

## BAB III METODOLOGI PENYELESAIAN MASALAH

### III.1 Metodologi

Metodologi yang digunakan pada penelitian ini adalah metodologi eksperimen. Metodologi eksperimen adalah sebuah percobaan yang bersistem dan berencana untuk membuktikan kebenaran dari suatu teori. Tujuannya untuk menemukan dan mengetahui hasil dari pengujian yang dilakukan. Adapun tahapan untuk metodologi eksperimen sebagai berikut [39]:

#### III.1.1 Memilih dan merumuskan masalah.

Perumusan masalah penelitian dijabarkan pada Tabel III.1.

Tabel III.1 Pemilihan dan merumuskan masalah

Masalah	Rumusan Masalah
<ol style="list-style-type: none"><li>1. Pergerakan robot akan statis jika tidak memiliki sensor. Integrasi dengan sensor kamera membuat pergerakan robot menjadi dinamis dan lebih mudah beradaptasi.</li><li>2. Klasifikasi menggunakan kamera dengan penggunaan metode warna dan ukuran memiliki kekurangan dalam kondisi cahaya lingkungan.</li></ol>	Bagaimana mendeteksi makanan menggunakan <i>computer vision</i> dengan posisi kamera menempel pada lengan robot?
<ol style="list-style-type: none"><li>1. Dalam industri makanan, proses seperti <i>pick and place</i>, <i>sorting</i>, dan <i>packaging</i> sering kali melibatkan tugas-tugas yang berulang, mengandalkan tenaga manual sangat tidak efisien.</li></ol>	Bagaimana melakukan <i>pick and place</i> makanan dari atas <i>conveyor</i> bergerak menggunakan robot dengan <i>soft gripper</i> ?

Masalah	Rumusan Masalah
2. Penggunaan <i>gripper</i> umumnya untuk benda keras sehingga tidak cocok dengan makanan yang memiliki tekstur semi kaku.  3. Perpindahan objek melalui <i>conveyor</i> pada industri makanan membantu perpindahan dari proses ke proses lebih cepat dibanding manual.	

Berdasarkan masalah yang didapatkan, agar lebih spesifik dalam pembahasan, maka ditulis beberapa batasan masalah yaitu:

1. Robot yang digunakan adalah robot Universal Robot 5e (UR5e)
2. Arah cengkaman objek hanya dari atas.
3. Objek untuk pendeteksian dan *pick and place* yaitu jeruk, pisang, roti bulat dan roti panjang.
4. Kecepatan *conveyor* tetap pada 1,29cm/detik.
5. Tidak membahas letak orientasi benda yang sudah ditentukan.
6. Tidak membahas desain efektor akhir.

### III.1.2 Memilih subjek dan instrumen pengukuran.

Subjek dalam penelitian ini yaitu *computer vision* dengan teknologi *deep learning* dapat melakukan deteksi objek dalam pencahayaan yang redup serta pengujian *computer vision* dan *soft gripper* pada proses *pick and place* oleh robot dengan objek makanan yang bergerak. Untuk instrumen pengukuran dari penggunaan *deep learning* dengan metode *SSDLite MobileNet V3* menerapkan berbagai standar yang digunakan untuk mengevaluasi model pembelajaran mesin. Pada penelitian [40] ada empat kemungkinan hasil berdasarkan kategori keluaran sampel uji dibandingkan dengan kategori label yang benar, sebagai berikut: *true positive* (TP), *false positive* (FP), *false negative* (FN), dan *true negative* (TN).

Tabel III.2 Penilaian keluaran deteksi objek

Kategorit	Keterangan
<i>True Positive</i> (TP)	Target objek terdeteksi dengan benar.
<i>True Negative</i> (TN)	Tidak mendeteksi target dan tidak ada <i>bounding box</i> .
<i>False Positive</i> (FP)	Tidak ada objek tetapi mendeteksi objek.
<i>False Negative</i> (FN)	Target mendeteksi objek tetapi salah.

Jumlah yang diamati digabungkan ke dalam metrik standar termasuk *recall*, presisi, skor F1 (F1), *mean average prediction* (MAP), dan *frame per second* (FPS). Dalam proses deteksi target, presisi adalah rasio target yang terdeteksi dengan benar terhadap jumlah semua target yang terdeteksi; recall adalah rasio jumlah target yang terdeteksi terhadap semua target dalam kumpulan sampel. Definisi presisi dan recall ditunjukkan pada di bawah ini:

$$Precision(P) = \frac{TP}{TP + FP} \dots\dots\dots(III.1)$$

$$Recall(R) = \frac{TP}{TP + FN} \dots\dots\dots(III.2)$$

F1 adalah rata-rata dari presisi dan *recall*. Skor F1 dapat dihitung dari tingkat presisi dan *recall*, seperti yang didefinisikan dalam rumus:

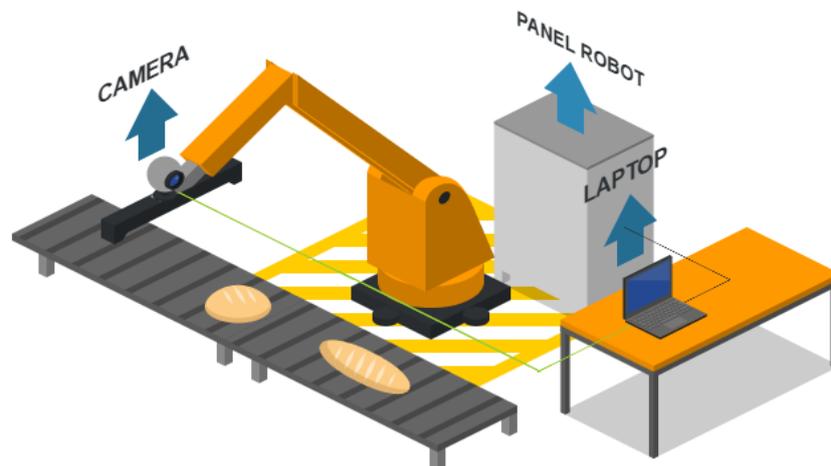
$$F1 = \frac{2PR}{P + R} \dots\dots\dots(III.3)$$

FPS adalah indikator umum untuk mengevaluasi kecepatan pendeteksian model. Ini mengacu ke jumlah gambar yang dapat diproses per detik. Untuk instrumen pengukuran *pick and place* dilaksanakan dengan menghitung keberhasilan yang dilakukan pada proses *pick* dan proses *place* serta memperhatikan perubahan bentuk pada makanan.

### III.1.3 Memilih desain penelitian.

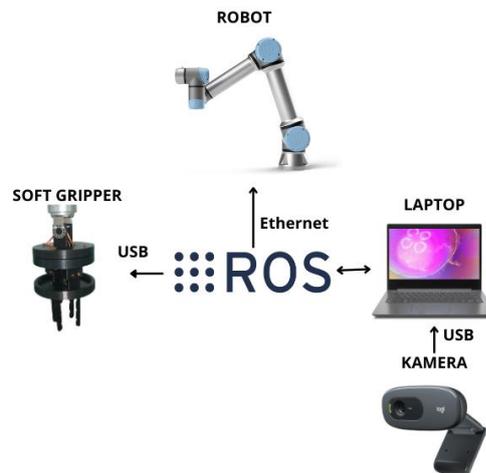
Sistem ini dirancang untuk melakukan proses *pick and place* makanan dari atas *conveyor* yang bergerak. Robot yang digunakan adalah UR5e, yang dilengkapi dengan efektor akhir berupa *soft gripper*. *Soft gripper* ini dipilih agar makanan yang dipindahkan tidak mengalami deformasi selama proses pemindahan.

Untuk mendeteksi dan mengenali makanan di atas *conveyor*, digunakan kamera Logitech C270 yang terpasang pada efektor akhir robot. Kamera ini terhubung ke laptop melalui USB dan berfungsi untuk mengambil citra makanan. Proses pengenalan citra memperhatikan aspek identifikasi objek dan menentukan koordinat objek tersebut. Ilustrasi sistem dapat dilihat pada Gambar III.1.



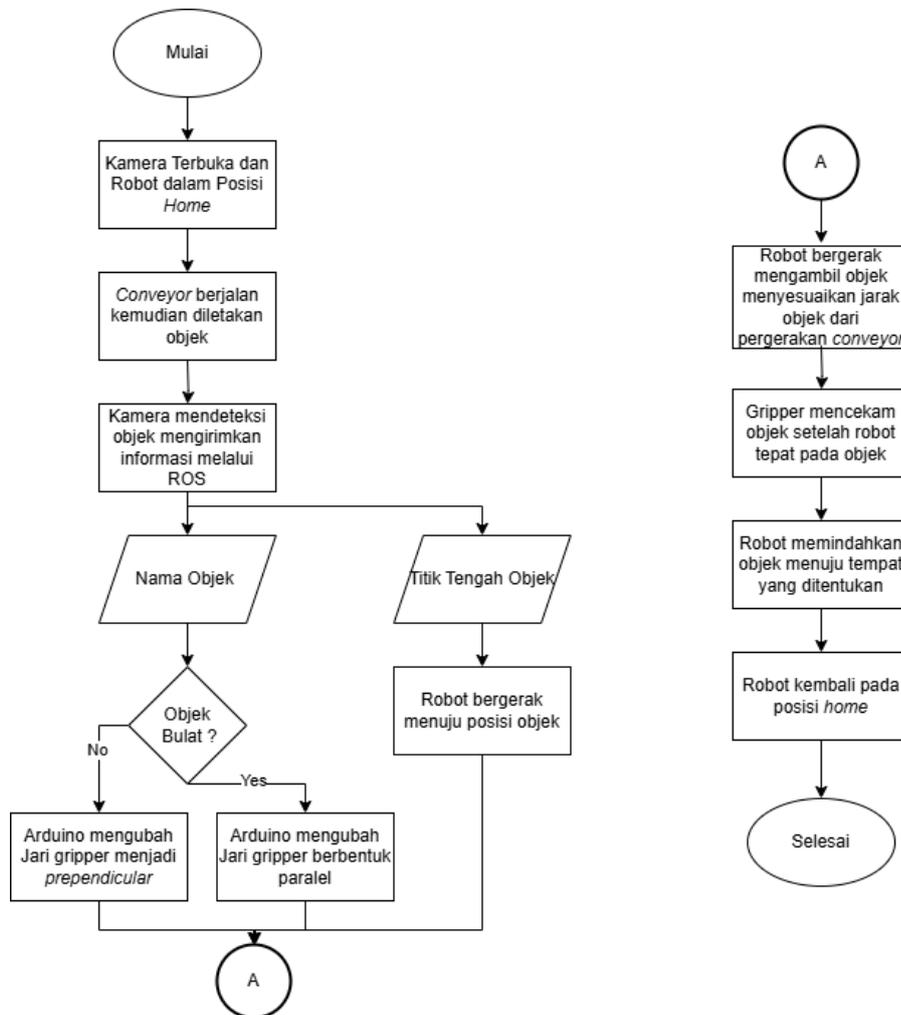
Gambar III.1 Ilustrasi 3D Sistem *Pick and Place*

Data yang diperoleh dari kamera berupa nama objek yang dideteksi dan koordinat objek kemudian diolah menggunakan ROS (*Robot Operating System*). Dalam ROS, informasi dari pengenalan citra diintegrasikan dengan sistem pengendali robot UR5e. Berdasarkan informasi koordinat objek yang dideteksi, robot kemudian melakukan gerakan untuk mencengkeram makanan, sedangkan informasi nama objek untuk mengubah bentuk *soft gripper* menjadi paralel atau *perpendicular* dan memindahkannya ke lokasi yang diinginkan. Dengan kata lain, sistem ini memungkinkan robot UR5e untuk mengambil dan memindahkan makanan dari *conveyor*, tanpa merusak bentuk makanan tersebut berkat penggunaan *soft gripper* dan *deep learning*.



Gambar III.2 Komunikasi Sistem

Berikut diagram alir dari cara kerja pada penelitian ini.



Gambar III.3 Diagram alur pada sistem *pick and place*

### III.1.4 Melaksanakan prosedur.

Prosedur penelitian sesuai dengan metode yang dipilih dari penelitian sebelumnya dijelaskan dalam teknis pelaksanaan penelitian pada:

#### III.1.4.1 Lingkungan Pengujian.

Pengujian dilakukan di Laboratorium Robotik dengan pencahayaan pada malam hari sebesar 260 lux. Pengukuran cahaya dilakukan dengan alat lux meter analog dengan merek *Kyoritsu*. Untuk pengujian deteksi objek dilakukan juga pada pencahayaan malam hari 60 lux, pengukuran ditunjukkan pada Gambar III.4 Pengukuran intensitas cahaya dengan lux meter. Selain itu, kecepatan *conveyor* konstan dengan kecepatan hasil pengukuran menggunakan *tacho* meter ditunjukkan pada Tabel III.3.

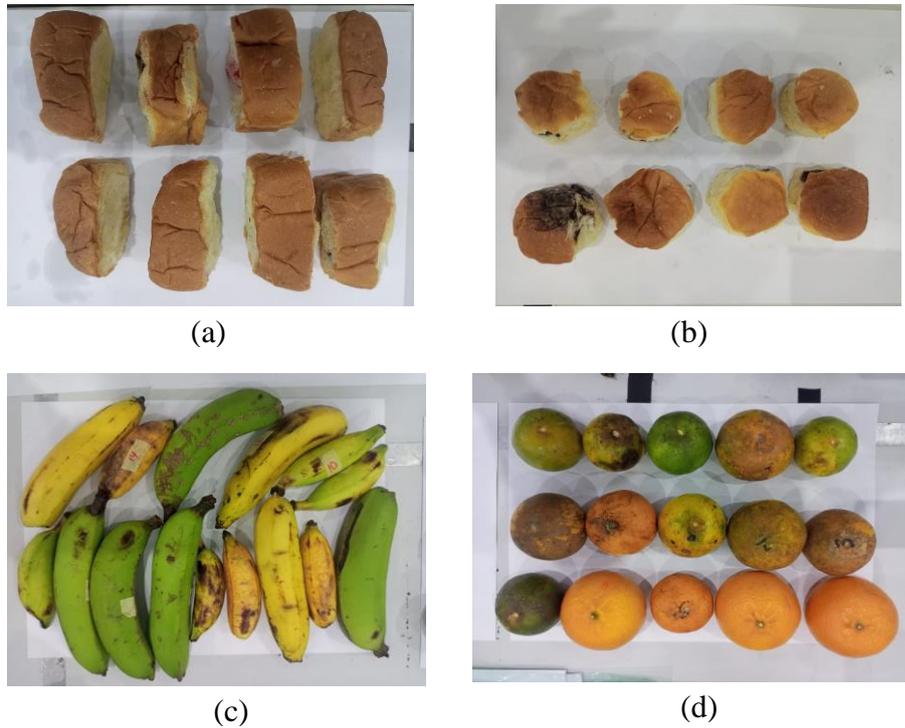


Gambar III.4 Pengukuran intensitas cahaya dengan lux meter (a) 260lux, (b)60lux

Tabel III.3 Pengukuran kecepatan *conveyor*

Percobaan	Kecepatan (m/menit)
1	0,75
2	0,8
3	0,75
4	0,79
5	0,74
6	0,8
7	0,76
8	0,68
9	0,75
10	0,83
11	0,86
12	0,8
Rata-rata (m/menit)	0,77
Konversi (cm/detik)	1,29

Pengujian benda akan dilakukan pada jeruk, pisang, roti bulat, dan roti panjang dengan macam variasinya. Berikut data ukuran dan berat yang digunakan:



Gambar III.5 Objek yang diuji coba (a) roti panjang, (b) roti bulat, (c) pisang, dan (d) jeruk

Pada buah-buahan yang digunakan memiliki warna dan ukuran yang berbeda untuk membuktikan jika objek deteksi menggunakan *deep learning* lebih unggul dibandingkan menggunakan deteksi warna dan ukuran objek.

Tabel III.4 Ukuran dari objek

Objek	Ukuran (p x l x t) (cm)	Berat (gram)
Jeruk 1	6,4 x 6,3 x 5,1	114,5
Jeruk 2	6,6 x 6,5 x 5,2	104,7
Jeruk 3	5 x 4,8 x 4,2	50,4
Jeruk 4	6,5 x 6,4 x 5,3	104,8
Jeruk 5	5,1 x 5 x 4,8	64,1
Jeruk 6	5,6 x 5,5 x 4,3	68,7
Jeruk 7	5,9 x 5,8 x 5,2	89,5
Jeruk 8	5,5 x 5,4 x 5	72
Jeruk 9	5,5 x 5,4 x 4,5	69,3
Jeruk 10	5,6 x 5,6 x 4,7	75,5
Jeruk 11	4,8 x 4,8 x 4,3	51,8
Jeruk 12	5,8 x 5,7 x 4,7	81,2
Jeruk 13	5,2 x 4,7 x 4,6	62,6

Objek	Ukuran (p x l x t) (cm)	Berat (gram)
Jeruk 14	5,1 x 5 x 4,7	64,7
Jeruk 15	4,6 x 4,6 x 4,3	49,7
Pisang 1	14,2 x 5,2 x 3,7	88
Pisang 2	8,5 x 3,2 x 3,1	35,6
Pisang 3	13,4 x 3,6 x 3,3	79,1
Pisang 4	8,3 x 3 x 2,8	35,5
Pisang 5	9,7 x 3 x 2,4	23,2
Pisang 6	14,5 x 4,4 x 3,3	84,5
Pisang 7	13,1 x 5,3 x 3,5	79,5
Pisang 8	14,5 x 4 x 3	81,9
Pisang 9	9,8 x 3,4 x 2,7	31,2
Pisang 10	9,9 x 3,3 x 2,4	27,9
Pisang 11	9,4 x 3 x 2,8	27,7
Pisang 12	13,3 x 4 x 3	70,9
Pisang 13	13,5 x 5,5 x 3,3	72,7
Pisang 14	8,8 x 3,4 x 3,2	37,7
Pisang 15	13,1 x 4,1 x 3,2	77,8
Roti Bulat 1	4,5 x 4,3 x 3,7	9,5
Roti Bulat 2	4,9 x 4,4 x 3	9,2
Roti Bulat 3	4,9 x 4,6 x 3,3	12,8
Roti Bulat 4	4,3 x 4,7 x 4	11,7
Roti Bulat 5	4,2 x 4,2 x 3,1	8,8
Roti Bulat 6	4 x 4 x 3,8	9,7
Roti Bulat 7	4,5 x 4,8 x 2,5	6,7
Roti Bulat 8	4,4 x 4,4 x 4,5	10
Roti Panjang 1	6,5 x 3,6 x 4,7	21,3
Roti Panjang 2	7,5 x 3,4 x 4,9	24,2
Roti Panjang 3	7 x 3,6 x 5,2	19,5
Roti Panjang 4	6,7 x 3 x 4,9	21,6
Roti Panjang 5	7 x 3 x 4,8	17,7
Roti Panjang 6	6,8 x 3,9 x 5,1	18,4
Roti Panjang 7	6,8 x 3,5 x 4,3	23,1
Roti Panjang 8	7 x 3 x 5,1	18,8

Sehingga berat maksimal untuk benda bulat sebesar 114,5gram dan benda panjang sebesar 88 gram. Untuk ukuran maksimal lebar pada benda bulat 6,5cm serta benda panjang 5,5cm.

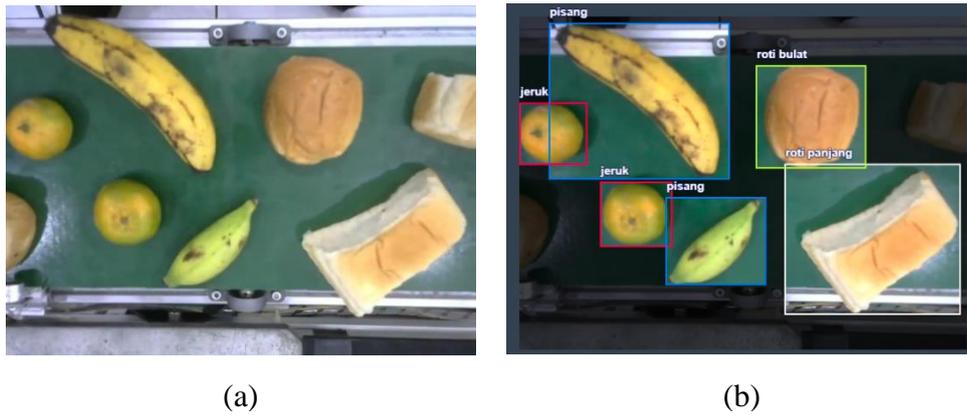
### III.1.4.2 Sistem *Computer Vision*.

#### 1. Dataset



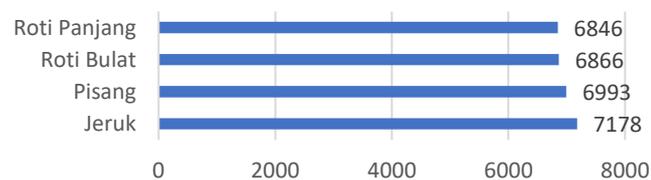
Gambar III.6 Pengelolaan Dataset

Dataset dikelola pada *website roboflow* yang memiliki fitur yang membantu dalam menangani dataset foto dalam jumlah banyak. Pengambilan bahan dataset diambil dari dataset kontributor lain di *website roboflow*, merekam video secara langsung, dan sumber video *youtube* yang sesuai dengan objek. Pengambilan dataset menggunakan video pada *roboflow* dapat di jadikan foto secara otomatis sesuai dengan pengaturan frekuensi penangkapan layar yang dilakukan. Pada foto tersebut kemudian diberi label dan *bounding box* sesuai dengan nama objek.



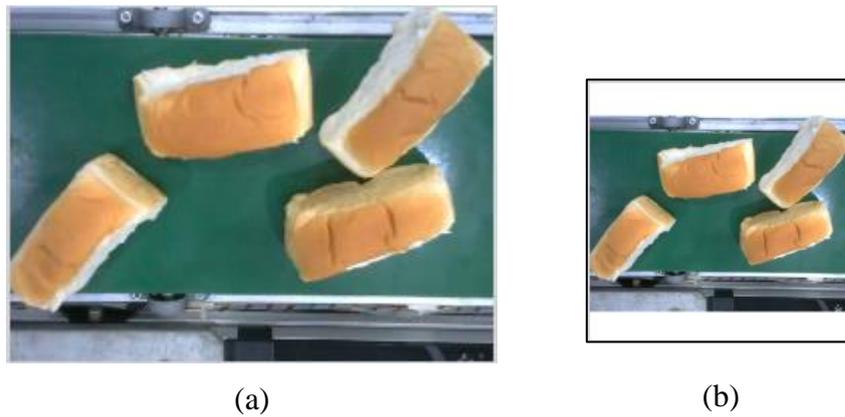
Gambar III.7 (a) sampel foto, (b) foto yang telah diberi label dan *bounding box*

Pemeriksaan ulang dilaksanakan agar setiap foto memiliki *bounding box* dan label sesuai dengan objek. Perlu diperhatikan agar pemilihan dataset diseleksi dengan baik agar dalam satu *frame bounding box* dan nama label tidak terdapat objek lain, karena berpotensi dari gangguan objek tersebut menjadi terdeteksi pada kamera. Hal ini akan berpengaruh pada pelatihan model yang menurunkan tingkat kepercayaan dalam mendeteksi hingga salah dalam pendeteksian objek. Dataset yang digunakan sebanyak 9035 foto serta pelabelan 4 objek (pisang, jeruk, roti bulat dan roti panjang) sebanyak 27,676, Jumlah pelabelan yang seimbang penting diperhatikan untuk menghindari penurunan deteksi pada label objek yang lebih rendah dan bias dalam pelatihan model karena kecenderungan mempelajari dataset yang lebih banyak. Berikut pelabelan pada setiap objek dataset ditunjukkan pada Gambar III.8.



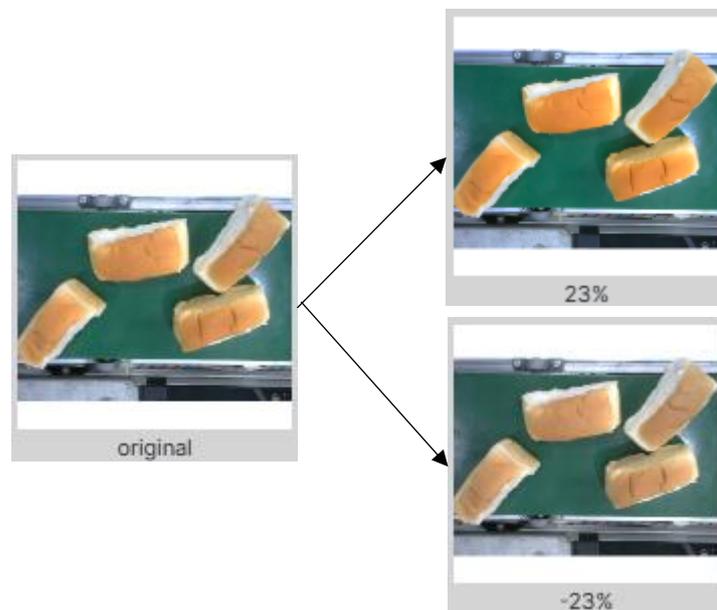
Gambar III.8 Grafik Label Setiap Objek

Nilai anotasi pada setiap objek memiliki perbedaan  $\pm 2\%$  dari rata-rata anotasi 6.970 objek. Dataset tersebut kemudian diolah agar ukuran pada setiap gambar memiliki ukuran yang sama dengan ketentuan model. Pada *SSDLite MobileNetv3* memiliki masukan gambar 320x320, perubahan dilakukan pada setiap gambar.

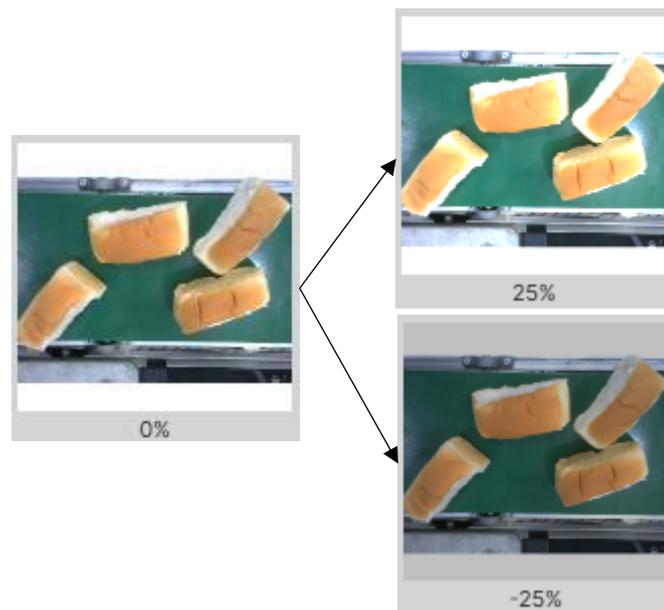


Gambar III.9 (a)720x480 pixel (b)320x320 pixel dengan penambahan latar belakang putih

Salah satu cara untuk mendapatkan dataset dengan jumlah banyak dapat dilakukan dengan *augmentasi* gambar. Cara kerja dari augmentasi pada *roboflow* dengan mengatur perubahan *brightness* dan *saturasi* gambar kemudian menyimpannya menjadi gambar yang berbeda secara acak. Dalam proses pengolahan data untuk deteksi objek, penyesuaian nilai *augmentasi* dilakukan dengan batasan  $\pm 25\%$ . Penyesuaian ini bertujuan untuk menjaga karakteristik asli dari objek agar tidak berubah drastis.



Gambar III.10 Penambahan *saturasi* pada gambar 23% dan -23%



Gambar III.11 Penambahan *brightness* 25% dan -25%

Jika nilai *augmentasi* terlalu berlebihan, dapat mengakibatkan hasil deteksi yang buruk karena model kesulitan mengenali objek yang telah mengalami perubahan signifikan dari sifat aslinya. Hal tersebut dikarenakan dalam proses *SSDLite* terdapat bagian ekstraksi fitur untuk mengenali pola, warna, bentuk dan tekstur pada setiap objek, sehingga akan lebih susah dalam memahami sifat satu objek dengan objek lainnya.

Dataset yang dihasilkan dari proses *augmentasi* ini menjadi 18,975 gambar. Pembagian dataset dilakukan dengan komposisi 79% untuk data pelatihan (*train*), 11% untuk data validasi (*valid*), dan 10% untuk data pengujian (*test*). Meskipun pembagian ini cukup umum digunakan, tidak ada ketentuan yang pasti dalam aplikasi deteksi objek. Komposisi terbaik dari data pelatihan, validasi, dan pengujian sering kali ditemukan melalui eksperimen dan pengujian pada model dengan dataset yang digunakan. Misalnya, jika pengujian hanya dilakukan pada lingkungan yang konstan, maka diperlukan porsi data pelatihan yang lebih besar untuk memastikan model dapat mengenali pola yang sama dengan dataset.

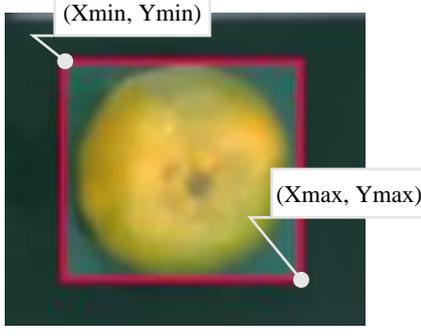
Pembagian dataset yang ideal dicapai dengan melakukan uji coba pada model. Ini melibatkan pengujian berbagai komposisi data untuk melihat mana yang memberikan kinerja terbaik pada model deteksi objek. Dataset yang akan diunduh

menggunakan format Pascal VOC (*Visual Object Classes*), yang merupakan format yang sering digunakan dalam proyek deteksi objek. Format ini menghasilkan file *Extensible Markup Language* (XML) yang berisi anotasi untuk setiap gambar. Berikut adalah contoh format file XML yang ditunjukkan pada Gambar III.12.

```

<object>
  <name>jeruk</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <occluded>0</occluded>
  <bndbox>
    <xmin>202</xmin>
    <xmax>256</xmax>
    <ymin>80</ymin>
    <ymax>130</ymax>
  </bndbox>
</object>

```

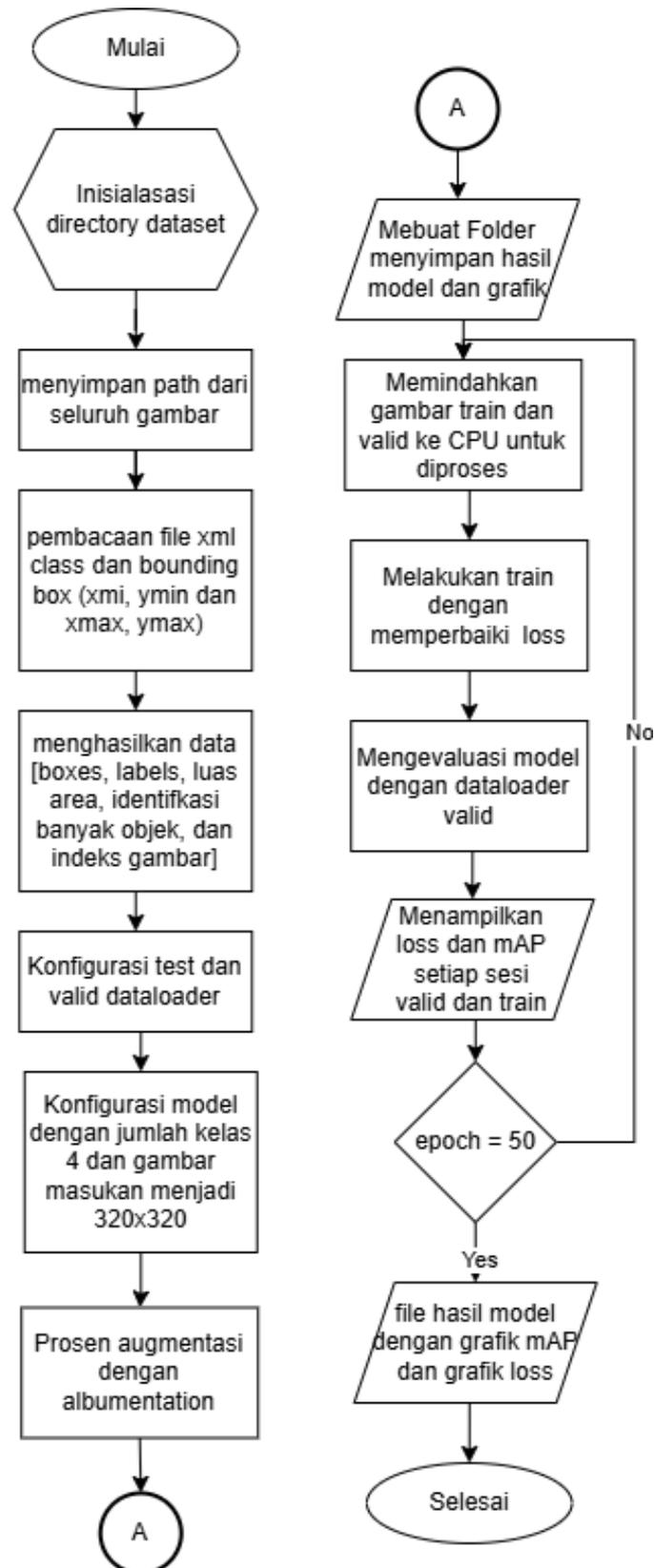


Nama objek: Jeruk  
Xmin = 202  
Xmax = 256  
Ymin = 80  
Ymax = 130

Gambar III.12 Penggunaan data pada file.xml

File XML ini mengandung berbagai informasi penting terkait gambar. Di dalamnya, terdapat elemen-elemen yang mencakup nama *file*, *path* atau jalur lokasi *file*, serta ukuran gambar. Selain itu, file ini juga berisi anotasi *bounding box* yang mendetail, yang menunjukkan lokasi serta ukuran objek di dalam gambar tersebut. Anotasi *bounding box* ini sangat penting bagi model deteksi objek, karena digunakan untuk melatih model agar dapat mempelajari dan mengenali objek-objek yang terdapat dalam gambar. Proses parsing file XML tersebut bertujuan untuk mengidentifikasi dan mengekstrak informasi tentang *bounding box* dan nama objek yang ada di gambar. Dengan demikian, informasi ini bisa digunakan untuk meningkatkan akurasi model dalam mendeteksi dan mengklasifikasikan objek. Sebagai contoh lengkap dari struktur file XML yang digunakan, Anda dapat merujuk ke Lampiran 4. Lampiran tersebut menyajikan detail komprehensif mengenai bagaimana informasi tentang *bounding box* dan objek diatur dan disimpan dalam format XML.

## 2. Pelatihan Model



Gambar III.13 Diagram Alir pelatihan Model

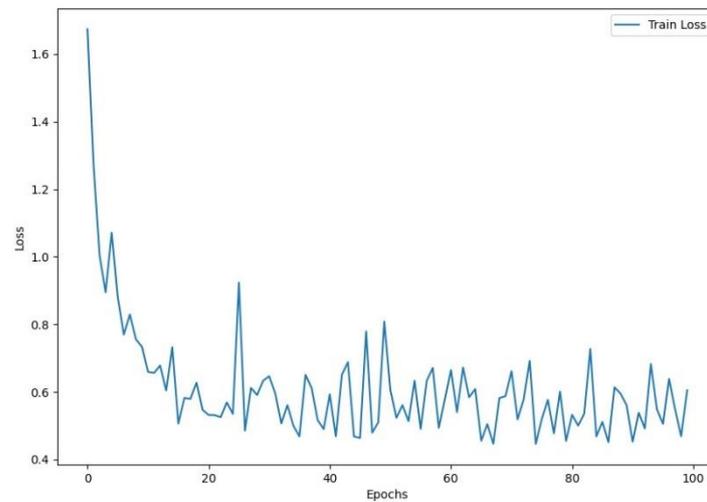
Persiapan sebelum melakukan pelatihan dengan mengorganisir file dataset berupa gambar dan file.xml. Kedua file tersebut kemudian diambil path dan nama foto hingga seluruh file tersimpan. Pada file.xml dilakukan *parsing* agar mendapatkan data bounding *box*, dan *class*,

```
"C:\Users\Ahmad\Pictures\2024-06-01-150529_mp4-0181_jpg.rf.763b7d5e84aedc3d6ab4184a266a515c.jpg "
```

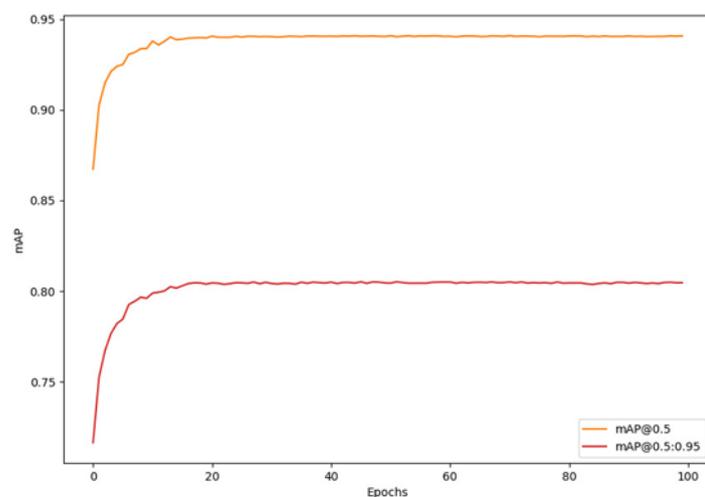
```
"C:\Users\Ahmad\Pictures\2024-06-01-150529_mp40181_jpg.rf.763b7d5e84aedc3d6ab4184a266a515c.xml"
```

Setelah mendapatkan *dataset* dibuat *dataloader* untuk memudahkan proses saat *train* dan validasi. *Dataloader* mengorganisir *batch\_size* (pembagian seluruh dataset menjadi bagian kecil, *shuffle* (untuk mengacak data agar tidak berurutan dalam pembelajaran), *num\_worker* (proses pembelajaran secara paralel). Pada model yang terdapat pada *library pytorch* dilakukan pemanggilan untuk digunakan saat *train*. Kemudian data tersebut divariasikan dengan *augmentasi* data atau penambahan data yang telah diedit *blur* dan warna dengan *library albumentation*. Hal tersebut akan menyebabkan data pelatihan yang lebih bervariasi.

Proses train dilakukan pada *website* Kaggle.com, dibandingkan dengan Google Colab, Kaggle memiliki keunggulan waktu pelatihan menggunakan GPU yang lebih banyak yaitu 30 jam. Selain itu GPU yang digunakan untuk Google Colab adalah Nvidia K80s atau GPU Tesla T4, sedangkan Kaggle NVIDIA Tesla P100 dan Dual Tesla T4 dengan komputasi yang lebih cepat.



Gambar III.14 Optimasi *Loss* pada model SSD MobileNetv3 100 *epoch*



Gambar III.15 mAP 0,5 dan 0,95 dari proses validasi 100 *epoch*

### 3. Penggunaan Model

Setelah model *SSDLite* melakukan *train* maka akan mendapatkan *file best.pth* sebagai *file* yang diintegrasikan dengan pendeteksi objek. *File.pth* berasal dari model *deep learning* yang dibuat menggunakan *pytorch*. Berikut penggunaan *best.pth* pada deteksi objek robot dengan Gambar III.16. Setelah *file.pth* telah didapatkan maka dimasukkan ke dalam algoritma agar saat memulai siaran langsung dapat memberikan prediksi pada setiap objek

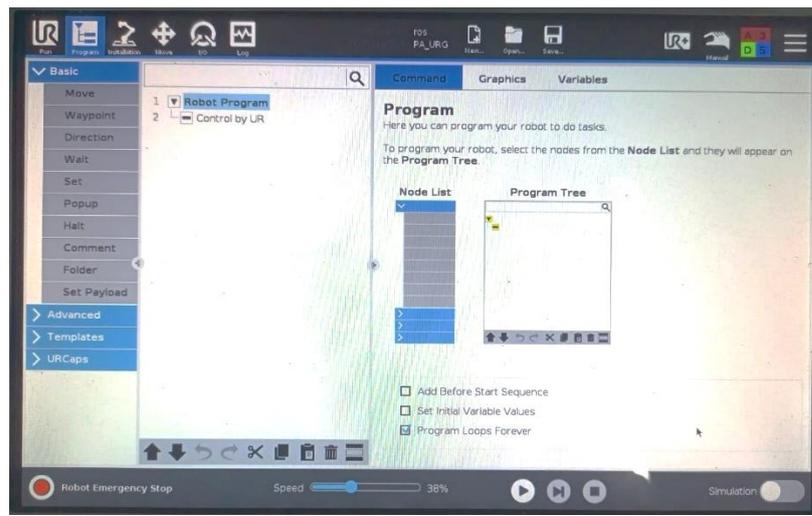


Gambar III.16 Diagram Alir Pendeteksian Objek

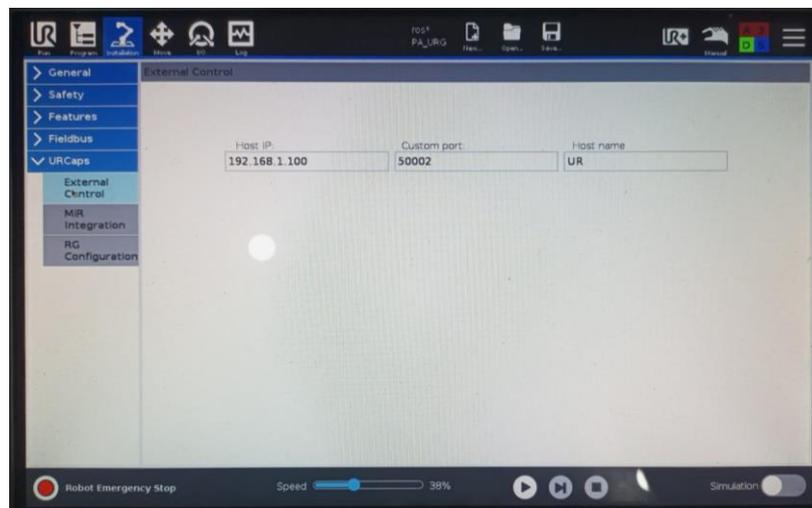
### III.1.4.3 Konfigurasi Robot.

Dalam penelitian ini sistem operasi komputer yang digunakan adalah Ubuntu 20,04 dengan ROS tipe Noetic. Pemilihan tersebut berdasarkan rekomendasi dari penyedia *driver* untuk menghubungkan ROS dan UR5e. Driver Universal Robots ROS driver diunduh dengan mengikuti panduan pada tautan *GitHub* pada **Lampiran 5**.

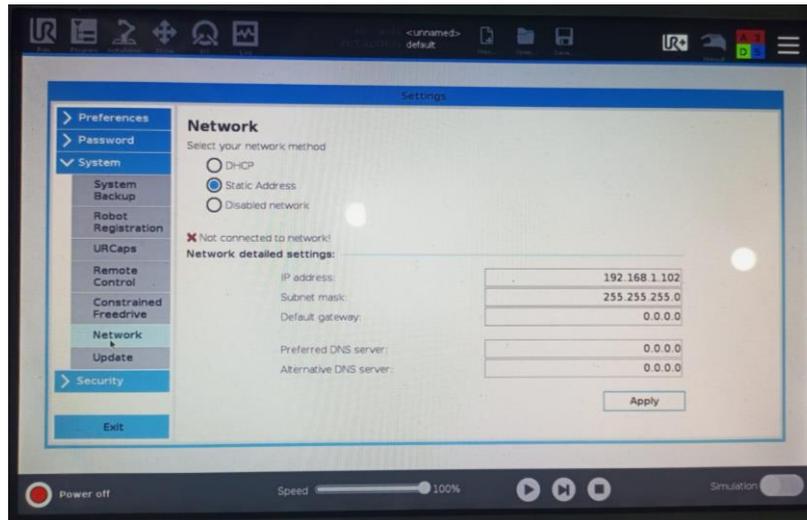
Program pada robot hanya menggunakan *external control* ditunjukkan pada Gambar III.17. Untuk pengaturan pada *pendant* robot menggunakan *external control* yang bernama “Control by UR”. *External Control* dimasukan sebagai program dari robot UR yang memiliki alamat IP 192.168.1.100 dan porta 50002 yang ditunjukkan oleh Gambar III.18. Alamat IP pada *external control* menjadi alamat dari IP laptop. Untuk alamat IP robot ditunjukkan pada Gambar III.19 dengan alamat 192.168.1.102.



Gambar III.17 Program *pendant* robot



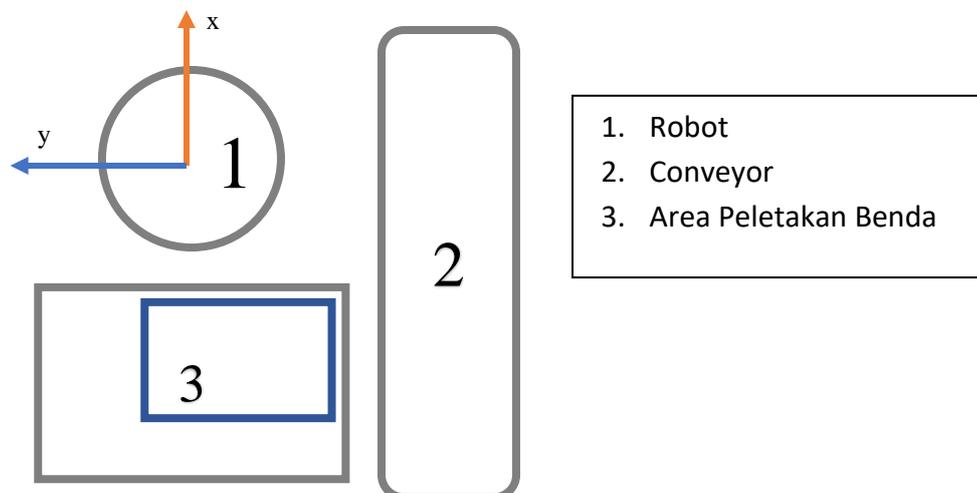
Gambar III.18 Konfigurasi *external control*



Gambar III.19 Pengaturan IP robot

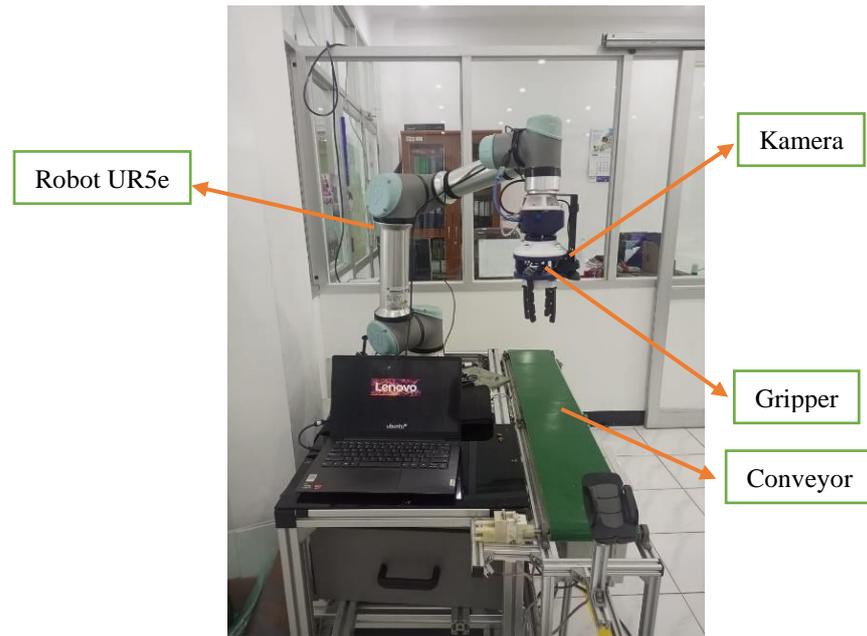
#### III.1.4.4 Perencanaan Area Robot.

Perencanaan robot disesuaikan untuk memudahkan dalam menggerakkan robot sesuai dengan koordinat *cartesian*, sehingga penyusunan lokasi dapat dilihat pada Gambar III.20. Kondisi sumbu robot dan *conveyor* dalam keadaan sejajar.



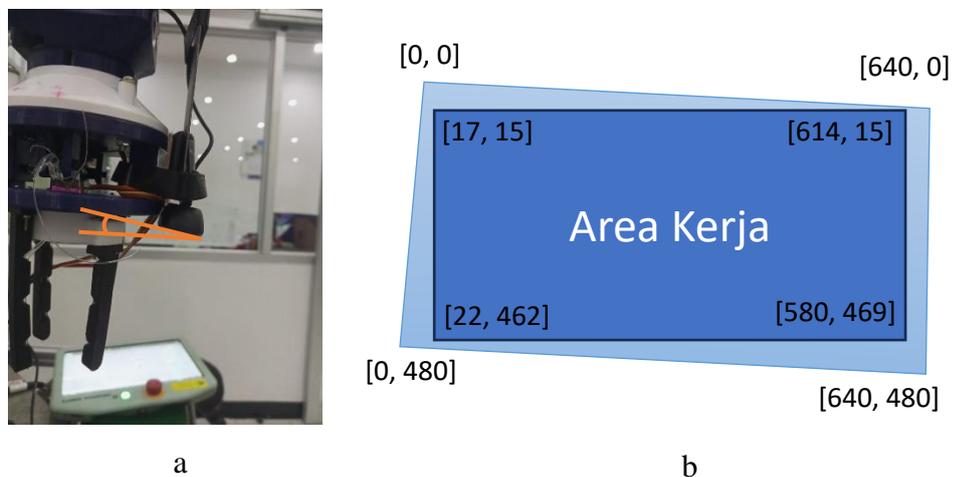
Gambar III.20 Posisi Area Robot

Dalam bentuk nyata pemasangan dari kamera berada di lengan robot setelah *gripper* ditunjukkan pada Gambar III.21.



Gambar III.21 Posisi komponen pada sistem *pick and place*

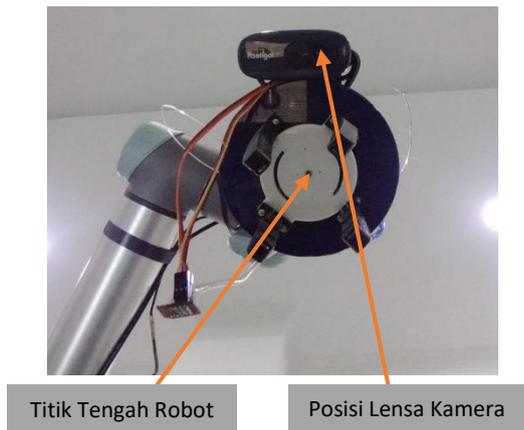
Permasalahan yang terjadi pada penempatan posisi kamera adalah sudut tangkapan kamera yang menimbulkan ukuran pixel yang tidak sama karena membentuk trapesium yang memiliki sifat melebar. Ilustrasi dapat dilihat pada Gambar III.22.



Gambar III.22 (a) posisi miring kamera, (b) perbaikan tangkapan lensa kamera dengan fungsi perspektif

#### III.1.4.5 Konversi Koordinat Kamera menuju Robot.

Perbedaan titik tengah robot dan kamera menyebabkan perbedaan jarak dalam hal penempatan pada tengah objek untuk pengambilan objek.



Gambar III.23 Jarak *error* titik tengah robot dan lensa.

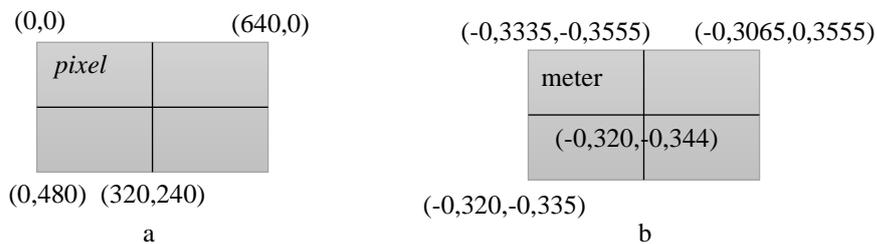
$$x \text{ error titik tengah} = 0,031 \text{ meter} \dots\dots\dots(\text{III.4})$$

$$y \text{ error titik tengah} = 0,02 \text{ meter} \dots\dots\dots(\text{III.5})$$

Penentuan titik *error* dilakukan dengan memposisikan sesuai rumus di bawah ini kemudian digerakkan menyesuaikan titik tengah dari objek, sehingga mendapatkan nilai *error* di atas.

$$x = \frac{x \text{ koordinat robot1} - x \text{ koordinat robot0}}{x \text{ koordinat kamera1} - x \text{ koordinat kamera0} - 320} \times (x \text{ koordinat objek pada kamera} - 320) - x \text{ error titik tengah} \dots\dots\dots(\text{III.6})$$

$$y = \frac{y \text{ koordinat robot1} - y \text{ koordinat robot0}}{y \text{ koordinat kamera1} - y \text{ koordinat kamera0} - 240} \times (y \text{ koordinat objek pada kamera} - 240) - y \text{ error titik tengah} \dots\dots\dots(\text{III.7})$$



Gambar III. 24 (a) Koordinat kamera (b) Koordinat robot

$$x = \frac{0,027}{640} \times (x \text{ koordinat objek pada kamera} - 320) - 0,031 \dots(\text{III.8})$$

$$y = \frac{0,021}{480} \times (y \text{ koordinat objek pada kamera} - 240) - 0,018 \dots(\text{III.9})$$

### III.1.4.6 Pergerakan *Conveyor*.

Dalam proses pengambilan objek yang bergerak, terdapat sejumlah tantangan yang perlu diatasi untuk memastikan akurasi pengambilan. Ketika titik tengah objek telah ditentukan dan pengaturan awal telah dilakukan, masalah muncul ketika objek tersebut tidak berada pada posisi tetap. Dengan kata lain, objek yang bergerak terus mengubah posisinya, sehingga membuat pengambilan data menjadi lebih kompleks.

Untuk mengatasi masalah ini, penting untuk memperhitungkan jarak tambahan yang harus ditambahkan agar sesuai dengan kecepatan gerakan objek. Ini berarti, selain hanya menentukan titik tengah objek, kita juga harus mempertimbangkan perubahan posisi objek selama periode pengambilan. Berikut adalah perhitungan jarak yang diperlukan berdasarkan kecepatan objek dan waktu pengambilan yang diperlukan:

$$S = v_{conveyor} \times t_{robot} \dots\dots\dots(III.10)$$

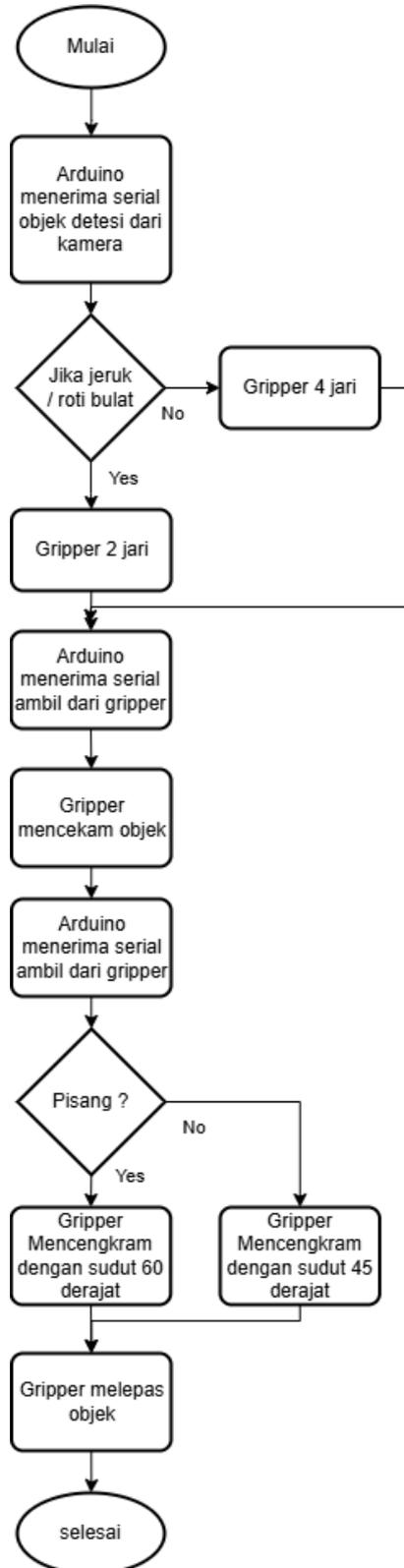
Pengukuran waktu robot melakukan pengambilan objek ditunjukkan pada Tabel III.5. Dengan pengambilan waktu secara manual menggunakan *stopwatch*.

Tabel III.5 Pengukuran waktu pengambilan objek pada robot.

Percobaan	Waktu (detik)
1	2,18
2	2,01
3	1,95
4	2,21
5	1,93
6	2,37
7	2,02
8	2,09
9	2,10
10	1,91
Rata-rata (detik)	2,07

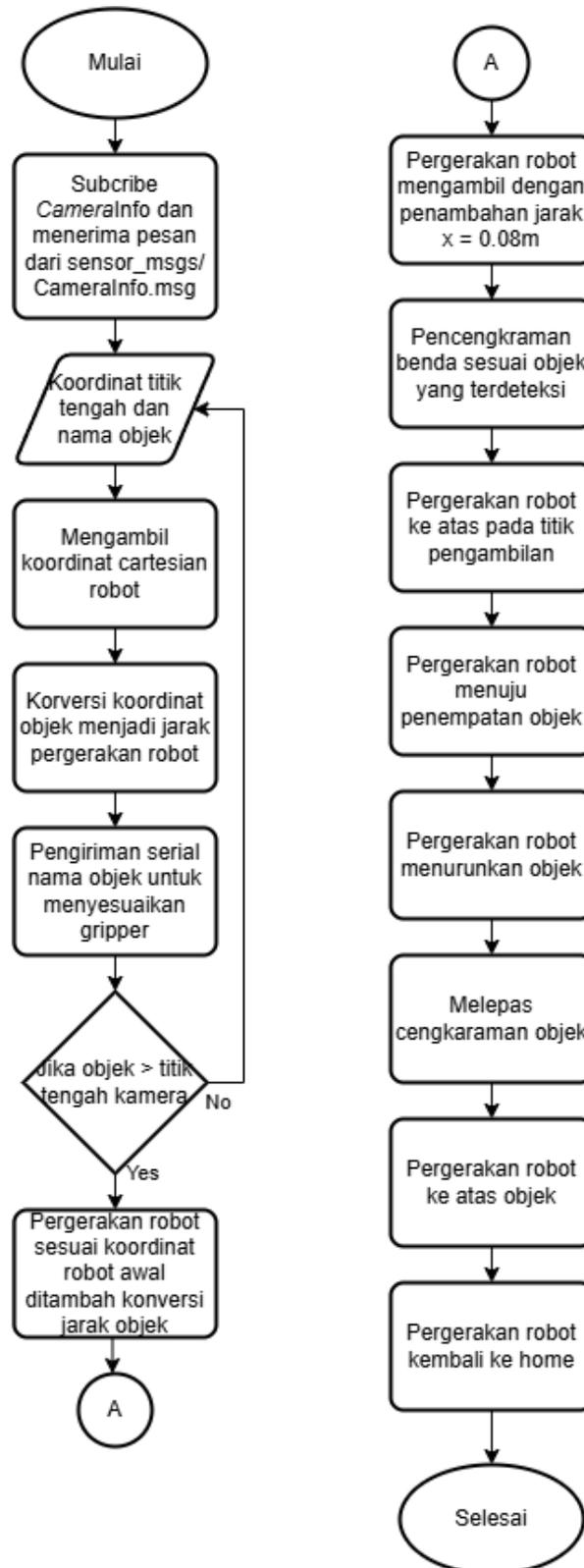
Maka penambahan jarak untuk pengambilan objek adalah  $1,29 \times 2,07 = 4,67 \text{ cm}$ .

### III.1.4.7 Pergerakan *Soft Gripper*.

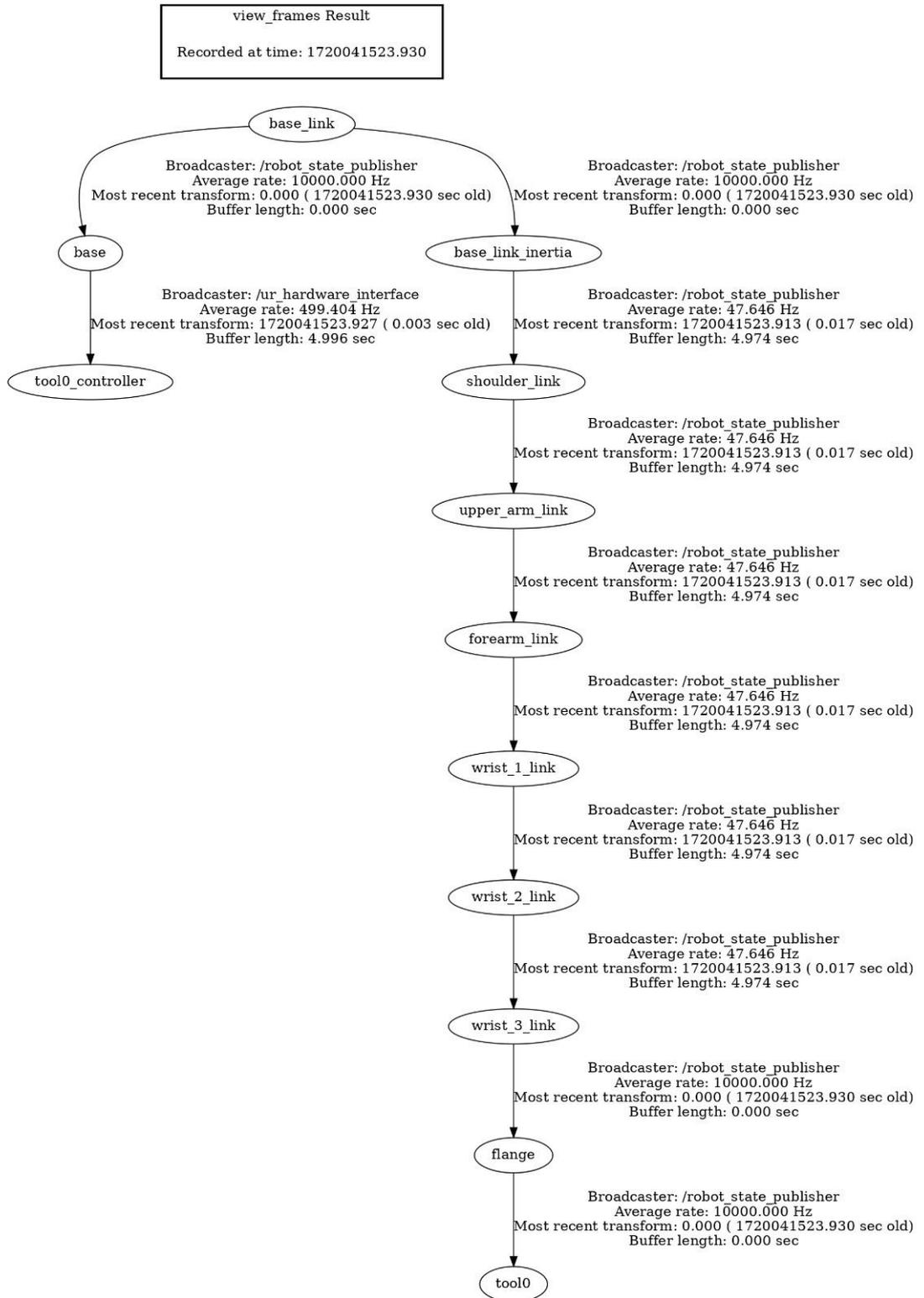


Gambar III.25 Diagram alir pergerakan *soft gripper*

III.1.4.8 Cara Kerja Robot



Gambar III.26 Diagram alir pergerakan robot



Gambar III.27 Transform ROS



### **III.1.5 Menganalisis Data.**

Tahap ini menginterpretasikan hasil eksperimen deteksi objek dengan metode *SSDLite MobileNetV3* pada poin IV.1 Hasil Pengujian *Computer Vision* serta hasil *pick and place* pada IV.2 Pengujian *Pick and Place*. Data disajikan melalui tabel atau grafik untuk mempermudah dalam melihat hasil. Hasil analisis dalam pelaksanaan penelitian dijelaskan pada BAB IV laporan Tugas Akhir

### **III.1.6 Merumuskan Kesimpulan.**

Kesimpulan adalah penjelasan dari hasil rumusan masalah yang dipilih. Rumusan kesimpulan dari penelitian dijelaskan pada BAB V penutup pada poin V.1 Kesimpulan laporan Tugas Akhir.