DAFTAR PUSTAKA

Admin. (2019). *siklus hidup udang vaname*. https://agrikan.id/siklus-hidup-udang-vaname/

AlexeyAB. (2019). *AlexeyAB/Yolo_mark: GUI for marking bounded boxes of objects in images for training neural network Yolo v3 and v2*. https://github.com/AlexeyAB/Yolo_mark

Ambalina, L. (2019). *What is Image Annotation? – An Intro to 5 Image Annotation Services*. https://hackernoon.com/what-is-image-annotation-an-intro-to-5-image-annotation-services-yt6n3xfj

Awalludin, E. A., Mat Yaziz, M. Y., Abdul Rahman, N. R., Yussof, W. N. J. H. W., Hitam, M. S., & T Arsad, T. N. (2019). Combination of Canny Edge Detection and Blob Processing Techniques for Shrimp Larvae Counting. *Proceedings of the 2019 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2019*, *2011*, 308–313. https://doi.org/10.1109/ICSIPA45851.2019.8977746

Awalludin, E. A., Wan Muhammad, W. N. A., Arsad, T. N. T., & Wan Yussof, W. N. J. H. (2020). Fish Larvae Counting System Using Image Processing Techniques. *Journal of Physics: Conference Series*, *1529*(5). https://doi.org/10.1088/1742-6596/1529/5/052040

Boyd, C. E., & Clay, J. (2002). Evaluation of Belize Aquaculture Ltd : A Superintensive Shrimp Aquaculture System E V A L U A T I O N O F B E L I Z E A Q U A C U L T U R E L T D : *By the Consortium*, 17.

Clarke, H., Horine, B., & Thomas-Hall, P. L. (2018). Image processing pipeline for automated larva counting. *I2MTC 2018 - 2018 IEEE International Instrumentation and Measurement Technology Conference: Discovering New Horizons in Instrumentation and Measurement, Proceedings*, 1–5. https://doi.org/10.1109/I2MTC.2018.8409867

Duraiappah, A. K., Israngkura, A., & Sae-hae, S. (2000). *Sustainable Shrimp Farming : Estimations of a Survival Function*. *31*.

Fast, A. W. ., & Lester, L. J. (1992). *Marine Shrimp Culture*. Elsevier. https://doi.org/10.1016/C2009-0-01033-2

Flores, A., Crisóstomo, P., & López, J. (2008). Peruvian scallop larvae counting system using image processing techniques. *Proceedings of the 7th International Caribbean Conference on Devices, Circuits and Systems, ICCDCS*, 8–11. https://doi.org/10.1109/ICCDCS.2008.4542660

Git. (n.d.). https://git-scm.com/

Haliman, R. W. (2005). *Udang Vannamei*. Penebar Swadaya. https://onesearch.id/Record/IOS6976.slims-6987#details

Hendarajat, E. A., Mangampa, M., & Suryanto, H. (2007). *BUDI DAYA UDANG VANNAMEI (Litopenaeus vannamei) POLA TRADISIONAL PLUS DI KABUPATEN MAROS, SULAWESI SELATAN*. *2*, *NO. 2*. https://doi.org/http://dx.doi.org/10.15578/ma.2.2.2007.67-70

Kaewchote, J., Janyong, S., & Limprasert, W. (2018). Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp. *Agriculture and Natural Resources*, *52*(4), 371–376. https://doi.org/10.1016/j.anres.2018.10.007

Khantuwan, W., & Khiripet, N. (2012). Live Shrimp Larvae Counting Method Using Co-occurrence Color Histogram. *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 1–4.

Lainez, S. M. D., & Gonzales, D. B. (2019). Automated fingerlings counting using convolutional neural network. *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, 67–72. https://doi.org/10.1109/CCOMS.2019.8821746

Loh, B. C. S., Raman, V., & Then, P. H. H. (2011). First Prototype of Aquatic Tool Kit: Towards Low-Cost Intelligent Larval Fish Counting in Hatcheries. *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, 192–195. https://doi.org/10.1109/DASC.2011.53

Low, J. (2020). *What is Image Annotation*. https://medium.com/supahands-techblog/what-is-image-annotation-caf4107601b7

Menristek. (2003). *Budidaya Udang Windu*. http//:agricta.org.com

Nguyen, K. T., Nguyen, C. N., Wang, C. Y., & Wang, J. C. (2020). Two-phase instance segmentation for whiteleg shrimp larvae counting. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, *2020-Janua*, 1–3. https://doi.org/10.1109/ICCE46568.2020.9043075

Novrihansa, R., Karnila, R., & Suparmi. (2016). *PENGARUH PENAMBAHAN KONSENTRASI GARAM BERBEDA SELAMA PEREBUSAN TERHADAP KANDUNGAN KOLESTEROL UDANG PUTIH (Penaeus indicu)*.

Nurlaela, N., Niswar, M., Nurtanio, I., Fujaya, Y., Kashihara, S., & Fall, D. (2019). Detection of Megalopa Phase Crab Larvae Using Digital Image Processing. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 269–272. https://doi.org/10.1109/ISRITI48646.2019.9034609

OEMUJATI, B. S. (1990). *Taksonomi avertebrata: pengantar praktikum laboratorium*. U.I. PRESS , 1990. https://onesearch.id/Record/IOS3239.slims-61356

Oshrimp. (2020). *5 Daerah Penghasil Udang Terbesar di Indonesia*. https://www.oshrimpseafood.com/2020/01/5-daerah-penghasil-udang-terbesar-di.html

Pokhrel, S. (2020). *Image Data Labelling and Annotation*. https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2017). Yolo V2.0. *Cvpr2017*, *April*, 187–213. http://www.worldscientific.com/doi/abs/10.1142/9789812771728_0012

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv*.

Robert B.Fisher, W. (2013). *Dictionary of Computer Vision and Image Processing, 2nd Edition*.

SEMINAR, K. (2000). The Design of Baby Fish Counter with Parallel Sensors. *Buletin Keteknikan Pertanian*, *14 NO.2*.

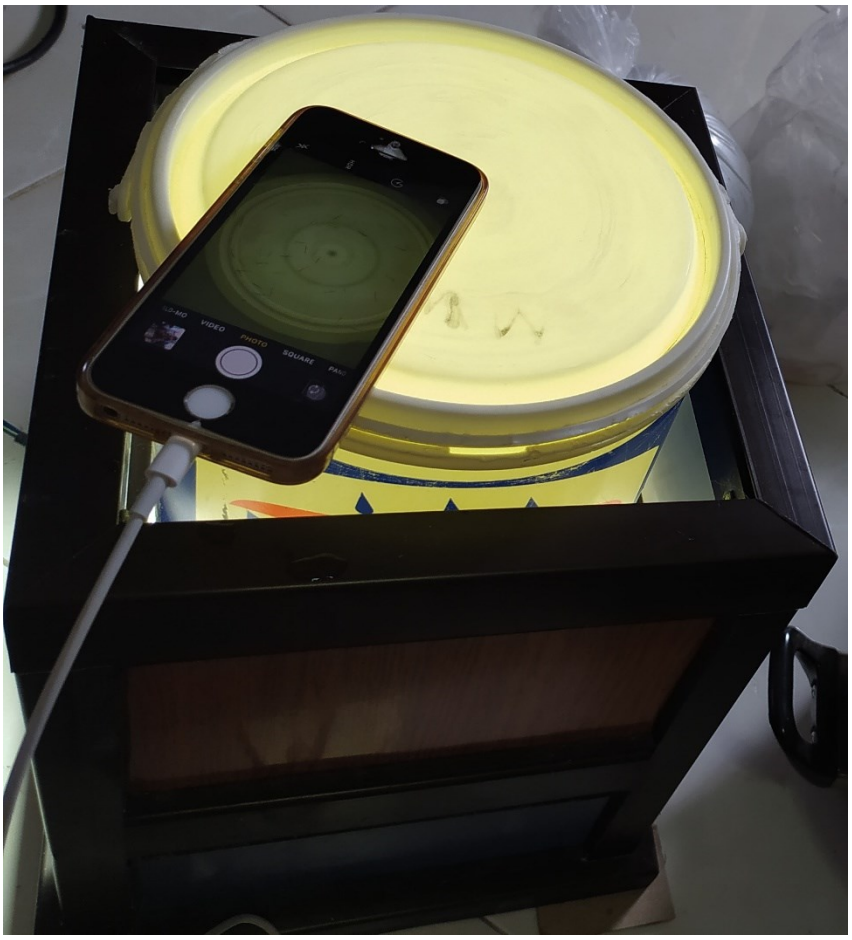
Solahudin, M., Slamet, W., & Dwi, A. S. (2018). Vaname (Litopenaeus vannamei) Shrimp Fry Counting Based on Image Processing Method. *IOP Conference Series: Earth and Environmental Science*, *147*(1). https://doi.org/10.1088/1755-1315/147/1/012014

Sterrer, W. (1986). Marine Fauna and Flora of Bermuda. A Systematic Guide to the Identification of Marine Organisms . *The Quarterly Review of Biology*, *61*(4), 566–567. https://doi.org/10.1086/415223

Tyagi, V. (2018). Understanding Digital Image Processing. *Understanding Digital Image Processing*, *November*. https://doi.org/10.1201/9781315123905

Zulfikar, wildan gayuh. (2020). *Kondisi Terkini Tambak Udang Indonesia: Trend Harga, Ekspor, dan Efek Pandemi*. https://app.jala.tech/kabar_udang/kondisi-terkini-tambak-udang-indonesia-trend-harga-ekspor-dan-efek-pandemi?redirect=https%3A%2F%2Fapp.jala.tech%2Fkabar_udang%2Fwabah-virus-corona-covid-19-tidak-mengganggu-budidaya-udang-indonesia

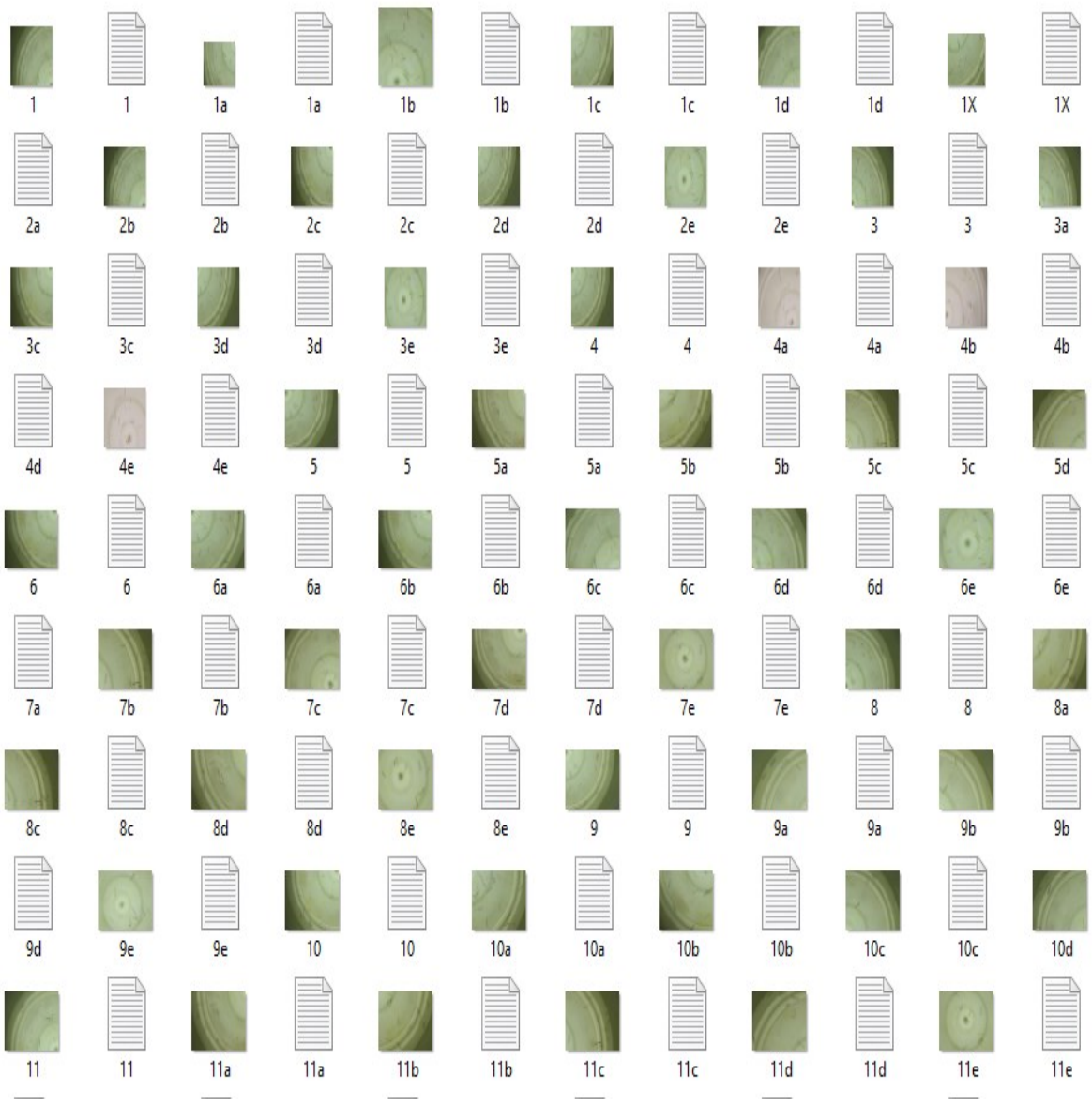
Lampiran 1. Proses Pengambilan Foto Larva Udang



Lampiran 2. Waktu Perhitungan Menggunakan Program Kepadatan < 50

Jumlah benih udang	Waktu perhitungan
1	2.54
2	2.49
3	2.6
4	2.35
15	2.7
29	2.99
30	2.61
35	2.72
37	2.65
39	3.08
42	3.14
43	2.73
44	2.88
44	2.7
44	2.9
45	3.19
45	2.85
45	2.65
46	2.34
46	2.31
46	2.3
47	2.84
49	2.98
50	2.15
50	2.15
Rata- rata	2.67

Lampiran 3. Data Training Sistem



Lampiran 4. File Konfigurasi Training small-larva-test.cfg

```
[net]
# Testing
batch=1
subdivisions=1
# Training
#batch=64
#subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches =4000
policy=steps
steps=1600,1800
scales=.1,.1


filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,
116,90,  156,198,  373,326
classes=1
num=9
```

Lampiran 5. Kode Program Pengujian Sistem

```
# import packages
import numpy as np
import argparse
import time
import cv2
import os

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
    help="minimum probability to filter weak detections")
ap.add_argument("-t", "--threshold", type=float, default=0.5,
    help="threshold when applying non-maxima suppression")
args = vars(ap.parse_args())

# load the YOLO class labels
labelsPath = "data/obj.names"
LABELS = open(labelsPath).read().strip().split("\n")

# paths to the YOLO weights and model configuration
weightsPath = "newweight/small-larva_final.weights"
configPath = "cfg/small-larva-test.cfg"

# initialize a list of colors to represent each possible class label
COLORS = (103, 220, 225)
# np.random.seed(42)
# COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
#   dtype="uint8")

# load our YOLO object detector
print("Processing...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

# load input image and grab its spatial dimensions
image = cv2.imread(args["image"])
# image = cv2.resize(image, (0,0), fx=0.7, fy=0.7)
(H, W) = image.shape[:2]

# determine only the *output* layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

# construct a blob from the input image and then perform a forward
# pass of the YOLO object detector, giving us our bounding boxes and
```

```
# associated probabilities
blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),swapRB=True,
crop=False)
net.setInput(blob)
start = time.time()
layerOutputs = net.forward(ln)
end = time.time()
print("Nilai blob: {}".format(blob.shape))

# initialize our lists of detected bounding boxes, confidences, and
# class IDs, respectively
boxes = []
confidences = []
classIDs = []

# loop over each of the layer outputs
for output in layerOutputs:
    # loop over each of the detections
    for detection in output:
        # extract the class ID and confidence (i.e., probability) form Image
        scores = detection[5:]
        classID = np.argmax(scores)
        confidence = scores[classID]
        # print ("Nilai Confidence dari object ialah :", confidence)
        # print ("Nilai ID dari Class ialah :", classID)

        # filter out weak predictions by ensuring the detected
        # probability is greater than the minimum probability
        if confidence > args["confidence"]:
            # scale the bounding box coordinates back relative to the size of
the image
            box = detection[0:4] * np.array([W, H, W, H])
            (centerX, centerY, width, height) = box.astype("int")
            # print ("Nilai dari B.Box ialah :", centerX," ", centerY, " ",
width, " ", height)

            # use the center (x, y)-coordinates to get the top and and left
            #corner of the bounding box
            x = int(centerX - (width / 2))
            y = int(centerY - (height / 2))
            print ("Nilai x dan y dari B.Box ialah :", x," ", y)

            # update our list of bounding box coordinates, confidences, and
class IDs
            boxes.append([x, y, int(width), int(height)])
            confidences.append(float(confidence))
            classIDs.append(classID)
```

```
# apply non-maxima suppression to suppress weak, overlapping bounding boxes
idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
args["threshold"])

# ensure at least one detection exists
if len(idxs) > 0:
    # loop over the indexes
    for i in idxs.flatten():
        # extract the bounding box coordinates
        (x, y) = (boxes[i][0], boxes[i][1])
        (w, h) = (boxes[i][2], boxes[i][3])

        # draw a bounding box rectangle and label on the image
        # color = [int(c) for c in COLORS[classIDs[i]]]
        cv2.rectangle(image, (x, y), (x + w, y + h), COLORS, 8)
        text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])
        print(text)
        cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,2,
COLORS, 8)
        # cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,0.2,
COLORS, 2)

# Font type
font = cv2.FONT_HERSHEY_SIMPLEX
# Font Coordinate
org = (20, 280)
# Font Size
fontScale = 10
# Font color with format (B,G,R)
color = (255, 0, 0)
# Font Thickness
thickness = 20

image = cv2.putText(image, '{} larva '.format(len(idxs)), org, font,
fontScale, color, thickness, cv2.LINE_AA)

# Write total of larva in the frame into larvaNumber.txt
with open("jumlahlarva.txt", "a") as myfile:
    myfile.write("Ada {} Larva yang terdeteksi\n".format(len(idxs)))

# show timing information on YOLO
print("YOLO took {:.6f} seconds".format(end - start))

# Save the output image
# cv2.imshow("Image.jpg", image)
cv2.imwrite("Image.jpg", image)
print ("ada", len(idxs), "Larva")
cv2.waitKey(0)
```