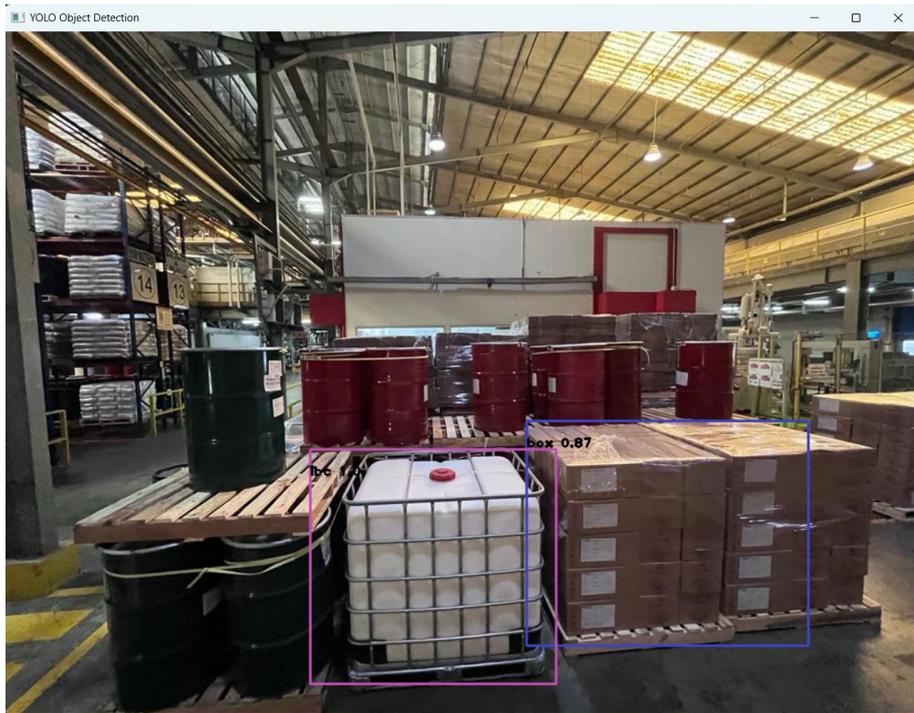
Program untuk mendeteksi objek pada pabrik mengacu pada penggunaan dataset yang berisi 6 kelas. Gambar-gambar pada dataset diantaranya terdiri dari *box*, *cone*, *drum*, *IBC*, *forklift*, dan *orang*. Proses deteksi objek menggunakan model YOLOv3, yang merupakan salah satu dari model machine learning yang cukup terkenal dalam hal deteksi objek. Prosesnya dimulai dengan memuat model YOLOv3 yang sudah ditrain dengan dataset sebelumnya, yaitu *yolov3-custom\_final.weights*. Setelah model dimuat, video input akan diambil frame per frame. Setiap frame tersebut akan dideteksi menggunakan model yolo. Deteksi objek merupakan hal yang harus diuji terlebih dahulu, dikarenakan pada sistem ini dibutuhkan *width* dari *bounding box* yang didapat guna untuk melakukan perhitungan estimasi jarak antara objek dengan kamera. Beberapa hasil dari pembacaan tersebut terdapat pada gambar 5.1 hingga gambar 5.5.



Gambar 5.1 Hasil Deteksi IBC / Totebin



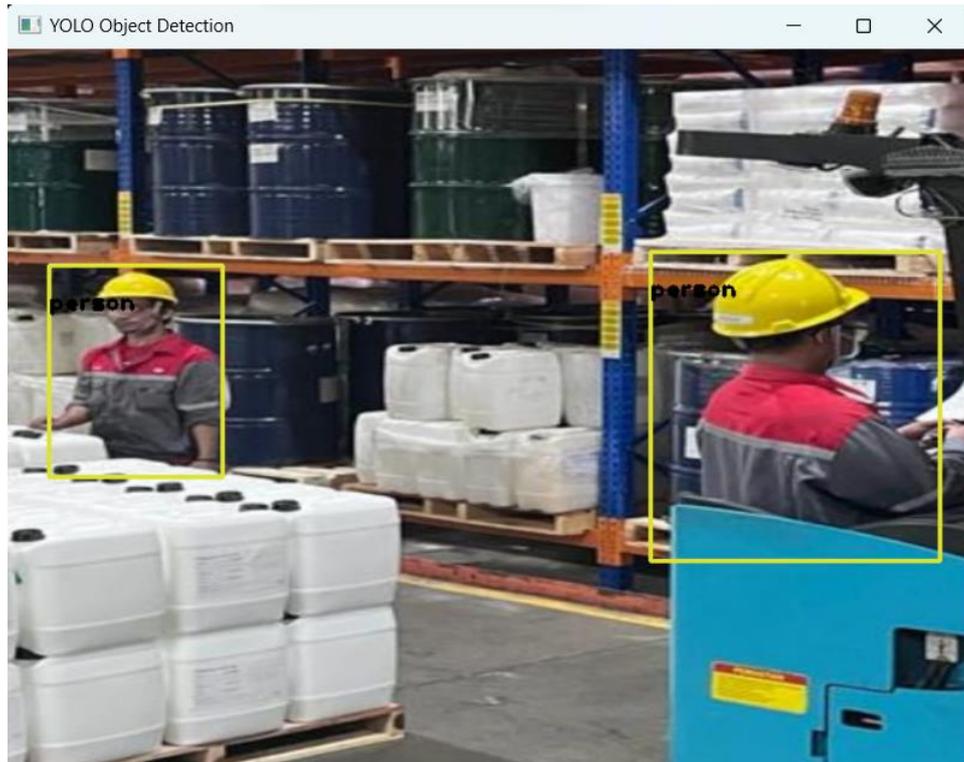
Gambar 5.2 Hasil Deteksi Drum dan IBC



Gambar 5.3 Hasil Deteksi IBC dan Box



Gambar 5.4 Hasil Deteksi Forklift



Gambar 5.5 Hasil Deteksi Orang

Hasil deteksi objek dapat dilihat pada Gambar 5.1 hingga Gambar 5.5 Proses deteksi objek diuji pada kondisi forklift diam. Model yang sudah dimuat sebelumnya yaitu *yolov3-custom\_final.weights* akan memindai gambar dan menerapkan beberapa fitur yang berupa pola persegi atau persegi panjang yang digunakan untuk mengidentifikasi keberadaan objek pada frame. Kemudian model akan menentukan apakah fitur-fitur tersebut cocok dengan pola objek yang ada pada dataset yang dilatihkan ke model. Hasil pada Gambar 5.1 hingga Gambar 5.5 merupakan output yang sudah diberlakukan NMS (*Non Max Suppression*) untuk output yang dihasilkan, sehingga bounding box yang terbentuk hanya satu untuk setiap objek yang terdeteksi.

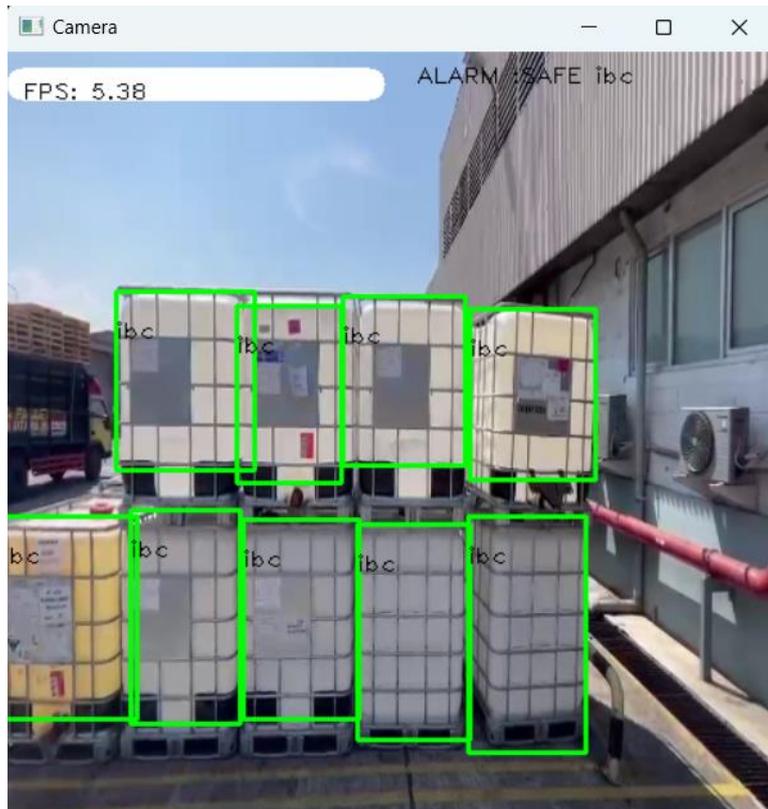
Setelah objek terdeteksi oleh model, perhitungan jarak akan dimulai dengan memanfaatkan nilai width dan height dari bounding box yang terbuat pada hasil deteksi. Perhitungan jarak menggunakan rumus yang sudah ditetapkan di awal program sebagai fungsi. Selanjutnya, hasil perhitungan jarak yang didapatkan akan dibedakan menjadi 4 output zone, yakni stop, bahaya, peringatan, dan aman. Output notifikasi suara yang dihasilkan pun akan berbeda tergantung pada zone setiap objek yang terdeteksi. Perbedaan jarak dengan output yang dihasilkan dapat dilihat pada Tabel 5.1 di bawah ini.

Tabel 5.1

Tabel Perbandingan Jarak dan Output

Jarak	Output
< 2 meter	Stop
2 – 3 meter	Danger
3 – 4 meter	Caution
> 4 meter	Safe

Pada Gambar 5.6 adalah hasil setelah diberlakukannya estimasi jarak pada sistem deteksi objek. Hasil kesimpulan dari estimasi terlihat pada pojok kanan atas frame gambar yang bertuliskan “ALARM :” serta FPS yang dihasilkan juga tertulis pada pojok kiri frame gambar yang bernilai 5 hingga 7 pada saat percobaan.



Gambar 5.6 Hasil Deteksi Objek dengan Estimasi Jarak

## 5.2 Pengujian Performance

Pada bab ini membahas mengenai pengujian dataset dalam proses training model. Pengujian dilakukan pada satu dataset yang berisi objek-objek yang ada pada pabrik PT Henkel Adhesive Technologies Pasuruan, tepatnya pada area industri.

### 5.2.1 Dataset secara Menyeluruh

Pada dataset ini terbagi menjadi 6 kelas yang memiliki total 483 gambar untuk data training dan 83 gambar untuk data testing, dimana masing-masing gambarnya telah dilakukan resize menjadi sebesar 224x224 dan telah ditambahkan bounding box untuk setiap objek yang akan dideteksi pada setiap gambarnya, yang sudah tersimpan pada file .txt satu per satu. Gambaran dataset secara keseluruhan dapat dilihat pada Gambar 5.7.



Gambar 5.7 Gambaran Dataset Secara Menyeluruh

Setelah dilakukan pre-processing, dataset tersebut siap digunakan untuk melatih model Yolov3 yang telah disesuaikan pada file yolov3.cfg, yaitu perubahan pada classes yang akan dideteksi pada baris 610, 696, dan 783 diubah menjadi 6 karena mengikuti banyaknya kelas yang akan dideteksi, dan juga pada bagian filters pada layer [convolutional] pada baris 603, 689, dan 776. Selama proses training, batch size yang digunakan sebesar 64 dengan max batch sebesar 12.000.

Setelah proses training selesai, model akan dilakukan evaluasi dengan dataset testing yang sudah disiapkan sebelum digunakan untuk deteksi objek. Dapat dilihat pada Tabel 5.2, kelas yang memiliki nilai average precision paling tinggi adalah drum yakni menyentuh angka 99%, dilanjutkan dengan IBC dan forklift yakni di angka 95% dan 91%. Hal tersebut bisa terjadi kemungkinan dikarenakan IBC dan forklift hanya memiliki 1 bentuk yang pasti dan tidak berbeda satu dan yang lain, sehingga model lebih mudah untuk membaca dan mengklasifikasikannya.

Tabel 5.2

Tabel *Average Precision* setiap Kelas.

<b>Class</b>	<b>Average Precision</b>	<b>TP</b>	<b>FP</b>
<i>box</i>	82.70%	89	8
<i>cone</i>	85.14%	34	6
<i>drum</i>	99.58%	57	4
IBC	91.67%	11	0
<i>forklift</i>	95.45%	21	1
<i>person</i>	87.21%	31	5

Pada Tabel 5.3 juga sudah disajikan hasil dari rata-rata precision dari model, hasil recall, dan f1-score yang diperoleh dari hasil training sebelumnya, dimana precision bernilai 0.91, recall bernilai 0.89 dan f1-score bernilai 0.90, yang berarti model yang terbuat memiliki hasil prediksi yang cukup baik. Selain itu juga terdapat nilai TP (*True Positive*) sebesar 243, FP (*False Positive*) sebesar 24, serta FN (*False Negative*) yang bernilai 30. Nilai dari Average IoU juga ditampilkan yakni sebesar 75,36%, IoU sendiri merupakan metrik yang digunakan dalam proses NMS (*Non Max Suppression*) yang dilakukan oleh model Yolov3

Tabel 5.3

Tabel Hasil Prediksi

<i>Precision</i>	0.91
<i>Recall</i>	0.89
<i>F1-score</i>	0.90
TP	243
FP	24
FN	30
<i>Average IoU</i>	75.36%

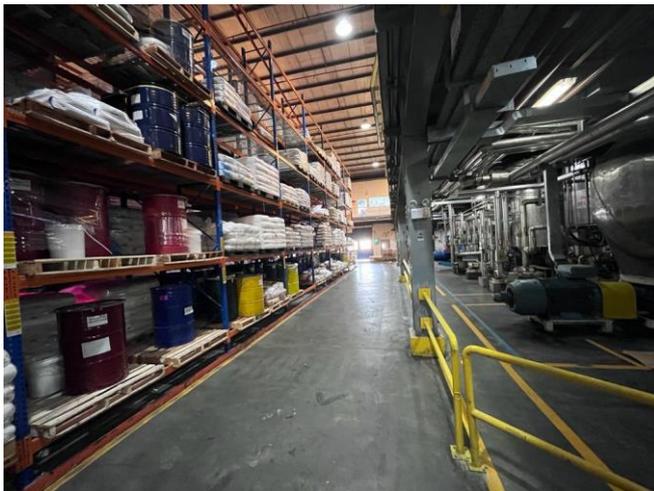
### 5.2.2 Penempatan Posisi Kamera

Pada sub bab ini akan membahas mengenai posisi penempatan kamera pada sisi-sisi forklift untuk mendeteksi objek yang berada di sekitar forklift. Penggunaan kamera berperan sebagai input yang akan membaca dan mengamati objek yang ada di sekitar forklift selama forklift berjalan atau berhenti, sehingga dapat mendeteksi jika ada objek yang sudah memasuki area bahaya forklift. Kamera ini akan ditempatkan di tiga sisi forklift yang berbeda, yakni kiri, kanan, dan belakang. Penempatan kamera ini didasarkan pada posisi yang memungkinkan kamera dapat menangkap pengelihatannya yang cukup luas, serta sisi-sisi yang lebih banyak terjadi *nearmiss* pada pabrik. Area sudut pandang untuk penempatan kamera yang pertama terlihat pada Gambar 5.8, kamera akan terpasang pada bagian sisi kiri forklift pada kanopi forklift. Dengan posisi ini, kamera dapat langsung menghadap ke area kiri forklift, memungkinkan kamera untuk mengambil gambar dengan sudut pandang area yang cukup untuk bagian sisi kiri forklift. Untuk sketsa penempatan kamera dapat dilihat pada bab 3 tepatnya pada Gambar 3.4



Gambar 5.8 Sudut Pandang Kamera Bagian Kiri

Area sudut pandang kamera kedua dapat dilihat pada Gambar 5.9, kamera diletakkan di bagian sisi kanan kanopi forklift. Posisi tersebut juga sangat bagus dikarenakan memungkinkan kamera untuk mengambil gambar dengan jangkauan yang cukup, sama halnya dengan kamera pada posisi pertama sebelumnya.



Gambar 5.9 Sudut Pandang Kamera Bagian Kanan

Area sudut pandang kamera ketiga dapat dilihat pada Gambar 5.10, kamera diletakkan pada bagian sisi belakang kanopi forklift. Posisi ini adalah posisi yang bisa dibilang paling krusial dikarenakan lebih banyak terjadi *nearmiss* yang melibatkan bagian belakang forklift. Oleh karena itu, pada posisi kamera ketiga ini dirancang agar kamera bisa mendapatkan pandangan yang luas terhadap keadaan bagian belakang forklift.



Gambar 5.10 Sudut Pandang Kamera Bagian Belakang

Setelah beberapa posisi kamera telah ditentukan, setiap kamera akan dilakukan uji coba untuk mengetahui apakah sistem dapat mendeteksi objek yang ada masuk pada area sekitar forklift. Pengujian akan dilakukan dengan memanfaatkan *yolov3-custom\_final.weights* untuk mendeteksi objek yang ada. Hal ini bertujuan agar mengetahui apakah posisi kamera yang telah ditentukan sudah optimal atau belum, dan apakah sistem dapat mendeteksi objek yang ada serta melakukan estimasi jarak terhadap objek tersebut, atau bahkan model yang harus dikembangkan lagi untuk menaikkan akurasi pada deteksi objek.

### 5.2.3 Hardware yang Digunakan

*Hardware* yang berperan sebagai *main processor* untuk pengolahan gambar yang ditangkap kamera, sekaligus juga tersambung pada 3 kamera yang sudah dibahas sebelumnya. Gambar 5.11 menunjukkan *hardware* yang digunakan, yaitu Intel NUC5i3ryh, dengan spesifikasi processor Intel core i3 gen 5 dengan OS Windows 8 yang akan tersambung dengan 3 usb webcam berspesifikasi 720p dengan 30 fps. NUC berperan sebagai *processor* untuk mengolah input gambar dari kamera.



Gambar 5.11 Hardware yang Digunakan

### 5.3 Hasil Pengujian

Pada bab ini, akan dibahas mengenai hasil pengujian model terhadap kondisi dimana forklift sedang berjalan. Pengujian ini akan dibagi menjadi 2 bagian, yaitu pengujian terhadap akurasi output yang akan dihasilkan berdasarkan perhitungan estimasi jarak yang telah dilakukan, apakah sama dengan jarak aslinya, dan pengujian kedua terhadap waktu yang dibutuhkan oleh sistem untuk mengeluarkan output notifikasi berdasarkan perhitungan estimasi jarak yang telah didapat. Pengujian juga dilakukan terhadap pergantian input, seberapa berpengaruh untuk hasil deteksi. Tak lupa juga untuk memperhatikan nilai FPS yang dihasilkan untuk setiap percobaannya, mengingat fps sangat berpengaruh dalam pembacaan kamera.

Untuk melaksanakan pengujian secara nyata, proses pengujian akan dilakukan dengan pengambilan video pada keadaan forklift oleh seseorang yang mengoperasikan forklift, dan seorang lagi yang akan merekam jalannya forklift dari luar forklift. Hasil rekaman pengujian berupa video akan dilampirkan dalam bentuk frame by frame dari video pengujian tersebut. Hal ini bertujuan untuk memberikan gambaran tentang setiap peristiwa yang terjadi selama pengujian, khususnya pada sudut pandang kamera.

Pada pengujian ini, input frame yang dimasukkan ke dalam model akan diresize menjadi ukuran (256,256) sebagai input dari model deteksi Yolov3. Alasannya dipilih ukuran (256,256) untuk *width* dan *height* untuk input dikarenakan penulis telah

melakukan beberapa percobaan dari yang awalnya input size default Yolo bernilai (416,416). Sistem menghasilkan fps yang sangat rendah, walaupun tingkat sensitivitas deteksi model sedikit meningkat. Mengingat bahwa pada kasus ini fps merupakan salah satu satuan yang perlu diperhatikan juga, maka *input size* diturunkan menjadi ukuran (256,256), sehingga fps meningkat dan lebih stabil dibandingkan sebelumnya walau sensitivitas deteksi sedikit menurun. Fps yang dihasilkan ketika input sizenya sebesar (416,416) bernilai 1 hingga 2 saja, sedangkan untuk input size (256,256) sebesar 2 hingga 6. Hasil perbandingan fps dapat dilihat pada Tabel 5.4 di bawah ini.

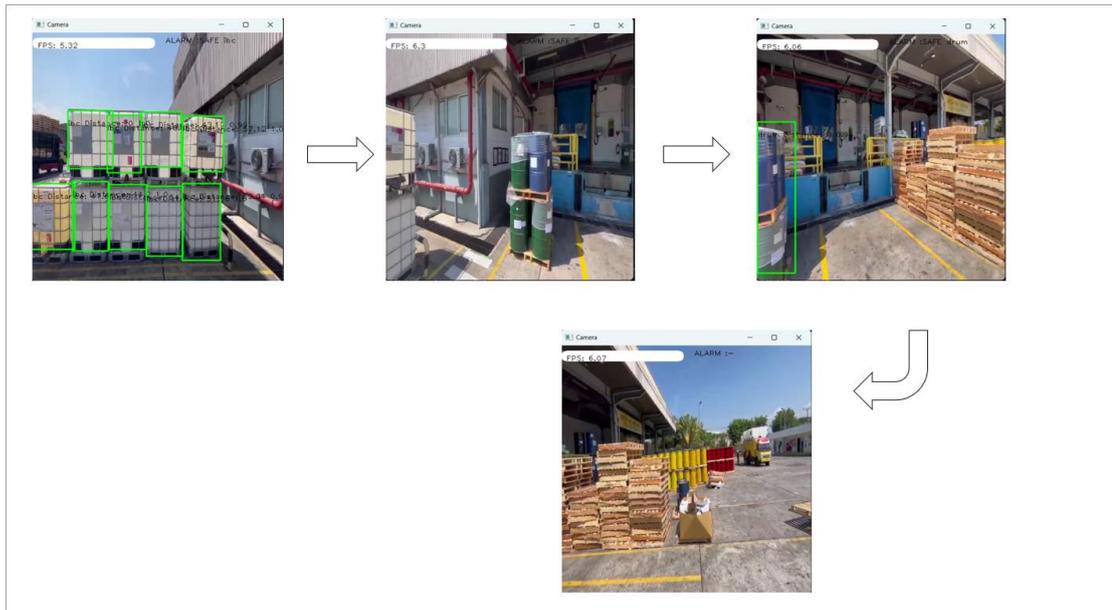
Tabel 5.4

Tabel Perbandingan Hasil Fps

Input	Hasil fps
(416,416)	0,33 – 2,7
(320,320)	1,92 – 4,28
(256,256)	2 – 6,1
(192,192)	3 – 9,8

### 5.3.1 Hasil Pengujian pada Area Pertama

Pada Gambar 5.12 merupakan potongan-potongan gambar video yang diambil *frame by frame*, yang merupakan hasil deteksi serta perhitungan estimasi jarak yang dilakukan menggunakan kamera. Pada sisi kiri frame terdapat nilai FPS yang dihasilkan pada saat proses berlangsung. Pada bagian kanan frame terdapat alarm yang menandakan di *zone* manakah objek yang terdeteksi ini berada.



Gambar 5.12 Hasil Deteksi dan Estimasi Jarak pada Area Pertama

Pada sistem ini terbagi menjadi 4 *zone* yaitu ada *stop*, *danger*, *caution*, dan *safe*, yang sudah dijelaskan pada Tabel 5.1. Pada Gambar 5.12, IBC dan drum yang terdeteksi berada pada jarak kurang lebih 3 meter dari kamera, sehingga alarm output yang dihasilkan adalah *safe*. Pada akhir *frame* video terdapat beberapa tumpukan drum yang tidak terdeteksi, hal ini dikarenakan benda tersebut yang terlalu jauh dengan kamera, sehingga memungkinkan kamera tidak dapat mendeteksi objek tersebut.

### 5.3.2 Hasil Pengujian pada Area Kedua

Pada Gambar 5.13 merupakan beberapa potongan gambar dari video yang diambil *frame by frame*, yang merupakan hasil deteksi dan perhitungan estimasi jarak menggunakan video keadaan pabrik secara langsung. Hasilnya dapat dilihat bahwa yang terdeteksi sebagian besar adalah IBC dan drum yang tersusun tidak terdeteksi oleh sistem, bahkan ada yang terdeteksi sebagai *box*, serta tumpukan pallet yang terdeteksi sebagai IBC.



Gambar 5.13 Hasil Deteksi dan Estimasi Jarak pada Area Kedua

Pada Gambar 5.13 merupakan beberapa potongan gambar dari video yang diambil *frame by frame*, yang merupakan hasil deteksi dan perhitungan estimasi jarak menggunakan video keadaan pabrik secara langsung. Hasilnya dapat dilihat bahwa yang terdeteksi sebagian besar adalah IBC, drum yang tersusun tidak terdeteksi oleh sistem, bahkan ada yang terdeteksi sebagai *box*, serta tumpukan pallet yang terdeteksi sebagai IBC.



Gambar 5.14 Drum yang Terdeteksi sebagai Box

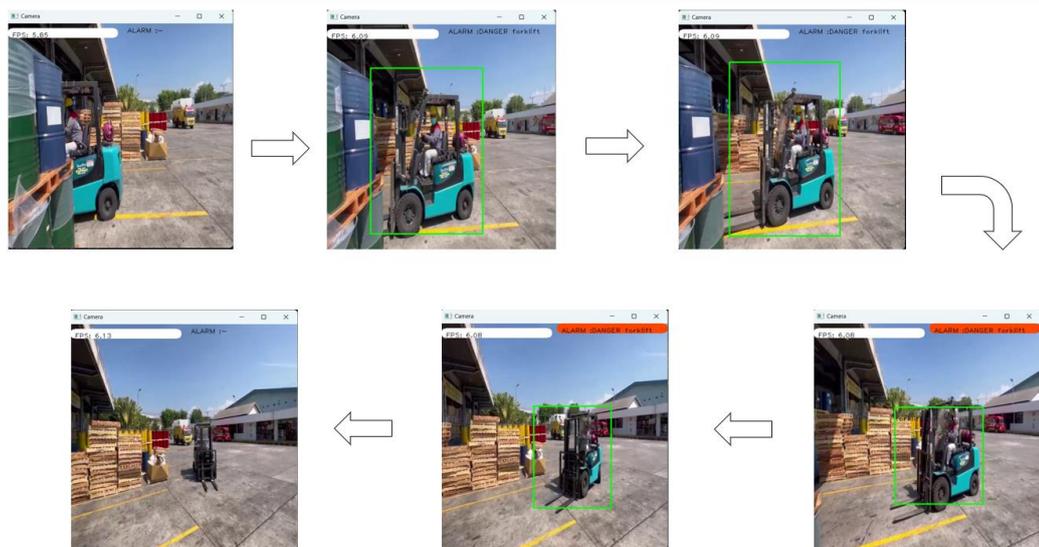


Gambar 5.15 Tumpukan Pallet yang Terdeteksi sebagai IBC

Pada Gambar 5.14 dan 5.15 merupakan potongan gambar yang ada pada Gambar 5.13, yang merupakan beberapa kesalahan deteksi oleh sistem. Hal ini bisa jadi dikarenakan drum berwarna kuning kecoklatan dan berdempetan sebanyak 3 drum, sehingga dapat terdeteksi sebagai box yang berukuran kotak jika dilihat dari depan oleh sistem, sedangkan tumpukan pallet dapat terdeteksi sebagai IBC bisa jadi dikarenakan memiliki bentuk yang hampir sama dengan IBC jika pallet ditumpuk, dan pada video tersebut tumpukan pallet dilapisi dengan plastik putih transparan yang merupakan warna dasar dari IBC.

### 5.3.3 Hasil Pengujian pada Area Ketiga

Pada Gambar 5.16 merupakan beberapa potongan gambar dari video yang diambil *frame by frame*, yang merupakan hasil deteksi dan perhitungan estimasi jarak pada saat posisi kamera/forklift sedang diam. Pada hasil *frame*, dapat dilihat bahwa forklift terdeteksi oleh sistem, dan alarm yang dihasilkan adalah danger karena forklift berada pada posisi 1 hingga 2 meter dari posisi kamera. Pada sistem ini juga dirancang agar pada saat ada forklift yang terdeteksi di *zone* manapun, alarm akan menghasilkan output *danger*, tujuannya agar operator forklift dapat berhenti sejenak atau lebih berhati-hati karena ada forklift lain yang sedang beroperasi di sekitar.



Gambar 5.16 Hasil Deteksi dan Estimasi Jarak pada Area Ketiga

Pada awal video terlihat ada tumpukan drum pada sebelah kiri kamera yang tidak terdeteksi oleh sistem. Hal tersebut bisa terjadi dikarenakan kurangnya dataset drum dengan warna yang bervariasi berdempetan, sehingga memungkinkan sistem kurang mampu untuk mendeteksi bahwa barang tersebut adalah drum. Di akhir video, tampak depan forklift juga tidak terdeteksi, hal tersebut dikarenakan kurangnya dataset tampak depan forklift, atau memang jarak antara forklift dengan kamera yang sudah diluar kemampuan sistem untuk mendeteksi objek tersebut.

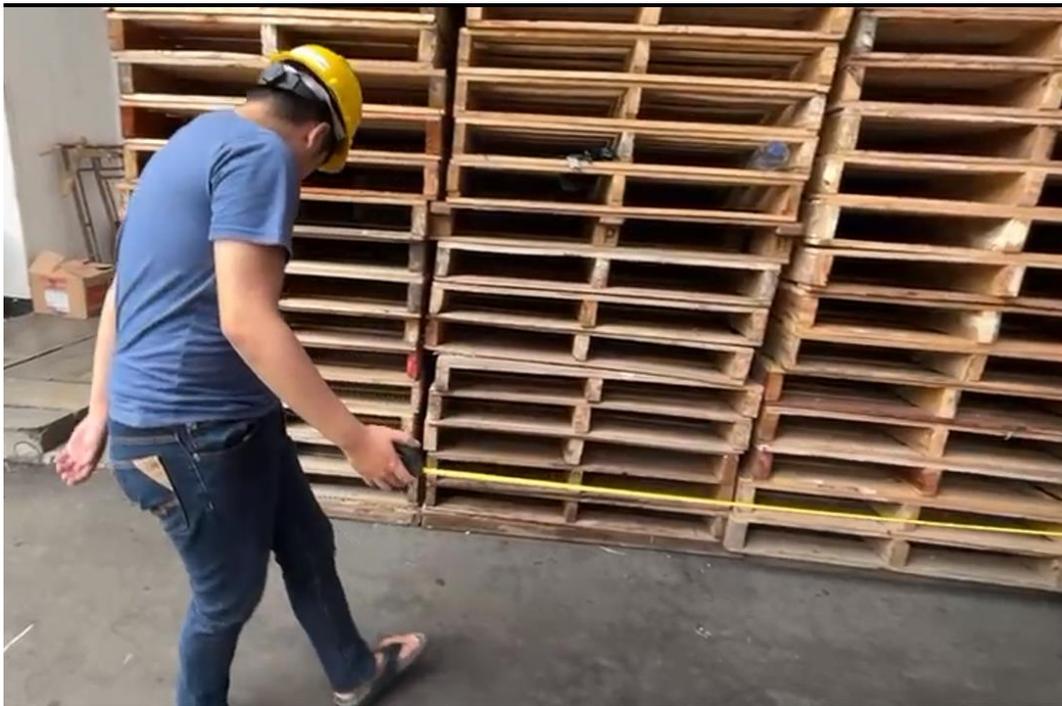
### 5.3.4 Pengujian terhadap Jarak Nyata

Pada sub bab ini akan dibahas tentang perbandingan nilai dari estimasi jarak yang diperoleh sistem dengan jarak nyata.

Pada pengujian ini akan diukur terlebih dahulu jarak suatu objek dengan kamera menggunakan meteran, lalu setelah itu dibandingkan dengan nilai estimasi jarak yang didapat dari perhitungan sistem. Pada Gambar 5.17, Gambar 5.18 merupakan pengujian jarak nyata dengan perhitungan estimasi jarak, dengan hasilnya dapat dilihat pada Tabel 5.5, merupakan hasil dari perbandingan jarak nyata dengan hasil estimasi jarak.



Gambar 5.17 Pengujian Jarak Nyata dengan Estimasi Jarak (1)



Gambar 5.18 Pengujian Jarak Nyata dengan Estimasi Jarak (2)

Tabel 5.5

Tabel Perbandingan Jarak Nyata dengan Estimasi Jarak

Percobaan ke -	Jarak Nyata	Estimasi Jarak
1	2,36 m	2,2 m
2	1,71 m	1,5 m
3	3 m	2,83 m
4	3,77 m	3,81 m
5	4,2 m	4,4 m

(ganti jadi error) Dari beberapa percobaan yang telah dilakukan pada Tabel 5.5 di atas, didapatkan rata-rata error sebesar 0,156 meter dari setiap percobaan yang telah dilakukan.

#### 5.3.4 Hasil Alarm Pengujian Estimasi Jarak

Pada sub bab ini akan membahas tentang kesimpulan yang dapat diambil dari beberapa percobaan di atas mengenai hasil output alarm yang dihasilkan oleh sistem dibandingkan dengan jarak aslinya.

Pada Tabel 5.6 di bawah ini merupakan kesimpulan hasil output alarm dari percobaan pada area 1 yang telah dilakukan pada sub bab 5.3.1.

Tabel 5.6

Tabel Hasil Deteksi dan Output Alarm pada Area 1

Objek	Ground Truth	Hasil Alarm	Counter
lbc	Safe	Safe	8
Drum		Safe	4

Pada saat pengambilan video pada keadaan asli pabrik, jarak antara kamera dengan objek berkisar kurang lebih 4 meter. Pada sistem, jarak 4 meter akan menghasilkan jarak *safe*, yang berarti dikategorikan pada zona aman. Pada percobaan area 1 ini, akurasi yang didapat untuk output alarm yang dihasilkan sistem adalah 100% karena semua objek yang terdeteksi mengeluarkan output *safe*, yang berarti benar adanya estimasi jarak yang terjadi menghasilkan jarak kurang lebih 4 meter.

Pada Tabel 5.7 di bawah ini merupakan kesimpulan hasil output alarm dari percobaan pada area 2 yang telah dilakukan pada sub bab 5.3.2.

Tabel 5.7

Tabel Hasil Deteksi dan Output Alarm pada Area 2

Objek	<i>Ground Truth</i>	Hasil Alarm	Counter
IBC	Danger	Stop	2
IBC		Danger	8
drum		Stop	1
drum		Danger	5
box		Danger	4

Pada saat pengambilan video, jarak antara kamera dengan objek berkisar antara kurang lebih 2 meter. Pada sistem, alarm yang dihasilkan terpantau berubah-ubah dari *stop* dan *danger*. Hal ini dikarenakan perhitungan nilai estimasi jarak yang dilakukan oleh sistem pasti berubah-ubah untuk setiap detiknya, tergantung juga pada *bounding box* yang terbentuk pada saat proses deteksi objek yang dilakukan oleh sistem. Alarm sempat mengeluarkan alarm *danger* dikarenakan perhitungan jarak sistem mencapai angka di atas 2 meter dan kurang dari 3 meter, yang dimana dikategorikan sebagai zona *danger*. Pada percobaan ini, akurasi yang diperoleh untuk IBC sebesar 80%, drum mencapai 83% dan *box* mencapai 100%.

Pada Tabel 5.8 di bawah ini merupakan kesimpulan hasil output alarm dari percobaan pada area 3 yang telah dilakukan pada sub-bab 5.3.3.

Tabel 5.8

Tabel Hasil Deteksi dan Output Alarm pada Area 3

Objek	<i>Ground Truth</i>	Alarm	Counter
Forklift	Danger	Danger	7

Pada saat pengambilan video jarak antara forklift kamera berkisar kurang lebih 3 meter, yang berarti pada zona *caution*. Alarm yang dihasilkan oleh sistem adalah alarm *danger*. Ini dikarenakan objek yang terdeteksi oleh sistem adalah forklift, pada program sudah dirancang jika objek yang terdeteksi merupakan forklift, maka berapapun jarak yang terhitung oleh sistem, alarm yang akan dikeluarkan adalah *danger*. Ini bertujuan agar

operator forklift lebih berhati-hati karena ada forklift lain yang juga beroperasi pada area tersebut.

### 5.3.5 Hasil Pengujian Waktu untuk Output Alarm

Pada sub bab ini akan membahas hasil perbandingan waktu yang dibutuhkan sistem untuk mengeluarkan output suara dengan kecepatan berjalan forklift

Pada Tabel 5.9 di bawah ini merupakan kesimpulan hasil waktu yang dibutuhkan oleh sistem untuk mengeluarkan output suara berdasarkan kecepatan forklift

Tabel 5.9

Tabel Hasil Waktu untuk Mengeluarkan Output dengan Nilai counter 10

Percobaan ke -	Waktu
1	1,7 detik
2	2,2 detik
3	2,2 detik
4	1,9 detik
5	2,1 detik

Menurut percobaan yang telah dilakukan pada Tabel 5.9 di atas, didapatkan hasil rata-rata waktu yang dibutuhkan oleh sistem untuk mengeluarkan output notifikasi suara setelah objek terdeteksi adalah 2 detik. Sistem memerlukan 2 detik untuk mengeluarkan output dikarenakan adanya variable *counter* yang berperan sebagai *stabilizer* alarm yang dihasilkan terhadap nilai estimasi jarak yang terhitung. Konsekuensi dari hasil 2 detik yang didapat ini adalah jika kecepatan forklift mencapai 10 km/jam atau lebih, dikarenakan forklift akan berjalan lebih cepat dibandingkan kecepatan sistem untuk mengeluarkan notifikasi alarm, sehingga masih berpotensi untuk menimbulkan *nearmiss*.

Variable *counter* yang sudah dijelaskan sebelumnya bertugas sebagai *stabilizer* output agar sistem tidak terlalu sensitif ketika ada objek yang terdeteksi. Untuk menurunkan waktu respon yang dibutuhkan, salah satu caranya adalah menurunkan *counter* yang digunakan pada sistem, agar balancing yang dilakukan untuk tiap *frame* yang terdeteksi tidak terlalu lama. Pada Tabel 5.10 di bawah ini merupakan hasil waktu yang dibutuhkan oleh sistem setelah nilai *counter* diubah dari 10 menjadi 2.

Tabel 5.10

Tabel Hasil Waktu untuk Mengeluarkan Output dengan Nilai counter 2

Percobaan ke -	Waktu
1	0,9 detik
2	1,3 detik
3	0,8 detik
4	1,2 detik
5	1,2 detik

Dari hasil Tabel 5.9 dan Tabel 5.10 dapat disimpulkan bahwa *counter* sangat berpengaruh terhadap kecepatan sistem untuk mengeluarkan notifikasi alarm. Pada Tabel 5.10 waktu yang dihasilkan oleh sistem rata-rata 1,08 detik, tetapi sistem menjadi lebih sensitif terhadap objek yang terdeteksi sehingga juga berdampak pada notifikasi alarm yang dikeluarkan.

#### 5.4 Pengujian Tambahan dengan Metode Yolov5

Hasil nilai fps yang dihasilkan menggunakan Yolov3 ketika dijalankan menggunakan perangkat Intel NUC menghasilkan fps yang sangat rendah, yakni 1 hingga 2, sehingga pada akhir waktu sisa pengerjaan, penulis mencoba mengganti model yang digunakan menjadi Yolov5, tepatnya Yolov5s. Fps yang dihasilkan dengan menggunakan model tersebut mencapai 15 hingga 20. Perlu diingat, yang diganti adalah model saja, dataset yang digunakan tetap sama seperti yang telah digunakan pada model Yolov3. Kenaikan fps yang sangat drastis tersebut kemungkinan dikarenakan ukuran file dari kedua model yang sangat berbeda jauh. *Weights* yang dihasilkan pada Yolov3 (.weights) mencapai 250MB, yang memungkinkan sistem menjadi terlalu berat untuk membaca file tersebut, sedangkan pada Yolov5s, ukuran file *weights* (.pt) yang dihasilkan hanya mencapai 15MB, yang memungkinkan sistem lebih cepat dalam membaca file tersebut karena ukurannya yang kecil. Nilai fps yang tinggi tentunya juga berdampak pada kecepatan sistem dalam mendeteksi objek yang ada serta menghitung estimasi jarak dengan lebih optimal, sehingga dapat meningkatkan akurasi yang dihasilkan tentunya dengan beberapa penyesuaian

Dari percobaan di atas, dapat disimpulkan bahwa Yolov5s layak untuk dilakukan penelitian lanjutan demi mengembangkan sistem yang sudah ada ini. Tak hanya dalam

segi model, penambahan dataset yang lebih variatif dan beragam dapat meningkatkan lagi performa dari sistem ini, tentunya dalam mendeteksi objek yang ada.