

MANUAL BOOK

**IF-SA02 (Illegal Fishing Smart Amphibious Aircraft): Surveillance System
for Illegal Fishing Prevention**



Nama: Agus Prayudi

NRP: 3110161058

Kelas: 4 Meka B

Daftar Isi

BAGIAN 1. SPESIFIKASI	2
[1] SPESIFIKASI SISTEM DETEKSI IF-SA02.....	2
BAGIAN 2. MANUAL OPERASIONAL	4
[1] Bagian-bagian hardware dan pengaturan	4
[2] Bagian-bagian pada software system	5
[3] Cara penggunaan software system (Fitur: mapping dan object detection)	7
[4] Cara penggunaan software system (Fitur: Plate Classification)	7
[5] Program maintenance	8
[6] Instalasi <i>Library</i>	9
BAGIAN 3. ADJUSMENT PROCEDURE	11
[1] Threshold Confidence.....	11
[2] Input Gambar untuk Pemetaan	12
[3] Registrasi Database	12
BAGIAN 4. TROUBLESHOOTING TABLE	13
[1] TROUBLESHOOTING TABLE.....	13
BAGIAN 5. OVERALL BLOCK DIAGRAM	14
[1] STATE MACHINE DIAGRAM	14
BAGIAN 6. PART GUIDE	15
[1] Part Elektrik	15
[2] Part Software	15

BAGIAN 1. SPESIFIKASI

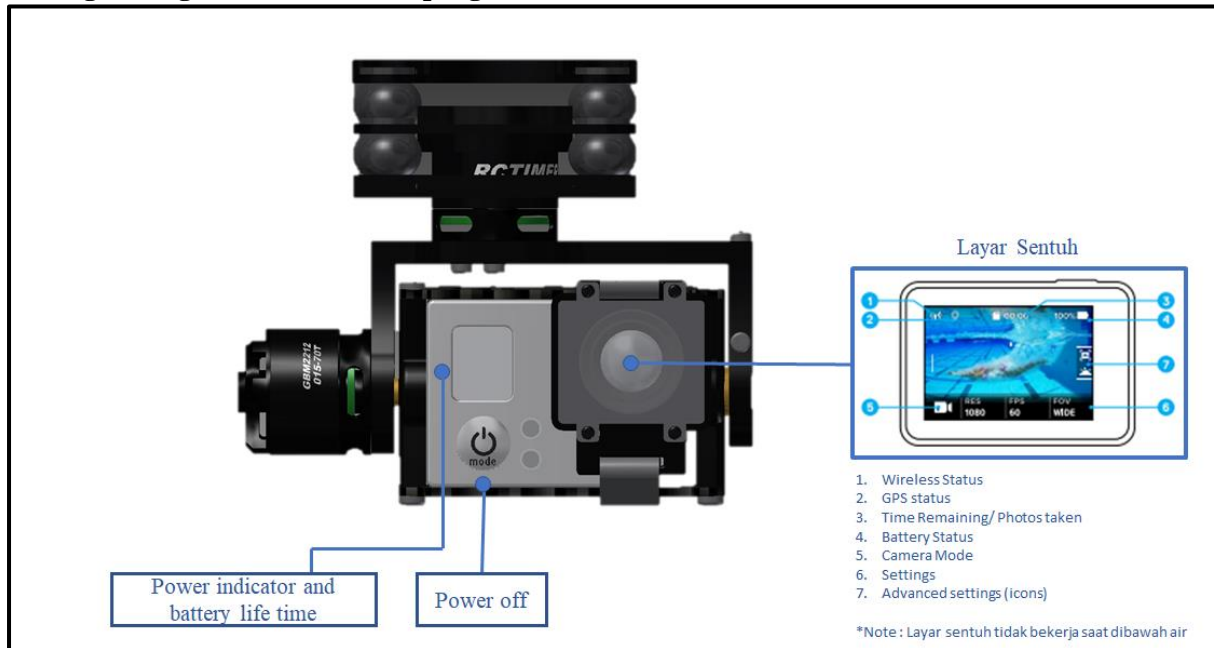
[1] SPESIFIKASI SISTEM DETEKSI IF-SA02

Item	Spesifikasi	Detail
HARDWARE		
Camera Sensor	Max Resolution	4000 x 3000
	Effective Resolution	12 Megapixels
	Water Resistance	10 m (Without Housing)
	GPS	Built-in
	Dimensions	62 x 45 x 33 mm (2.44 x 1.77 x 1.3")
Microprocessor	Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
	Memory	1GB LPDDR2 SDRAM
	Connectivity	<ul style="list-style-type: none"> 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports
	Power Requirement	<ul style="list-style-type: none"> 5V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
	Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
	SD Card Support	Micro SD format for loading operating system and data storage
Akselerator <i>Deep Learning</i>	Processor	Intel Movidius Myriad X Vision Processing Unit (VPU)
	Connectivity	USB 3.0 Type-A
	Dimensions	72.5mm X 27mm X 14mm
	Operation Temperature	0° C to 40° C
<i>Total Power Requirement</i>	-	-
<i>Total Power Consumption</i>	-	-
<i>Temperature system</i>	Operating Temperature	0–50°C

<i>Total Dimensions</i>		-
<i>Weight</i>		1.5 kg
SOFTWARE		
<i>Operating System</i>	-	Ubuntu
<i>Framework Support</i>	-	TensorFlow
<i>Accuracy</i>		90%
<i>FPS</i>	On Microprocessor	10 Fps
	On GPU	22 Fps
<i>Portable Software</i>		System on Container

BAGIAN 2. MANUAL OPERASIONAL

[1] Bagian-bagian hardware dan pengaturan



Gambar 2.1 Bagian dan pengaturan pada hardware.

Tabel 2.1 Gesture Layar sentuh pada Gopro hero 5 black.

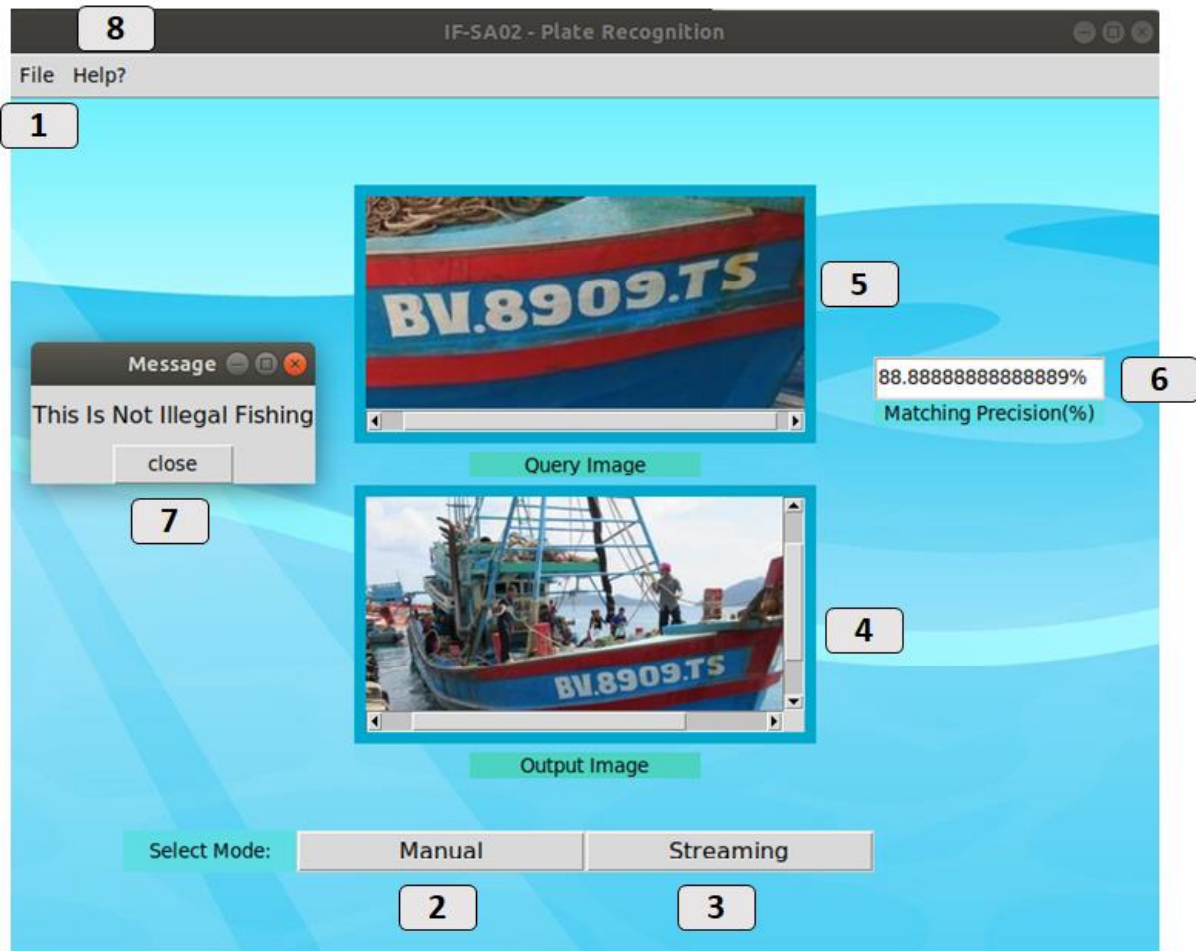
Gesture layar sentuh		
No.	Gesture	Function
1.		Nyalakan/matikan pengaturan
2.		Mengakses pengaturan <i>Exposure control</i>
3.		Mengakses pengaturan tingkat lanjut untuk mode
4.		Menampilkan media
5.		Membuka koneksi dan <i>preference menu</i> atau kembali pada menu utama

[2] Bagian-bagian pada software system



Gambar 2.2 Tampilan GUI pada fitur mapping and detection.

1. Start Mapping : Tombol untuk memulai proses mapping.
2. Detection : Tombol untuk memulai proses deteksi dari hasil pemetaan.
3. File : Fitur Menubar untuk menginputkan file-file dataset mapping.
4. Jumlah Kapal : Display untuk melihat jumlah kapal yang terdeteksi dalam gambar.
5. Date and Time: Display untuk melihat waktu pemetaan dan deteksi secara real-time.
6. Help? : Tombol untuk melihat bantuan atau instruksi penggunaan system.
7. Mode : Fitur Menubar untuk mengganti mode deteksi atau plate recognition.

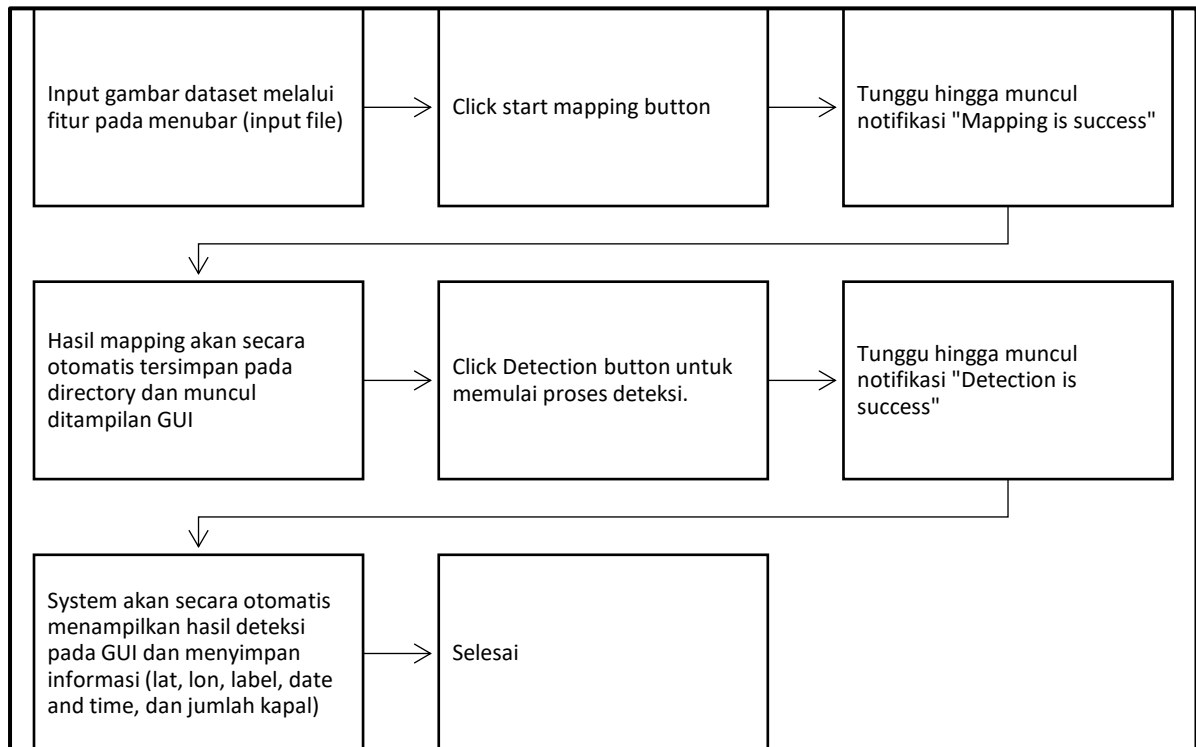


Gambar 2.3 Tampilan GUI pada fitur plate classification.

1. File : Fitur Menubar untuk menginputkan file query (gambar plat lambung hasil tangkap).
2. Manual : Fitur klasifikasi plat secara manual menggunakan data gambar.
3. Streaming : Fitur untuk klasifikasi plat secara streaming menggunakan data video.
4. Output Image : Tampilan dari hasil klasifikasi
5. Query Image : Tampilan input gambar plat lambung hasil tangkap.
6. Precision Disp.: Hasil kepresisian kecocokan gambar query dengan database.
7. Pop up msg : Notifikasi hasil klasifikasi (legal atau ilegal).
8. Help? : Fitur manual procedure penggunaan fitur klasifikasi plat kapal.

[3] Cara penggunaan software system (Fitur: mapping dan object detection)

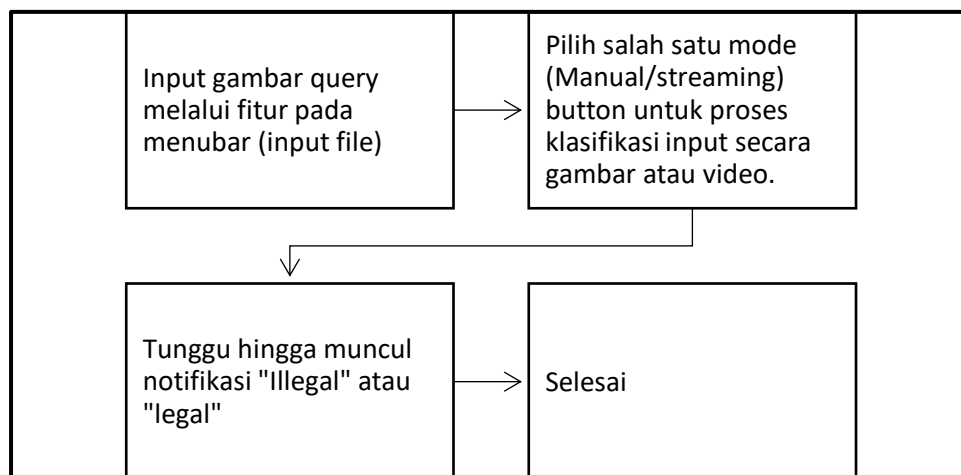
Pada bagian ini dijelaskan mengenai bagaimana prosedur penggunaan fitur mapping dan deteksi pada GUI yang telah dibuat, yakni di bawah ini:



Gambar 2.4 Cara Penggunaan fitur mapping dan deteksi pada GUI.

[4] Cara penggunaan software system (Fitur: Plate Classification)

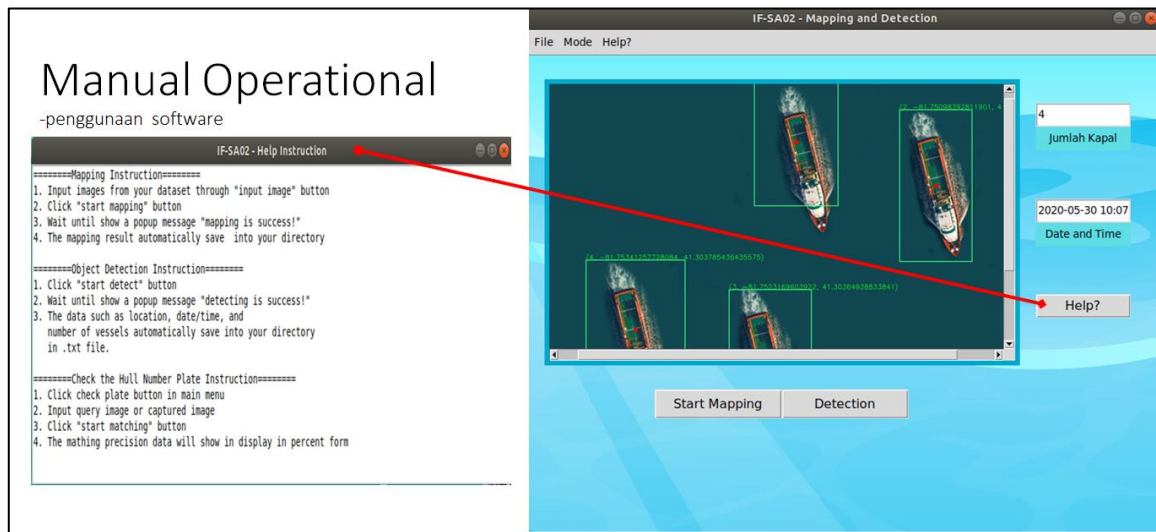
Pada bagian ini dijelaskan mengenai bagaimana prosedur penggunaan fitur mapping dan deteksi pada GUI yang telah dibuat, yakni di bawah ini:



Gambar 2.5 Cara penggunaan fitur klasifikasi plat nomor kapal pada GUI.

Note:

Cara penggunaan fitur deteksi, mapping dan plate classification juga telah terintegrasi dalam system yang telah dibuat, bisa dicek melalui fitur help button. Seperti gambar berikut ini.



Gambar 2.6 Manual prosedur yang disediakan fitur help pada GUI.

[5] Program maintenance

Program maintenance berguna untuk mengetahui bagian-bagian dari *list program* dan melakukan *troubleshooting* untuk permasalahan pada software. Berikut adalah bagian-bagian program *software* deteksi yang terbagi menjadi 5 bagian.

```
import numpy as np
import tensorflow as tf
import cv2 as cv
import glob
import os
```

Gambar 2.7 Software library yang harus disiapkan.

```
# Read the graph.
with tf.gfile.FastGFile('/home/agus/models/research/object_detection/IF-SA02/mobilenet_model/frozen_kapal.pb', 'rb') as f:
    graph_def = tf.GraphDef()
    graph_def.ParseFromString(f.read())
```

Gambar 2.8 Import deep learning model.

```
# Read and preprocess an image.
for file in glob.glob("*.jpg"):
    img = cv.imread(file)
    #img = cv.imread('/home/agus/models/research/object_detection/IF-SA02/Images/kapal1500/0001.jpg')
    #print img
    rows = img.shape[0]
    cols = img.shape[1]
    inp = cv.resize(img, (400, 400))
    inp = inp[:, :, [2, 1, 0]] # BGR2RGB
```

Gambar 2.9 Input gambar dalam program deteksi.

```
# Run the model
out = sess.run([sess.graph.get_tensor_by_name('num_detections:0'),
sess.graph.get_tensor_by_name('detection_scores:0'),
sess.graph.get_tensor_by_name('detection_boxes:0'),
sess.graph.get_tensor_by_name('detection_classes:0')],
feed_dict={'image_tensor:0': inp.reshape(1, inp.shape[0], inp.shape[1], 3)})
```

Gambar 2.10 Menjalankan model dan memberi *score* deteksi.

```
# Visualize detected bounding boxes.
num_detections = int(out[0][0])
for i in range(num_detections):
    classId = int(out[3][0][i])
    score = float(out[1][0][i])
    bbox = [float(v) for v in out[2][0][i]]
    if score > 0.5:
        x = bbox[1] * cols
        y = bbox[0] * rows
        right = bbox[3] * cols
        bottom = bbox[2] * rows
        cv.rectangle(img, (int(x), int(y)), (int(right), int(bottom)), (125, 255, 51), thickness=2)
    value = ["kapal", score, x, y, right, bottom]
    cv.imshow('TensorFlow MobileNet-SSD', img)
    cv.waitKey()
    print(value)
    f = open((file.rsplit( ".", 1 ) [ 0 ] ) + ".txt", "w")
    for listitem in value:
        f.write("%s " %listitem)
    f.close()
```

Gambar 2.11 Memberi *bounding box* dan menampilkan hasil deteksi.

[6] Instalasi *Library*

Instalasi library python pada PC (Personal Computer) atau *microprocessor*, untuk library utama yang perlu diinstal adalah Tensorflow, Numpy, OpenCV. Jika instalasi dilakukan pada GPU, maka diperlukan Cuda dan Cudnn dengan konfigurasi sebagai berikut.

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow_gpu-1.14.0	2.7, 3.3-3.7	GCC 4.8	Bazel 0.24.1	7.4	10.0
tensorflow_gpu-1.13.1	2.7, 3.3-3.7	GCC 4.8	Bazel 0.19.2	7.4	10.0
tensorflow_gpu-1.12.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.11.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.10.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0	7	9
tensorflow_gpu-1.9.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.11.0	7	9
tensorflow_gpu-1.8.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.10.0	7	9
tensorflow_gpu-1.7.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.9.0	7	9
tensorflow_gpu-1.6.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.9.0	7	9
tensorflow_gpu-1.5.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.8.0	7	9
tensorflow_gpu-1.4.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.5.4	6	8
tensorflow_gpu-1.3.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.5	6	8
tensorflow_gpu-1.2.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.5	5.1	8
tensorflow_gpu-1.1.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.2	5.1	8
tensorflow_gpu-1.0.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.2	5.1	8

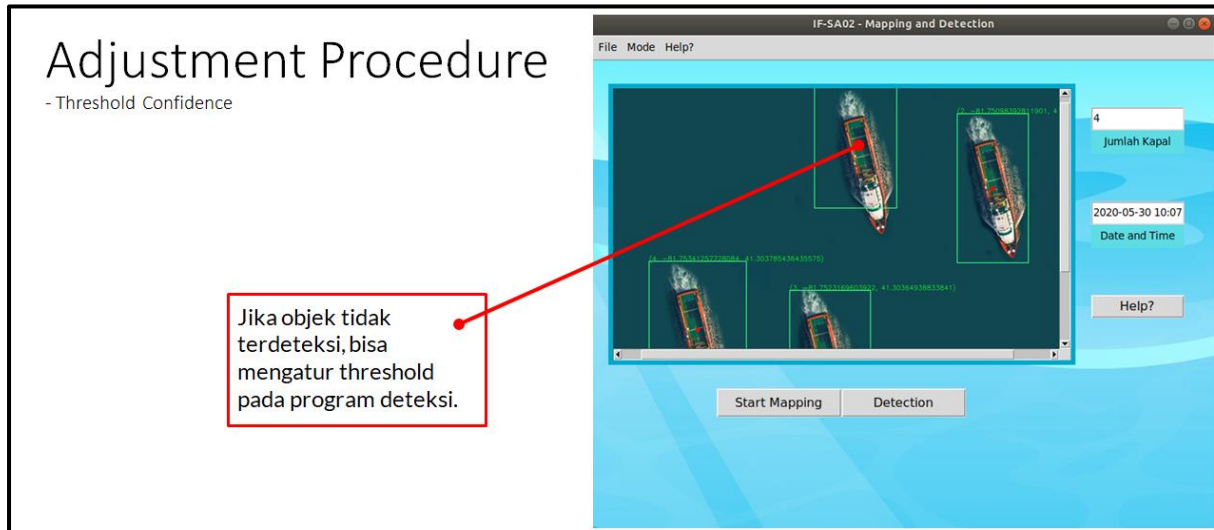
Gambar 2.12 Konfigurasi instalasi pada GPU.

Version	Python version	Compiler	Build tools
tensorflow-1.14.0	2.7, 3.3-3.7	GCC 4.8	Bazel 0.24.1
tensorflow-1.13.1	2.7, 3.3-3.7	GCC 4.8	Bazel 0.19.2
tensorflow-1.12.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0
tensorflow-1.11.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0
tensorflow-1.10.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.15.0
tensorflow-1.9.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.11.0
tensorflow-1.8.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.10.0
tensorflow-1.7.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.10.0
tensorflow-1.6.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.9.0
tensorflow-1.5.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.8.0
tensorflow-1.4.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.5.4
tensorflow-1.3.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.5
tensorflow-1.2.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.5
tensorflow-1.1.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.2
tensorflow-1.0.0	2.7, 3.3-3.6	GCC 4.8	Bazel 0.4.2

Gambar 2.13 Konfigurasi instalasi pada CPU.

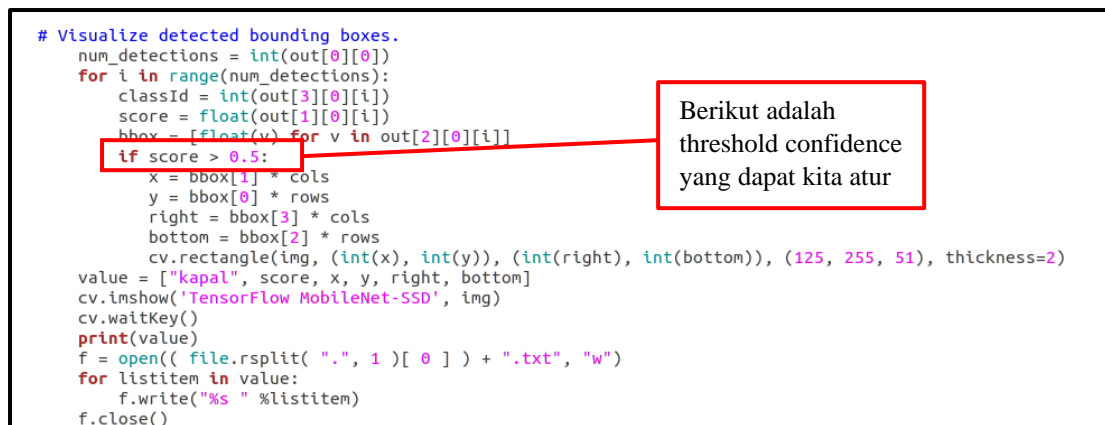
BAGIAN 3. ADJUSTMENT PROCEDURE

[1] Threshold Confidence



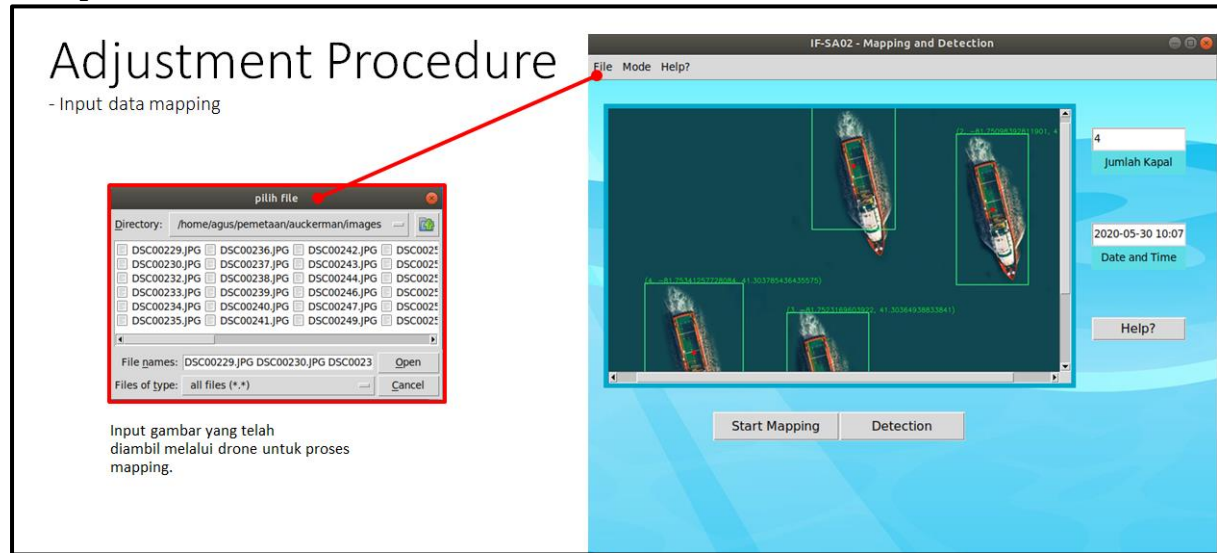
Gambar 3.1 Adjustment procedure pada threshold confidence.

Hal ini dilakukan bertujuan untuk menghindari terjadinya kesalahan deteksi atau tidak terdeteksinya objek dalam suatu gambar, sehingga diperlukan setting threshold yang sesuai agar objek dapat terdeteksi dengan baik. Threshold confidence dapat dilakukan dengan mengubah program system pada bagian if statement untuk mengatur score deteksi, sebagai contoh dapat dilihat di bawah ini.



Gambar 3.2 Adjustment pada list program objek deteksi (threshold confidence).

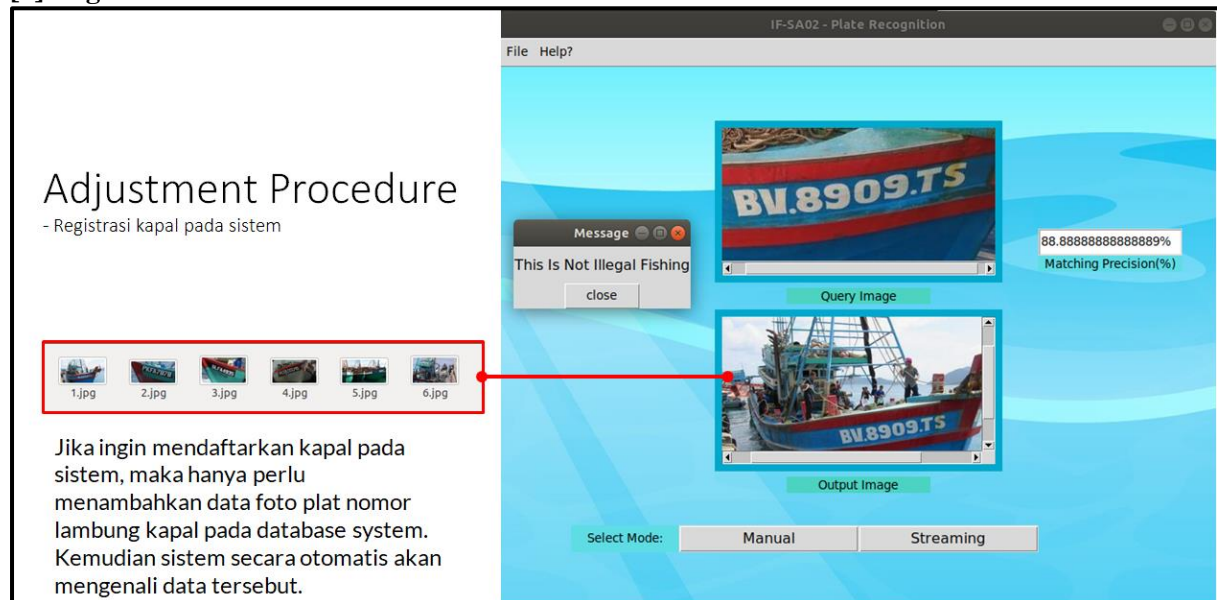
[2] Input Gambar untuk Pemetaan



Gambar 3.3 Adjustment procedure pada input file untuk mapping process.

Hal ini dilakukan bertujuan untuk menginputkan gambar dataset untuk proses pemetaan, input gambar dilakukan dengan melakukan pemilihan ekstensi file, direktori, dan jumlah dataset yang ingin dimasukkan ke dalam proses pemetaan.

[3] Registrasi Database



Gambar 3.4 Adjustment procedure pada registrasi data plat kapal ke database system.

Hal ini dilakukan bertujuan untuk mengumpulkan data gambar pada database untuk kemudian dijadikan sebagai acuan klasifikasi kapal yang terdeteksi *illegal* atau tidak. Registrasi sangat mudah untuk dilakukan, hanya dengan memasukan gambar plat lambung kapal dengan format nama file (nomor urut.jpg).

BAGIAN 4. TROUBLESHOOTING TABLE

[1] TROUBLESHOOTING TABLE

Tabel 4.1 Troubleshooting pada sistem pemetaan.

Pemetaan	
KENDALA	SOLUSI
Interface tidak menampilkan hasil pemetaan	Cek direktori hasil pemetaan, apabila tidak ada, maka coba cek aktivasi nodeODM pada docker dan pastikan library nodeODM dan pyODM telah ter- <i>install</i> .
	Jika direktori terdapat hasil pemetaan, tapi tidak muncul, maka cek kembali program GUI (IFSA_GUI2.py), pastikan program membaca direktori yang benar.

Tabel 4.2 Troubleshooting pada sistem deteks objek.

Deteksi Objek	
KENDALA	SOLUSI
Interface tidak menampilkan deteksi dari hasil pemetaan sama sekali.	Cek direktori hasil pemetaan, apabila tidak ada, maka coba cek aktivasi nodeodm pada docker
Interface tidak menampilkan hasil deteksi, tapi hasil pemetaan ada/tampil.	Cek akurasi hasil pemetaan, apabila kurang dari score threshold, maka perlu peningkatan performa model <i>deep learning</i> atau sesuaikan threshold confidence sampai sistem mampu mendeteksi objek.

Tabel 4.3 Troubleshooting pada sistem klasifikasi plat nomor lambung kapal.

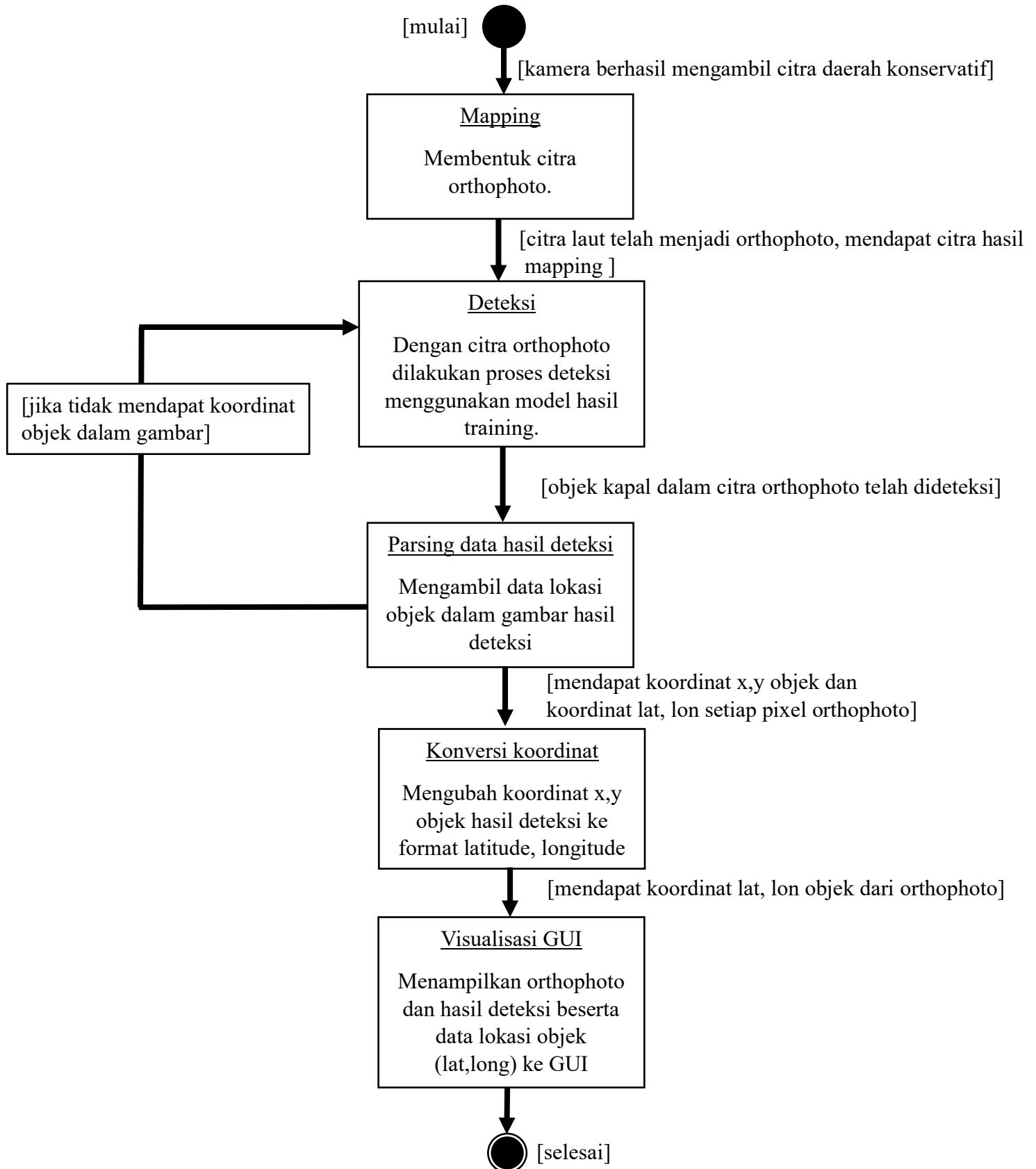
Klasifikasi Plat Nomor Lambung Kapal	
KENDALA	SOLUSI
Interface tidak menampilkan hasil klasifikasi plat nomor lambung kapal.	Cek function <code>image_matching()</code> pada program IFSA_GUI2.py, pastikan akses direktori dataset telah sesuai.
	Jika direktori telah sesuai, namun tetap tidak muncul, maka kapal tersebut tidak terdaftar dalam sistem.

Tabel 4.4 Troubleshooting pada sistem logging data.

Simpan data lokasi, waktu, dan label nomor kapal pada .txt	
KENDALA	SOLUSI
Sistem tidak menyimpan data hasil deteksi kapal pada file dengan format .txt	Cek function <code>coorxdetect()</code> pada program IFSA_GUI2.py, kemudian cek bagian list program save to .txt

BAGIAN 5. OVERALL BLOCK DIAGRAM

[1] STATE MACHINE DIAGRAM



Gambar 5.1 State machine overall system.

BAGIAN 6. PART GUIDE

[1] Part Elektrik

Tabel 6.1 Part guide elektrik.

Elektrik		
No.	Nama	Deskripsi
1.	Microcontroller	Raspberry Pi 3b+
2.	Camera	Gopro hero 5 black
3.	Gimbal	Gimbal 2 axis
4.	Waterproof cover gopro	Up to 45 meter
5.	Baterai	5v 4000mAh

[2] Part Software

Tabel 6.2 Library atau part software.

Software		
No.	Nama	Deskripsi
1.	GUI	Tkinter
2.	Objek deteksi	OpenCV, tensorflow, numpy, pandas, matplotlib
3.	Mapping	PyODM, NodeODM, Docker
4.	Plate classification	OpenCV, numpy, glob, matplotlib
5.	Bahasa pemrograman	Python3
6.	Operating system	Ubuntu