DAFTAR PUSTAKA

Ali, R., Yusro, M. M., Hitam, M. S., & Ikhwanuddin, M. (2021). Machine Learning With Multistage Classifiers For Identification Of Of Ectoparasite Infected Mud Crab Genus Scylla. *Telkomnika (Telecommunication Computing Electronics and Control)*, *19*(2), 406–413. https://doi.org/10.12928/TELKOMNIKA.v19i2.16724

Ario, R., Djunaedi, A., Pratikto, I., Subardjo, P., & Farida, F. (2019). Perbedaan Metode Mutilasi Terhadap Lama Waktu Molting Scylla serrata. *Buletin Oseanografi Marina*, *8*(2), 103. https://doi.org/10.14710/buloma.v8i2.24886

Ferdiansyah, F. F., Rahmat, B., & Yuniar, I. (2020). Klasifikasi Dan Pengenalan Objek Ikan Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, *1*(2), 522–528.

Fouad, M. M. M., Zawbaa, H. M., El-Bendary, N., & Hassanien, A. E. (2014). Automatic Nile Tilapia fish classification approach using machine learning techniques. *13th International Conference on Hybrid Intelligent Systems, HIS 2013*, 173–178. https://doi.org/10.1109/HIS.2013.6920477

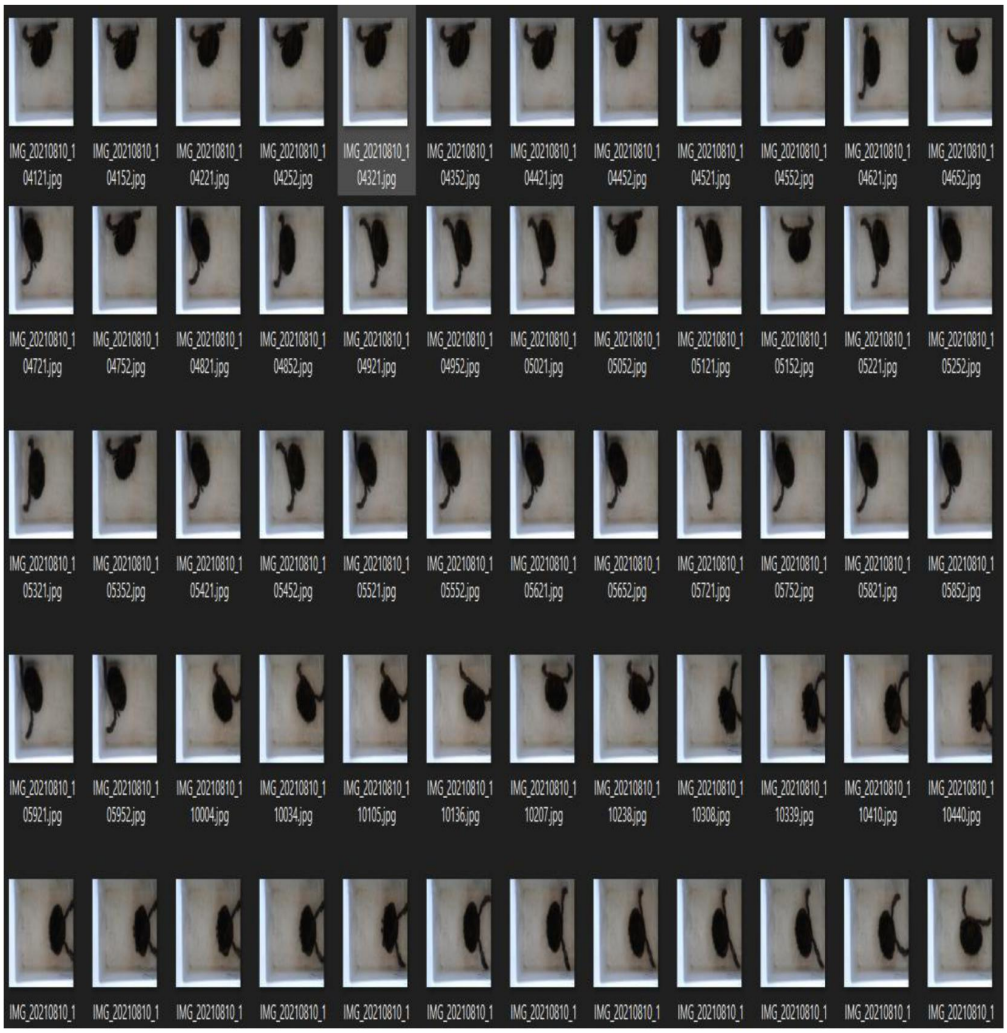
Fujaya, Y. (2011). Pertumbuhan dan molting kepiting bakau yang diberi dosis vitomolot berbeda Growth and molting of mud crab administered by different doses of vitomolt. *Journal of Indonesia Aquaculture*, *10*(1), 24–28.

Quach, L.-D., Hoang, L. Q., Trung, N. D., & Nguyen, C. N. (2020). *Towards Machine Learning Approaches To Identify Shrimp Diseases Based on Description*. https://doi.org/10.15625/vap.2019.00063

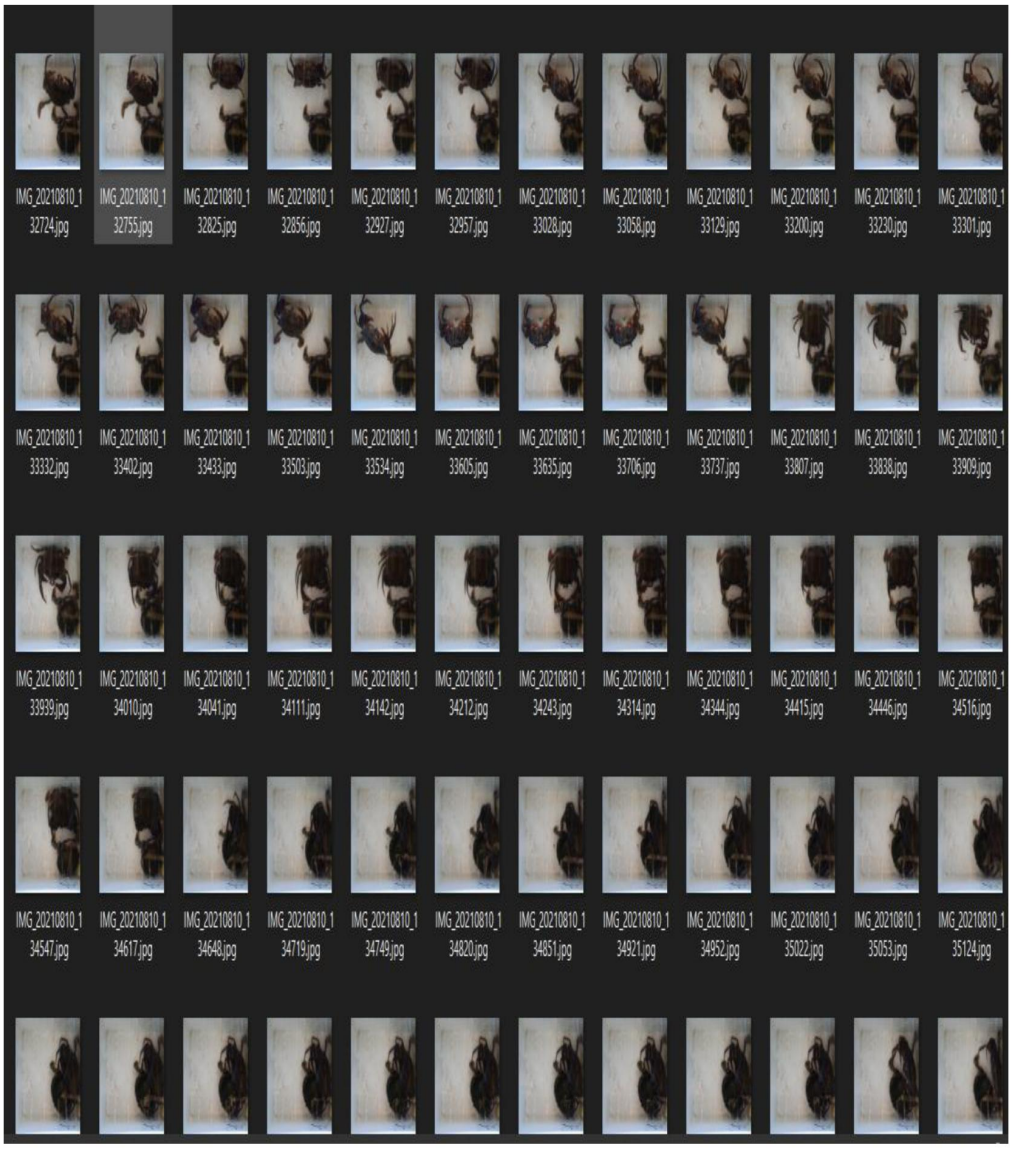
Ramadhani, M., & Murti, D. H. (2018). Klasifikasi Ikan Menggunakan Oriented Fast and Rotated Brief (Orb) Dan K-Nearest Neighbor (Knn). *JUTI: Jurnal Ilmiah Teknologi Informasi*, *16*(2), 115. https://doi.org/10.12962/j24068535.v16i2.a711

Yushinta Fujaya, Siti Aslamyah, Letty Fudjaja, N. A. (2019). *Budidaya dan Bisnis Kepiting Lunak: Stimulasi Molting Dengan Ekstrak Bayam* (Agus Wijaya (ed.)). Firstbox Media.

L A M P I R A N



Dataset Kepiting Non-Molting



Dataset Kepiting Molting

Listing program python

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import pandas as pd
from skimage.color import rgb2gray
from skimage.transform import rescale, resize, downscale_local_mean
from sklearn.model_selection import train_test_split
from skimage import data, color, feature
from skimage.feature import hog
import glob


def load_data(dataset, tipo):
    label=[]
    arr = []
    strr = "DATASET/"+dataset+"/" + tipo + "/*"
    for file_ in glob.glob(strr):
      img = cv2.imread(file_)
      arr.append(img)
      label.append(dataset)
    return arr,label

def whole_train_data(tipo):
  molting_data, molting_label = load_data('Molting', tipo)
  nonmolting_data, nonmolting_label = load_data('Non_molting', tipo)
  data =np.concatenate((molting_data,nonmolting_data))
  labels =np.concatenate((molting_label, nonmolting_label))
  return data, labels

data_train, labels_train = whole_train_data('Train')
data_test, labels_test = whole_train_data('Test')
data_train.shape, labels_train.shape

def preprocessing(arr):
    arr_prep=[]
    for i in range(arr.shape[0]):
        img=cv2.cvtColor(arr[i], cv2.COLOR_BGR2GRAY)
```

```
        img=resize(img, (72, 72),anti_aliasing=True)
        arr_prep.append(img)
    return arr_prep



data_train_p = preprocessing(data_train)
data_test_p = preprocessing(data_test)
type(data_train[0])

def ExtractHOG(img):
    ftr,_=hog(img, orientations=8, pixels_per_cell=(16, 16),
            cells_per_block=(1, 1), visualize=True, multichannel=False)
    return ftr
def preprocessing_part_two(arr):
    arr_feature=[]
    for i in range(np.shape(arr)[0]):
        arr_feature.append(ExtractHOG(arr[i]))
    return arr_feature

data_train_ftr = preprocessing_part_two(data_train_p)
data_test_ftr= preprocessing_part_two(data_test_p)

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import precision_score, \
    recall_score, confusion_matrix, classification_report, \
    accuracy_score, f1_score
model = KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='euclidean',
                    metric_params=None, n_jobs=None, n_neighbors=15,
                    weights='distance')
model.fit(data_train_ftr, labels_train)

predicted_y = model.predict(data_test_ftr)

from sklearn.metrics import classification_report
print (classification_report(labels_test, predicted_y,
target_names=["Molting", "Non_molting"]))

percentage = accuracy_score(labels_test, predicted_y)
import seaborn as sn
from sklearn.metrics import confusion_matrix, accuracy_score
res = confusion_matrix(labels_test, predicted_y)
print(" Confusion Matrix")
sn.heatmap(res, annot=True, fmt='d', xticklabels=["Molting", "Nonmolting"],
yticklabels=["Molting", "Nonmolting"])
```

```
print(f" Test Set: {len(data_test_p)}")
print(f" Accuracy = {percentage*100} %")
```