

DAFTAR PUSTAKA

- [1] Y. Soetrisno, "Pengembangan Budidaya Udang dan Potensi Pencemarannya Pada Perairan Pesisir," *J. Teknol. Lingkung.*, vol. 5, no. 3, pp. 187–192, 2004.
- [2] M. Junda, "Identifikasi Genus Fitoplankton Pada Salah Satu Tambak Udang Di Desa Bontomatene Kecamatan Segeri Kabupaten Pangkep," pp. 108–115, 2012.
- [3] T. Wulandari, N. Widyorini, and P. W. P, "Hubungan Pengelolaan Kualitas Air Dengan Kandungan Bahan Organik, NO₂ dan NH₃ Pada budidaya Udang Vannamei (*Litopenaeus Vannamei*) di Desa Keburuhan Purworejo," *Relig. Soc.*, vol. 4, no. 3, pp. 42–48, 2015.
- [4] G. G. Simatupang, S. Panduan, and P. Skala, "Buku Panduan Perikanan Skala Kecil Budidaya Udang Vannamei," 2014.
- [5] D. Haliman, R. W. Dan Adijaya, *Udang Vaname*. Jakarta: Penebar Swadaya, 2005.
- [6] R, Aditya. 2018. Yolo (*You Only Look Once*).
<https://machinelearning.mipa.ugm.ac.id/2018/08/05/yolo-you-only-look-once/> (diakses 27 September 2020)
- [7] R. Kusumanto and A. Tomponu, "PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB," *Stud. Environ. Sci.*, 2011.
- [8] S. Faza, "Peningkatan Kinerja dalam Pengklasifikasian Menggunakan Deep Learning," 2018.
- [9] R. K. S. C. Putri, "Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network untuk Klasifikasi Gambar," Universitas Islam Yogyakarta, 2018.
- [10] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," *J. Teknol. Indones.*, no. October, p. 3, 2017.
- [11] T. Nurhikmat, "Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek," Universitas Islam Indonesia, 2018.
- [12] A. Santoso and G. Ariyanto, "Implementasi Deep Learning Berbasis Tensorflow," *J. Emit.*, vol. 18, no. 01, pp. 22–27, 2018.
- [13] A. Santoso, G. Ariyanto, F. Learning, C. N. Network, and I. Pendahuluan, "IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK WAJAH," vol. 18, no. 01, pp. 15–21.
- [14] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.

- [15] Redmon, Joseph. 2018. *YOLO: Real-Time Object Detection*.
<https://pjreddie.com/darknet/yolo/> (diakses 13 oktober 2020)
- [16] Tim Perikanan WWF-Indonesia, “Better Management Practices Budidaya Udang Vannamei Tambak Semi Intensif dengan Instalasi Pengolahan Air Limbah (IPAL), Seri Panduan Perikanan Skala Kecil,” *J. Akuakultur Indones.*, vol. 22, pp. 1–22, 1998.

LAMPIRAN

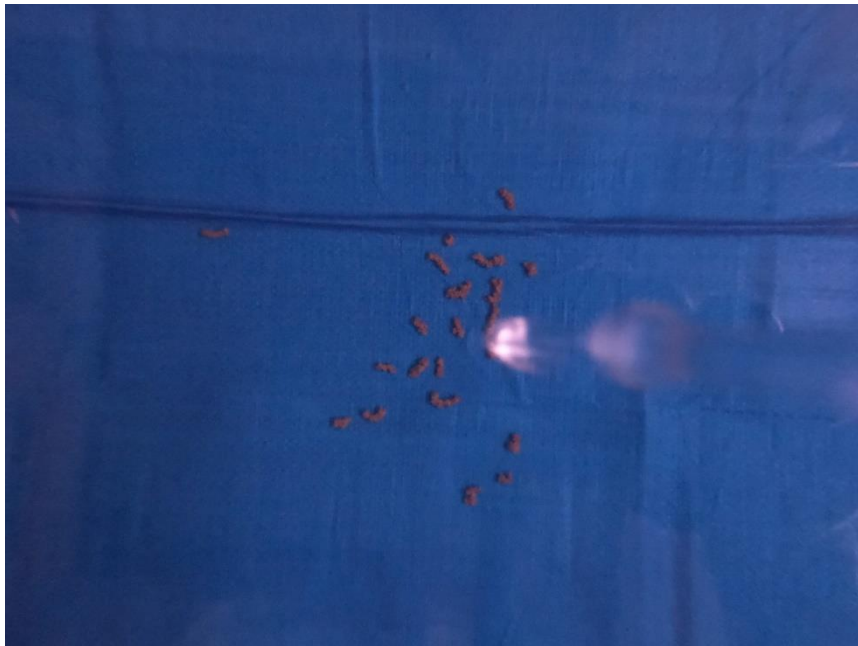
Lampiran 1. Gambar pakan udang

Gambar diambil tegak lurus

Pada ketinggian 20 cm

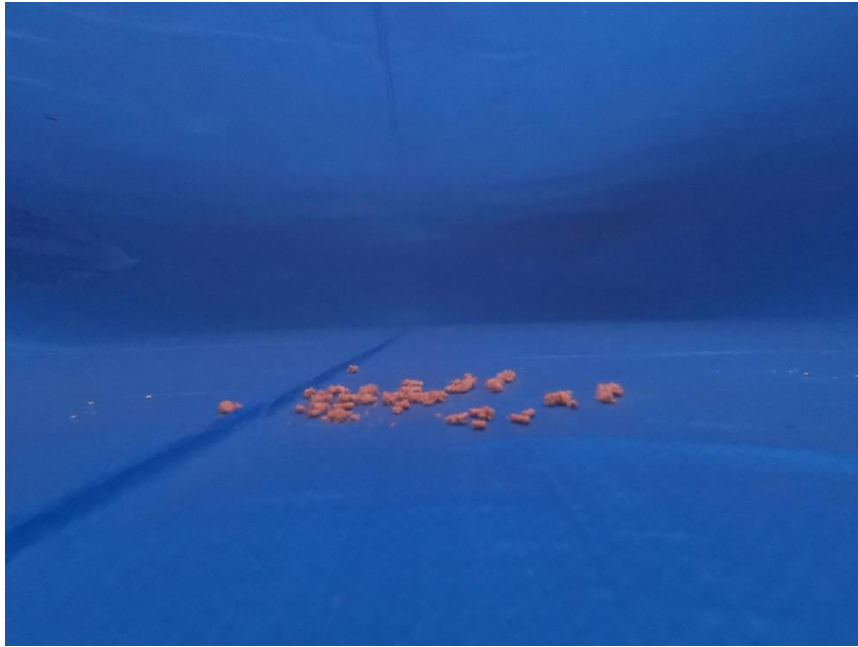


Pada ketinggian 25 cm



Gambar diambil pada jarak 25 cm

Pada ketinggian 0 cm



Pada ketinggian 10 cm



Gambar diambil pada jarak 25 cm

Pada ketinggian 20 cm



Gambar diambil pada jarak 50 cm

Pada ketinggian 0 cm



Gambar diambil pada jarak 50 cm

Pada ketinggian 10 cm



Pada ketinggian 20 cm



Gambar pakan udang dengan pasir, batu dan daun

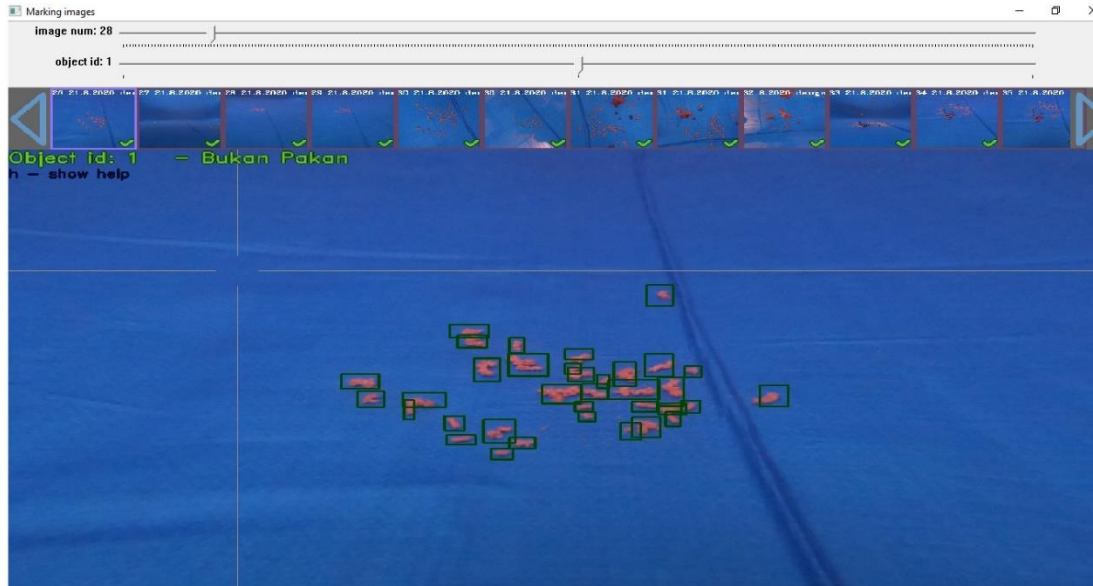
Pakan udang, pasir dan batu



Pakan udang, daun dan batu



Lampiran 2. *Labeling Image menggunakan software YOLO Mark*



Lampiran 3. Tampilan Google Collaboratory

The image displays three sequential screenshots of a Google Colab notebook interface, each titled 'pakanudg.ipynb' and last edited on September 26. The interface includes a top menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', along with 'Comment', 'Share', and settings icons. A left sidebar contains navigation icons for home, search, back, and file explorer.

First Screenshot: Shows a code cell with the following code:

```
from google.colab import drive
drive.mount('/content/drive')
```

The output below the code indicates: 'Mounted at /content/drive'.

Second Screenshot: Shows a code cell with the following code:

```
[ ] ! chmod +x ./darknet

[ ] % cd /content/drive/My Drive/darknet/
! git clone https://github.com/Jtuluran/trainingYOLO-master.git
% cd trainingYOLO-master

! make

[ ] ! wget http://pjreddie.com/media/files/darknet53.conv.74
```

The output shows an error: 'chmod: cannot access './darknet': No such file or directory'.

Third Screenshot: Shows a code cell with the following code:

```
[ ] --2020-09-25 01:47:55-- https://pjreddie.com/media/files/darknet53.conv.74
Connecting to pjreddie.com (pjreddie.com)|128.208.4.108|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 162482580 (155M) [application/octet-stream]
Saving to: 'darknet53.conv.74'

darknet53.conv.74 100%[=====] 154.96M 405KB/s in 11m 48s

2020-09-25 01:59:45 (224 KB/s) - 'darknet53.conv.74' saved [162482580/162482580]

[ ] ! ./darknet_detector train data/obj.data cfg/pakan_udg.cfg darknet53.conv.74 -dont_show
```

Lampiran 4. File Konfigurasi

```
[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 = 10000
policy=steps
steps=8000,9000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1

stride=1
pad=1
activation=leaky

[convolutional]
# Downsample
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=32
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
```

activation=leaky

[shortcut]

from=-3

activation=linear

Downsample

[convolutional]

batch_normalize=1

filters=128

size=3

stride=2

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=64

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=128

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=64

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=128

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

Downsample

[convolutional]
batch_normalize=1
filters=256
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

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

[shortcut]
from=-3
activation=linear

[convolutional]

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

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

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1

filters=256

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=128

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=256

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=128

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=256

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=128

size=1

stride=1

pad=1

activation=leaky

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

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

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

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

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

[shortcut]
from=-3
activation=linear

Downsample

[convolutional]
batch_normalize=1
filters=512

size=3
stride=2
pad=1
activation=leaky

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256

size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512

size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

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

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

Downsample

[convolutional]
batch_normalize=1

filters=1024

size=3

stride=2

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=1024

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=1024

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=1024

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

filters=1024

size=3

stride=1

pad=1

activation=leaky

[shortcut]

from=-3

activation=linear

#####

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

size=3

stride=1

pad=1

filters=1024

activation=leaky

[convolutional]

batch_normalize=1

filters=512

size=1

stride=1

pad=1

activation=leaky

[convolutional]

batch_normalize=1

size=3

stride=1

pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]

mask = 6,7,8
anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326
classes=2
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1

[route]
layers = -4

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

[upsample]
stride=2

[route]
layers = -1, 61

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

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

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

[convolutional]
batch_normalize=1
size=3
stride=1

pad=1
filters=512
activation=leaky

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

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=21
activation=linear

[yolo]

stride=1

pad=1

filters=256

activation=leaky

[convolutional]

batch_normalize=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=21

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=2

num=9

jitter=.3

ignore_thresh = .7

truth_thresh = 1

random=1