

## DAFTAR PUSTAKA

- Anggraini, N. *et al.* (2018) ‘The population and habitat of mungkus fish (Sicyopterus cynocephalus) in Bengkenang Waters South of Bengkulu’, *Journal of Physics: Conference Series*, 1116(5). Available at: <https://doi.org/10.1088/1742-6596/1116/5/052005>.
- Cakra, C. *et al.* (2022) ‘Analisis Kesegaran Ikan Mujair Dan Ikan Nila Dengan Metode Convolutional Neural Network’, *Simtek : jurnal sistem informasi dan teknik komputer*, 7(2), pp. 74–79. Available at: <https://doi.org/10.51876/simtek.v7i2.138>.
- Christiawan, M., Santoso, L.W. and Setiabudi, D.H. (2021) ‘Deteksi Tingkat Kesegaran Ikan Menggunakan Metode Convolutional Neural Network Dengan Parameter Mata dan Warna Insang’, *Jurnal Infra*, 9(2), pp. 213–219.
- Hidayat, R. *et al.* (2024) ‘Klasifikasi Jenis Buah berdasarkan Citra menggunakan Metode Ekstraksi Ciri’, *Jurnal Teknologi dan Sains Modern*, 1(1), pp. 36–48. Available at: <https://doi.org/10.69930/jtsm.v1i1.62>.
- Luthfirana, Z., Nafi, R. and Hidayatullah, M.S. (2024) ‘Deteksi Kesegaran Ikan Nila Menggunakan Convolutional Neural Networks Berbasis Citra Digital’, 02(03), pp. 608–615.
- Muchtar, M. *et al.* (2024) ‘Penerapan Metode Naïve Bayes Dalam Klasifikasi Kesegaran Ikan Berdasarkan Warna Pada Citra Area Mata’, *Jurnal*

*Informatika dan Teknik Elektro Terapan*, 12(1), pp. 611–617. Available at: <https://doi.org/10.23960/jitet.v12i1.3879>.

Noverdo, Anggi, P.S. (2023) *Populasi Ikan Mungkus Semakin Sedikit, Ini Penyebabnya*. Available at: [https://rbtv.disway.id/read/4317/populasi-ikan-mungkus-semakin-sedikit-ini-penyebabnya#google\\_vignette](https://rbtv.disway.id/read/4317/populasi-ikan-mungkus-semakin-sedikit-ini-penyebabnya#google_vignette) (Accessed: 7 October 2024).

Pujiarini, E.H. (2023) ‘Convolution Neural Network Untuk Identifikasi Tingkat Kesegaran Ikan Nila Berdasarkan Perubahan Warna Mata’, *Jurnal Khatulistiwa Informatika*, 11(1), pp. 21–25. Available at: <https://doi.org/10.31294/jki.v11i1.14305>.

Rosyani, P., Saprudin, S. and Amalia, R. (2021) ‘Klasifikasi Citra Menggunakan Metode Random Forest dan Sequential Minimal Optimization (SMO)’, *Jurnal Sistem dan Teknologi Informasi (Justin)*, 9(2), p. 132. Available at: <https://doi.org/10.26418/justin.v9i2.44120>.

Sandi, G.D.K., Syauqy, D. and Maulana, R. (2020) ‘Sistem Pendeteksi Kesegaran Ikan Bandeng Berdasarkan Bau Dan Warna Daging Berbasis Sensor Mq135 Dan Tcs3200 Dengan Metode Naïve Bayes’, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, 2548(10), p. 964X.

SITUMORANG, R.N. (2021) ‘KLASIFIKASI KESEGERAN IKAN BERDASARKAN EKSTRAKSI FITUR MENGGUNAKAN METODE K-NEAREST NEIGHBOR DAN HUE SATURATION VALUE’.

Yulianto, M.F., Hana, F.M. and Prihandono, A. (2025) ‘Penerapan Algoritma Naïve

Bayes dalam Analisis Sentimen terhadap Mobil Listrik', 22(1), pp. 109–115. Available at: <https://doi.org/10.47065/bit.v5i2.2029>.

**L**

**A**

**M**

**P**

**I**

**R**

**A**

**N**

## Naïve Bayes

### 1. Input Gambar

```
[file, path] = uigetfile({'*.jpg;*.png;*.bmp', 'Images (*.jpg, *.png, *.bmp)'});
if isequal(file, 0)
    disp('Tidak ada gambar yang dipilih.');
```

return;

```
end

inputImage = imread(fullfile(path, file));
axes(handles.axes1); % Pilih axes1 sebagai parent
imshow(inputImage); % Tampilkan gambar
title('Gambar Input');
```

handles.inputImage = imread(fullfile(path, file)); % Simpan gambar input

handles.inputGray = rgb2gray(handles.inputImage); % Simpan grayscale image

guidata(hObject, handles); % Simpan data GUI

%% 3. Load Dataset

datasetPath = 'IkanMungkus/'; % Ganti dengan folder dataset

imds = imageDatastore(datasetPath, 'IncludeSubfolders', true, 'LabelSource',  
                        'foldernames');

% Split dataset (80% training, 20% testing)

[trainImds, testImds] = splitEachLabel(imds, 0.8, 'randomized');

### 4. Ekstraksi Fitur Menggunakan GLCM

```
disp('Ekstraksi fitur dari dataset...');
```

trainFeatures = []; trainLabels = [];

testFeatures = []; testLabels = [];

for i = 1:numel(trainImds.Files)

```

img = imread(trainImds.Files{i});
imgGray = rgb2gray(img); % Ubah ke grayscale
glcm = graycomatrix(imgGray, 'Offset', [0 1]); % GLCM
stats = graycoprops(glcm, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'});

% Simpan fitur dan label
trainFeatures = [trainFeatures; stats.Contrast, stats.Correlation, stats.Energy,
    stats.Homogeneity];
trainLabels = [trainLabels; categorical(trainImds.Labels(i))];
end

for i = 1:numel(testImds.Files)
    img = imread(testImds.Files{i});
    imgGray = rgb2gray(img);
    glcm = graycomatrix(imgGray, 'Offset', [0 1]);
    stats = graycoprops(glcm, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'});

    testFeatures = [testFeatures; stats.Contrast, stats.Correlation, stats.Energy,
        stats.Homogeneity];
    testLabels = [testLabels; categorical(testImds.Labels(i))];
end

%% 5. Latih Model Naïve Bayes
disp('Melatih model Naïve Bayes...');
Mdl = fitcnb(trainFeatures, trainLabels);

%% 6. Uji Model
disp('Menguji model...');
predictedLabels = predict(Mdl, testFeatures);

```

```

% Evaluasi akurasi

accuracy = sum(predictedLabels == testLabels) / numel(testLabels) * 100;

fprintf('Akurasi model: %.2f%%\n', accuracy);

axes(handles.axes2);

cm = confusionchart(testLabels, predictedLabels);

cm.Title = 'Confusion Matrix Naïve Bayes';

cm.ColumnSummary = 'column-normalized';

cm.RowSummary = 'row-normalized';

confMat = confusionmat(testLabels, predictedLabels);

%% Hitung Precision, Recall, dan F1-Score per Kelas

precision = diag(confMat) ./ sum(confMat, 1); % True Positive / (TP + FP)

recall = diag(confMat) ./ sum(confMat, 2); % True Positive / (TP + FN)

%% Tampilkan Precision, Recall

disp('=== Precision per Kelas ===');

disp(precision);

disp('=== Recall per Kelas ===');

disp(recall);

% Tambahkan teks akurasi ke dalam figure

annotation('textbox', [0.15, 0.85, 0.1, 0.1], 'String', sprintf('Akurasi: %.2f%%',
    accuracy), 'FontSize', 12, 'FontWeight', 'bold', 'EdgeColor', 'none');

```

```
disp('Klasifikasi selesai.');
```

```
handles.Mdl = Mdl; % Simpan model ke dalam handles
```

```
handles.accuracy = accuracy; % Simpan akurasi model ke dalam handles
```

```
guidata(hObject, handles); % Simpan data GUI
```

```
Tombol Klasifikasi
```

```
handles = guidata(hObject);
```

```
% Pastikan model telah dilatih dan gambar telah dipilih
```

```
if isfield(handles, 'Mdl') && isfield(handles, 'inputGray') && isfield(handles, 'accuracy')
```

```
    disp('Melakukan klasifikasi pada gambar input...');
```

```
% Ekstraksi fitur dari gambar input
```

```
glcmInput = graycomatrix(handles.inputGray, 'Offset', [0 1]);
```

```
statsInput = graycoprops(glcmInput, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'});
```

```
inputFeatures = [statsInput.Contrast, statsInput.Correlation, statsInput.Energy, statsInput.Homogeneity];
```

```
% Prediksi kelas gambar input
```

```
predictedInputLabel = predict(handles.Mdl, inputFeatures); % Gunakan model yang tersimpan
```

```
% Tampilkan hasil klasifikasi pada axes3
```

```
axes(handles.axes3);
```

```

cla(handles.axes3, 'reset'); % Bersihkan axes sebelum update
imshow(handles.inputImage); % Tampilkan gambar input
title(sprintf('Prediksi:      %s\nAkurasi      Model:      %.2f%%',
             string(predictedInputLabel), handles.accuracy));

% Tampilkan di Command Window
disp(['Gambar diklasifikasikan sebagai: ', char(predictedInputLabel)]);
else
    disp('ERROR: Model belum dilatih atau gambar belum dipilih.');
```

end

Tombol Reset / Repeat

```

choice = questdlg('Apakah ingin mengulang klasifikasi gambar lain?', ...
                 'Reset / Ulangi', ...
                 'Ya', 'Tidak', 'Tidak');
```

switch choice

```

    case 'Ya'
        close all;
        return; % Ulang proses dari awal
    case 'Tidak'
        disp('Program selesai.');
```

return; % Keluar dari loop

end

## CNN

### 1. Input Gambar

```
[file, path] = uigetfile({'*.jpg;*.png;*.bmp'}, 'Pilih Gambar Ikan');
```

```
if isequal(file, 0)
```

```
    disp('Tidak ada gambar yang dipilih.');
```

```
    return;
```

```
end
```

```
% Simpan gambar asli
```

```
inputImageOriginal = imread(fullfile(path, file));
```

```
% Resize untuk CNN saja (tidak ditampilkan)
```

```
inputImageResized = imresize(inputImageOriginal, [64 64], 'bicubic');
```

```
% Simpan di handles
```

```
handles.inputImageOriginal = inputImageOriginal; % Untuk tampilan
```

```
handles.inputImage = inputImageResized; % Untuk CNN
```

```
guidata(hObject, handles);
```

```
% TAMPILKAN GAMBAR ASLI AGAR JERNIH
```

```
axes(handles.axes1);
```

```
imshow(inputImageOriginal, 'InitialMagnification', 'fit');
```

```
title('Gambar Input Asli');
```

### 2. Load Dataset dari Folder

```
datasetPath = 'IkanMungkus';
```

```
imds = imageDatastore(datasetPath, 'IncludeSubfolders', true, 'LabelSource',  
    'foldernames');
```

```

% Split data

[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.8, 'randomized');
inputSize = [64 64 3];

% Data Augmentasi

augmenter = imageDataAugmenter('RandRotation', [-10 10], 'RandXReflection',
    true);

augimdsTrain = augmentedImageDatastore(inputSize, imdsTrain,
    'DataAugmentation', augmenter);

augimdsValidation = augmentedImageDatastore(inputSize, imdsValidation);

% CNN Layer

numClasses = numel(categories(imdsTrain.Labels));

layers = [
    imageInputLayer(inputSize)
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    fullyConnectedLayer(64)
    reluLayer
    dropoutLayer(0.3)

```

```

    fullyConnectedLayer(numClasses)

    softmaxLayer

    classificationLayer];

% Training Options
options = trainingOptions('adam', ...
    'InitialLearnRate', 0.005, ...
    'MaxEpochs', 8, ...
    'MiniBatchSize', 64, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', augimdsValidation, ...
    'Plots', 'training-progress', ...
    'Verbose', false);

% Train Model
net = trainNetwork(augimdsTrain, layers, options);

% Evaluasi Akurasi
YPred = classify(net, augimdsValidation);
YValidation = imdsValidation.Labels;
accuracy = sum(YPred == YValidation) / numel(YValidation) * 100;
disp(['Validation Accuracy: ', num2str(accuracy, '%.2f'), '%']);

% Confusion Matrix
axes(handles.axes2);
cm = confusionchart(YValidation, YPred);

```

```

cm.Title = 'Confusion Matrix CNN - Klasifikasi Kesegaran Ikan';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';

% Simpan model ke handles
handles.net = net;
guidata(hObject, handles);
if ~isfield(handles, 'inputImageOriginal')
    msgbox('Silakan masukkan gambar terlebih dahulu!', 'Error', 'error');
    return;
end
if ~isfield(handles, 'net')
    msgbox('Silakan latih model terlebih dahulu!', 'Error', 'error');
    return;
end

% Ambil data
inputImageResized = handles.inputImage; % Untuk prediksi
inputImageOriginal = handles.inputImageOriginal; % Untuk ditampilkan jernih
net = handles.net;
inputSize = [64 64 3];

% Prediksi
inputImageDS = augmentedImageDatastore(inputSize, inputImageResized);
[predictedLabel, scores] = classify(net, inputImageDS);
predictionAccuracy = max(scores) * 100;

```

```

% TAMPILKAN GAMBAR ASLI AGAR JERNIH
axes(handles.axes3);
imshow(inputImageOriginal, 'InitialMagnification', 'fit');
title(['Prediksi: ', char(predictedLabel), ' (Akurasi: ', num2str(predictionAccuracy,
    '%.2f'), '%)']);

% Tampilkan juga di Command Window
disp(['Gambar diklasifikasikan sebagai: ', char(predictedLabel), ...
    ' dengan akurasi ', num2str(predictionAccuracy, '%.2f'), '%']);

datasetPath = 'IkanMungkus';
imds = imageDatastore(datasetPath, 'IncludeSubfolders', true, 'LabelSource',
    'foldernames');
[~, imdsValidation] = splitEachLabel(imds, 0.8, 'randomized');
augimdsValidation = augmentedImageDatastore(inputSize, imdsValidation);

% Validasi ulang untuk confusion matrix
YPred = classify(net, augimdsValidation);
YValidation = imdsValidation.Labels;
[confMat] = confusionmat(YValidation, YPred);

% Hitung Precision dan Recall
precision = diag(confMat) ./ sum(confMat, 1); % True Positive / (TP + FP)
recall = diag(confMat) ./ sum(confMat, 2);

classLabels = categories(YValidation);

```

```

% Tampilkan di Command Window
disp('=== Precision, Recall per Kelas ===');
fprintf('%-20s %-10s %-10s %-10s\n', 'Kelas', 'Precision', 'Recall');
for i = 1:numel(classLabels)
    fprintf('%-20s %-10.2f %-10.2f %-10.2f\n', ...
        classLabels{i}, precision(i)*100, recall(i)*100);
end

```

Tombol Reset / Repeat

```

choice = questdlg('Apakah ingin mengulang klasifikasi gambar lain?', ...
    'Reset / Ulangi', ...
    'Ya', 'Tidak', 'Tidak');

switch choice
    case 'Ya'
        close all;
        return; % Ulang proses dari awal
    case 'Tidak'
        disp('Program selesai.');
```

return; % Keluar dari loop

```

end

```