DAFTAR PUSTAKA

Bachtiar, M. (2021). *Pengolahan Citra Digital: Teori dan Implementasi*. Andi Publisher.

Budianita, E., Rahman, A., & Santoso, B. (2024). Analisis Performa Algoritma K-Nearest Neighbor pada Klasifikasi Data Citra. *Jurnal Teknologi Informasi Dan Sains*, *8*(2), 115–124.

Fadjeri, A. (2023). *Klasifikasi bentuk biji kopi menggunakan metode K-Nearest Neighbor (K-NN) berbasis fitur morfologis*. Jurnal Informatika, 10(2), 115–123.

Fata, N., & Avianto, D. (2024). Klasifikasi Mutu Biji Kopi Robusta Menggunakan Algoritma Naive Bayes. *Jurnal Ilmiah Teknologi Pertanian Dan Komputer*, *9*(1), 33–41.

FTNews. (2023). *Indonesia Jadi Negara Penghasil Kopi Terbesar Ketiga di Dunia*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Elsevier.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. 3rd Edition. Morgan Kaufmann.

Iswara, I. B. A. I., Anandita, I. B. G., & Dahul, M. (2024). *Comparative analysis of Naïve Bayes and K-Nearest Neighbor (KNN) algorithms in stroke classification*. Journal of Computer Networks, Architecture and High Performance Computing, 6(3), 1415–1424.

Loka, A., & Marsal, E. (2023). Perbandingan Algoritma K-NN dan Naive Bayes dalam Klasifikasi Status Gizi Balita. *Jurnal Informatika Dan Sistem Cerdas*, *7*(3), 152–160.

Nugroho, B. I., Khusni, M. W., Ananda, P. S., & Gunawan, G. (2024). *Comparison of Naïve Bayes and KNN for herbal leaf classification*. Jurnal Mandiri IT, 13(1), 18–27.

Pemungkas, R., & Asnawi, M. (2024). Analisis Naive Bayes dan K-NN dalam Klasifikasi Keluarga Miskin Berdasarkan Data Sosial Ekonomi. *Jurnal Data*

Mining Indonesia, *6*(2), 98–106.

Putra, D., & Santoso, J. (2021). "Penerapan Metode KNN untuk Klasifikasi Mutu Biji Kopi Arabika dan Robusta." *Jurnal Teknologi Pertanian Indonesia*, 12(2), 45–52.

Ramadhan, A. S., Magdalena, L., & Febima, M. (2025). *Perbandingan kinerja algoritma K-Nearest Neighbor dan Naive Bayes untuk klasifikasi loyalitas pelanggan*. RIGGS: Journal of Artificial Intelligence and Digital Business.

Rumbia, Y. N., et al. (2025). *Perbandingan metode KNN dan Naive Bayes untuk klasifikasi kelulusan mahasiswa*. Jurnal PTI, 12(1), 7–13.

Sahar, N. (2020). Analisis Kinerja Naive Bayes dan K-NN dalam Klasifikasi Penyakit Jantung. *Jurnal Ilmu Komputer Dan Aplikasi*, *14*(1), 22–30.

Sari, M., & Nugroho, A. (2022). "Pengenalan Jenis Biji Kopi Menggunakan CNN." *Jurnal Informatika dan Sains Komputer*, 14(3), 210–218.

Tapidingan, Y. C., & Paseru, D. (2023). *Comparative analysis of classification methods of KNN and Naïve Bayes to determine stress level*. Indonesian Journal of Information Systems, 2(2).

Wisdayani, D. (2019). Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas Menggunakan Algoritma K-NN dan Naive Bayes. *Jurnal Rekayasa Sistem Dan Informatika*, *5*(4), 187–196.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

LAMPIRAN

Tahapan Penelitian untuk GUI Klasifikasi Biji Kopi

```
function GUI_klasifikasi

clc; close all;

%% ================= VARIABEL GLOBAL / SHARED =================
trainPath = '';
testPath = '';
mdlKNN = [];
modelNB = [];
imgSize = [64 64];
K = 5; % Jumlah tetangga KNN
hogCellSize = [8 8]; % Default Cell Size
hasilGUI = {};
praProsesData = {};
akurasiKNN = 0;
akurasiNB = 0;
labelAsliAll = {};
predKNNAll = {};
predNBAll = {};
CM_KNN = [];
CM_NB = [];
hogFeaturesAll = []; % Variabel penyimpan nilai HOG asli

% ===== Container Detail Perhitungan (Untuk Export Excel) =====
trainFiles = {};
testFiles = {};
trainLabels = [];
Xtrain = [];
muZ = [];
sigZ = [];
knnIdx5 = [];
knnDist5 = [];
knnVoteText = {};
knnFile5Text = {};
knnLbl5Text = {};
nbLogPostMat = [];
nbPostMat = [];
nbLogPriorMat = [];
nbConstMat = [];
nbQuadMat = [];
kelas = {};
nClass = 0;

%% ======== GUI MAIN WINDOW =====
fig = uifigure('Name', 'KNN vs Naive Bayes Biji Kopi', ...
    'Position', [50 50 1350 720]);
uilabel(fig, 'Text', 'PERBANDINGAN KNN DAN NAIVE BAYES UNTUK KLASIFIKASI
BIJI KOPI', ...
    'Position', [260 675 850 30], ...
    'FontSize', 16, 'FontWeight', 'bold', ...
    'HorizontalAlignment', 'center');

%% ================ PANEL KONTROL & HASIL (KIRI) ================
```

```
panel = uipanel(fig, 'Title', 'Kontrol & Ringkasan Dataset', ...
    'Position', [20 20 300 645], ...
    'FontSize', 12, 'FontWeight', 'bold');

% Tombol & Path Data Latih
uibutton(panel, 'Text', 'Pilih Folder Data Latih', ...
    'Position', [30 575 240 35], ...
    'FontSize', 11, 'FontWeight', 'bold', ...
    'ButtonPushedFcn', @pilihLatih);
lblTrain = uilabel(panel, 'Text', 'Folder Latih: (Belum dipilih)', ...
    'Position', [20 540 260 30], 'WordWrap', 'on', ...
    'FontSize', 9, 'FontWeight', 'bold', 'FontColor', [0 0 0]);

% Tombol & Path Data Uji
uibutton(panel, 'Text', 'Pilih Folder Data Uji', ...
    'Position', [30 500 240 35], ...
    'FontSize', 11, 'FontWeight', 'bold', ...
    'ButtonPushedFcn', @pilihUji);
lblTest = uilabel(panel, 'Text', 'Folder Uji: (Belum dipilih)', ...
    'Position', [20 465 260 30], 'WordWrap', 'on', ...
    'FontSize', 9, 'FontWeight', 'bold', 'FontColor', [0 0 0]);

% ===== PILIHAN UKURAN SEL HOG =====
uilabel(panel, 'Text', 'Pilih Parameter HOG (Panjang Fitur):', ...
    'Position', [30 435 240 22], ...
    'FontSize', 10, 'FontWeight', 'bold');
ddHOG = uidropdown(panel, ...
    'Items', {'Cell Size 16x16 (324 Fitur)', 'Cell Size 8x8 (1.764
Fitur)', 'Cell Size 4x4 (8.100 Fitur)'}, ...
    'ItemsData', {[16 16], [8 8], [4 4]}, ...
    'Value', [8 8], ...
    'Position', [30 405 240 28], ...
    'FontSize', 10, 'FontWeight', 'bold');

% Tombol Eksekusi
uibutton(panel, 'Text', 'Mulai Klasifikasi', ...
    'Position', [30 350 240 45], ...
    'FontSize', 12, 'FontWeight', 'bold', ...
    'BackgroundColor', [0.2 0.6 0.9], 'FontColor', [1 1 1], ...
    'ButtonPushedFcn', @prosesKlasifikasi);
uibutton(panel, 'Text', 'Export Excel (+Gambar HOG)', ...
    'Position', [30 300 240 40], ...
    'FontSize', 11, 'FontWeight', 'bold', ...
    'BackgroundColor', [0.1 0.7 0.3], 'FontColor', [1 1 1], ...
    'ButtonPushedFcn', @exportHasil);

% ===== PANEL KARTU HASIL AKURASI =====
pCard = uipanel(panel, 'Title', 'Hasil Perbandingan Akurasi', ...
    'Position', [15 10 270 275], ...
    'FontSize', 11, 'FontWeight', 'bold', ...
    'BackgroundColor', [0.96 0.96 0.98]);
lblAkurasiKNN = uilabel(pCard, 'Text', ' Akurasi K-NN (K=5): -', ...
    'Position', [15 220 240 25], ...
    'FontSize', 10, 'FontWeight', 'bold', 'FontColor', [0.1 0.4 0.8]);
```

```
lblDetailKNN = uilabel(pCard, 'Text', 'Benar: - | Salah: -', ...
    'Position', [25 200 230 20], ...
    'FontSize', 9, 'FontColor', [0.3 0.3 0.3]);
lblAkurasiNB = uilabel(pCard, 'Text', ' Akurasi Naive Bayes: -', ...
    'Position', [15 165 240 25], ...
    'FontSize', 10, 'FontWeight', 'bold', 'FontColor', [0.8 0.3 0.1]);
lblDetailNB = uilabel(pCard, 'Text', 'Benar: - | Salah: -', ...
    'Position', [25 145 230 20], ...
    'FontSize', 9, 'FontColor', [0.3 0.3 0.3]);
lblSelisih = uilabel(pCard, 'Text', ' Selisih Akurasi: -', ...
    'Position', [15 110 240 25], ...
    'FontSize', 10, 'FontWeight', 'bold', 'FontColor', [0.2 0.2 0.2]);
lblInfoHOG = uilabel(pCard, 'Text', ' Panjang Fitur HOG: -', ...
    'Position', [15 75 240 25], ...
    'FontSize', 10, 'FontWeight', 'bold', 'FontColor', [0.5 0.2 0.6]);
lblMetodeTerbaik = uilabel(pCard, 'Text', 'Model Terbaik: -', ...
    'Position', [15 25 240 30], ...
    'FontSize', 11, 'FontWeight', 'bold', 'FontColor', [0.1 0.6 0.2]);

%% ===== TABEL HASIL UTAMA =================
tblHasil = uitable(fig, ...
    'Position', [340 20 980 645], ...
    'ColumnName', {'File', 'Label Asli', 'Prediksi KNN', 'Prediksi Naive
Bayes'}, ...
    'FontSize', 11);

%% ================= CALLBACK SELEKSI FOLDER =================
function pilihLatih(~,~)
    p = uigetdir(pwd, 'Pilih Folder Data Latih');
    if p == 0, return; end
    trainPath = p;
    lblTrain.Text = ['Folder Latih: ', p];
end

function pilihUji(~,~)
    p = uigetdir(pwd, 'Pilih Folder Data Uji');
    if p == 0, return; end
    testPath = p;
    lblTest.Text = ['Folder Uji: ', p];
end

%% ================= PROSES UTAMA KLASIFIKASI =================
function prosesKlasifikasi(~,~)
    if isempty(trainPath) || isempty(testPath)
        uialert(fig, 'Pilih folder data latih dan data uji terlebih
dahulu!', 'Peringatan');
        return;
    end
    hogCellSize = ddHOG.Value;

    %% ===== 1. PEMROSESAN DATA LATIH =====
    imdsTrain = imageDatastore(trainPath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');
    trainFiles = imdsTrain.Files;
```

```
    Y_lowercase = categorical(lower(string(imdsTrain.Labels)));
    Y = Y_lowercase;
    trainLabels = Y;
    kelas = categories(Y);
    nClass = numel(kelas);
    nTrain = numel(imdsTrain.Files);

    for i = 1:nTrain
        img = readimage(imdsTrain, i);
        img = imresize(img, imgSize);
        if size(img, 3) == 3
            img = rgb2gray(img);
        end
        feat = extractHOGFeatures(im2single(img), 'CellSize',
hogCellSize);
        if i == 1
            X = zeros(nTrain, length(feat), 'single');
        end
        X(i,:) = feat;
    end

    [Xz, muZ, sigZ] = zscore(double(X));
    sigZ(sigZ == 0) = 1;
    Xz = single(Xz);
    Xtrain = Xz;

    mdlKNN = fitcknn(Xz, Y, 'NumNeighbors', K);
    modelNB = struct();

    for k = 1:nClass
        idx = (Y == kelas{k});
        Xk = Xz(idx, :);
        modelNB.prior(k) = size(Xk, 1) / size(Xz, 1);
        modelNB.mu{k} = mean(double(Xk), 1);
        modelNB.sigma{k} = var(double(Xk), 1) + eps;
    end
    modelNB.kelas = kelas;
    clear X Xz;

    %% ===== 2. PEMROSESAN DATA UJI =====
    imdsTest = imageDatastore(testPath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');
    testFiles = imdsTest.Files;
    nTest = numel(imdsTest.Files);

    hasilGUI = cell(nTest, 4);
    praProsesData = cell(nTest, 6);
    knnIdx5 = zeros(nTest, K);
    knnDist5 = zeros(nTest, K);
    knnVoteText = cell(nTest, 1);
    knnFile5Text = cell(nTest, 1);
    knnLbl5Text = cell(nTest, 1);
    nbLogPostMat = zeros(nTest, nClass);
    nbPostMat = zeros(nTest, nClass);
```

```
    nbLogPriorMat = zeros(nTest, nClass);
    nbConstMat = zeros(nTest, nClass);
    nbQuadMat = zeros(nTest, nClass);
    benarKNN = 0;
    benarNB = 0;
    labelAsliAll = cell(nTest, 1);
    predKNNAll = cell(nTest, 1);
    predNBAll = cell(nTest, 1);

    % Inisialisasi wadah penyimpan HOG asli
    hogFeaturesAll = zeros(nTest, size(Xtrain, 2), 'single');

    for i = 1:nTest
        img = readimage(imdsTest, i);
        img = imresize(img, imgSize);
        if size(img, 3) == 3
            imgGray = rgb2gray(img);
            nChannel = 3;
        else
            imgGray = img;
            nChannel = 1;
        end

        lbl = lower(char(imdsTest.Labels(i)));
        grayMean = mean(imgGray(:));

        % Ekstraksi HOG
        feat = extractHOGFeatures(im2single(imgGray), 'CellSize',
hogCellSize);

        % SIMPAN NILAI HOG ASLI SEBELUM Z-SCORE
        hogFeaturesAll(i, :) = feat;

        % Normalisasi Z-Score untuk pengujian (Syarat KNN)
        feat = (double(feat) - muZ) ./ sigZ;
        feat = single(feat);

        % Klasifikasi KNN
        pk = char(predict(mdlKNN, feat));
        diff = double(Xtrain) - double(feat);
        dAll = sqrt(sum(diff.^2, 2));
        [ds, idxK] = mink(dAll, K);

        knnIdx5(i,:) = idxK(:)';
        knnDist5(i,:) = ds(:)';
        lbl5 = string(trainLabels(idxK));
        file5 = strings(K, 1);

        for t = 1:K
            [~, nm, ex] = fileparts(trainFiles{idxK(t)});
            file5(t) = string(nm) + string(ex);
        end

        ct = countcats(categorical(lbl5, kelas));
```

```
        voteParts = strings(nClass, 1);
        for k = 1:nClass
            voteParts(k) = string(kelas{k}) + "=" + string(ct(k));
        end
        voteText = strjoin(voteParts, " , ");

        knnFile5Text{i} = strjoin(file5, " | ");
        knnLbl5Text{i} = strjoin(lbl5, " | ");
        knnVoteText{i} = voteText;

        % Klasifikasi Naive Bayes
        [pn, logPost, postProb, detNB] = predictNB_detail(modelNB,
double(feat));

        nbLogPostMat(i,:) = logPost(:)';
        nbPostMat(i,:) = postProb(:)';
        nbLogPriorMat(i,:) = detNB.logPrior(:)';
        nbConstMat(i,:) = detNB.constTerm(:)';
        nbQuadMat(i,:) = detNB.quadTerm(:)';
        pn = char(pn);

        [~, nm, ex] = fileparts(imdsTest.Files{i});
        lblTabel = [upper(lbl(1)) lbl(2:end)];
        pkTabel = [upper(pk(1)) pk(2:end)];
        pnTabel = [upper(pn(1)) pn(2:end)];

        hasilGUI(i,:) = {[nm ex], lblTabel, pkTabel, pnTabel};
        praProsesData(i,:) = {[nm ex], lblTabel, grayMean,
[num2str(imgSize(1)) 'x' num2str(imgSize(2))], length(feat), nChannel};

        labelAsliAll{i} = lbl;
        predKNNAll{i} = pk;
        predNBAll{i} = pn;
    end

    benarKNN = sum(strcmp(predKNNAll, labelAsliAll));
    benarNB = sum(strcmp(predNBAll, labelAsliAll));

    akurasiKNN = (benarKNN / nTest) * 100;
    akurasiNB = (benarNB / nTest) * 100;
    salahKNN = nTest - benarKNN;
    salahNB = nTest - benarNB;

    % MASUKKAN HASIL KE TABEL TAMPILAN
    tblHasil.Data = hasilGUI;

    % AKTIFKAN SENSOR KLIK UNTUK POP-UP HISTOGRAM HOG
    tblHasil.CellSelectionCallback = @tampilkanPreviewHOG;

    CM_KNN = confusionmat(categorical(labelAsliAll, kelas),
categorical(predKNNAll, kelas));
    CM_NB = confusionmat(categorical(labelAsliAll, kelas),
categorical(predNBAll, kelas));
```

```
    % UPDATE TAMPILAN PERSENTASE DI PANEL KIRI
    lblAkurasiKNN.Text = sprintf(' Akurasi K-NN (K=5): %.2f%%',
akurasiKNN);
    lblDetailKNN.Text = sprintf('Benar: %d | Salah: %d', benarKNN,
salahKNN);
    lblAkurasiNB.Text = sprintf(' Akurasi Naive Bayes: %.2f%%',
akurasiNB);
    lblDetailNB.Text = sprintf('Benar: %d | Salah: %d', benarNB,
salahNB);
    lblSelisih.Text = sprintf(' Selisih Akurasi: %.2f%%', abs(akurasiKNN
- akurasiNB));
    lblInfoHOG.Text = sprintf(' Panjang Fitur HOG: %d', length(feat));

    if akurasiKNN > akurasiNB
        lblMetodeTerbaik.Text = 'Model Terbaik: K-NN';
        lblMetodeTerbaik.FontColor = [0.1 0.5 0.2];
    elseif akurasiNB > akurasiKNN
        lblMetodeTerbaik.Text = 'Model Terbaik: Naive Bayes';
        lblMetodeTerbaik.FontColor = [0.8 0.3 0.1];
    else
        lblMetodeTerbaik.Text = 'Model Terbaik: Seimbang (Sama)';
        lblMetodeTerbaik.FontColor = [0.2 0.2 0.2];
    end

    uialert(fig, sprintf('Klasifikasi Selesai!\nPanjang Fitur HOG:
%d\nK-NN: %.2f%% | Naive Bayes: %.2f%%', length(feat), akurasiKNN,
akurasiNB), 'Klasifikasi Sukses', 'Icon', 'success');
end

%% ================ EXPORT KE EXCEL MULTI-SHEET ================
function exportHasil(~,~)
    if isempty(hasilGUI)
        uialert(fig, 'Belum ada hasil. Jalankan klasifikasi terlebih
dahulu!', 'Peringatan');
        return;
    end
    outDir = uigetdir(pwd, 'Pilih Folder Penyimpanan Excel');
    if outDir == 0, return; end

    hogCellSize = ddHOG.Value;
    lenFeat = praProsesData{1,5};
    excelFile = fullfile(outDir,
sprintf('Hasil_Klasifikasi_HOG_%d_Fitur.xlsx', lenFeat));
    nTest = size(hasilGUI, 1);
    fileCol = string(hasilGUI(:,1));
    lblCol = string(hasilGUI(:,2));
    pkCol = string(hasilGUI(:,3));
    pnCol = string(hasilGUI(:,4));
    statusKNN = strings(nTest, 1);
    statusNB = strings(nTest, 1);

    for i = 1:nTest
        if strcmpi(lblCol(i), pkCol(i)), statusKNN(i) = "BENAR"; else,
statusKNN(i) = "SALAH"; end
```

```
        if strcmpi(lblCol(i), pnCol(i)), statusNB(i) = "BENAR"; else,
statusNB(i) = "SALAH"; end
    end

    % PROSES PENULISAN TEKS KE EXCEL
    T_Utama = table(fileCol, lblCol, pkCol, statusKNN, pnCol, statusNB,
...
        'VariableNames', {'Nama_File', 'Label_Asli', 'Prediksi_KNN',
'Status_KNN', 'Prediksi_NaiveBayes', 'Status_NB'});
    writetable(T_Utama, excelFile, 'Sheet', 'Hasil_Klasifikasi');

    writetable(cell2table(praProsesData, ...
        'VariableNames', {'Nama_File', 'Label_Asli', 'Mean_Grayscale',
'Ukuran_Citra', 'Panjang_HOG', 'Jumlah_Channel'}), ...
        excelFile, 'Sheet', 'Pra_Proses');

    distText = strings(nTest, 1);
    for i = 1:nTest
        distText(i) = strjoin(compose('%.6f', knnDist5(i,:)), " | ");
    end

    Tknn = table(fileCol, lblCol, pkCol, statusKNN,
string(knnFile5Text), string(knnLbl5Text), distText,
string(knnVoteText), ...
        'VariableNames', {'Nama_File', 'Label_Asli', 'Prediksi_KNN',
'Status_KNN', 'Tetangga_File', 'Tetangga_Label',
'Tetangga_Jarak_Euclidean', 'Voting_Hasil'});
    writetable(Tknn, excelFile, 'Sheet', 'Detail_KNN');

    Tnb = table(fileCol, lblCol, pnCol, statusNB, ...
        'VariableNames', {'Nama_File', 'Label_Asli', 'Prediksi_NB',
'Status_NB'});
    for k = 1:nClass
        nm = matlab.lang.makeValidName(char(kelas{k}));
        Tnb.(strcat("LogPosterior_", nm)) = nbLogPostMat(:,k);
        Tnb.(strcat("Posterior_Persen_", nm)) = nbPostMat(:,k) * 100;
        Tnb.(strcat("LogPrior_", nm)) = nbLogPriorMat(:,k);
        Tnb.(strcat("ConstTerm_", nm)) = nbConstMat(:,k);
        Tnb.(strcat("QuadTerm_", nm)) = nbQuadMat(:,k);
    end
    writetable(Tnb, excelFile, 'Sheet', 'Detail_NB');

    T_ImgRef = table(fileCol, lblCol, pkCol, pnCol, ...
        'VariableNames', {'Nama_File', 'Label_Asli', 'Prediksi_KNN',
'Prediksi_NB'});
    writetable(T_ImgRef, excelFile, 'Sheet', 'Visualisasi_HOG');

    % ======= KODE BARU: MENGEKSPOR NILAI HOG ASLI ======
    T_HOG_Asli = array2table(hogFeaturesAll);
    T_HOG_Asli.Properties.VariableNames = strcat('Fitur_',
string(1:size(hogFeaturesAll,2)));
    T_HOG_Lengkap = [table(fileCol, 'VariableNames', {'Nama_File'}),
T_HOG_Asli];
    writetable(T_HOG_Lengkap, excelFile, 'Sheet', 'Nilai_HOG_Asli');
```

```
    % ====================================================

    pause(2);

    % PROSES PEMBUATAN GAMBAR HOG
    folderGambar = fullfile(outDir, sprintf('Gambar_HOG_%d_Fitur',
lenFeat));
    if ~exist(folderGambar, 'dir')
        mkdir(folderGambar);
    end

    imgPathsSaved = cell(nTest, 1);
    for i = 1:nTest
        img = imread(testFiles{i});
        if size(img, 3) == 3, imgGray = rgb2gray(img); else, imgGray =
img; end
        imgResized = imresize(imgGray, imgSize);
        [~, hogVis] = extractHOGFeatures(im2single(imgResized),
'CellSize', hogCellSize);

        fH = figure('Visible', 'off');
        imshow(imgResized); hold on; plot(hogVis); hold off;
        cleanName = strrep(fileCol(i), '.', '_');
        savePath = fullfile(folderGambar, sprintf('HOG_%s.png',
cleanName));
        saveas(fH, savePath);
        close(fH);
        imgPathsSaved{i} = savePath;
    end

    % PROSES PENYISIPAN GAMBAR ACTIVEX KE EXCEL
    try
        Excel = actxserver('Excel.Application');
        Excel.Visible = false;
        Excel.DisplayAlerts = false;
        Workbook = Excel.Workbooks.Open(excelFile);
        SheetHOG = Workbook.Sheets.Item('Visualisasi_HOG');
        SheetHOG.Range('E1').Value = 'Visualisasi_HOG';
        SheetHOG.Columns.Item(5).ColumnWidth = 25;

        for i = 1:nTest
            row = i + 1;
            SheetHOG.Rows.Item(row).RowHeight = 70;
            cellRange = SheetHOG.Range(sprintf('E%d', row));
            left = cellRange.Left + 2;
            top = cellRange.Top + 2;
            pathGambar = char(imgPathsSaved{i});
            SheetHOG.Shapes.AddPicture(pathGambar, 0, 1, left, top, 110,
65);
        end

        Workbook.Save();
        Workbook.Close();
        Excel.Quit();
```

```
        Excel.delete();
        uialert(fig, sprintf('SUKSES! Laporan Excel & Gambar HOG (%d
Fitur) berhasil disisipkan di:\n%s', lenFeat, excelFile), 'Export
Berhasil', 'Icon', 'success');
    catch ME
        if exist('Excel', 'var')
            try
                Workbook.Close(false);
                Excel.Quit();
                Excel.delete();
            catch
            end
        end
        uialert(fig, sprintf('Teks berhasil disimpan di Excel dan gambar
tersimpan di folder "Gambar_HOG_%d_Fitur".\n\nPesan Error Excel: %s',
lenFeat, ME.message), 'Peringatan Excel', 'Icon', 'warning');
    end
end

%% =============== FUNGSI PERHITUNGAN NAIVE BAYES DETAIL ===============
function [label, logPosterior, posterior, detail] =
predictNB_detail(model, x)
    nClass = numel(model.kelas);
    logPosterior = zeros(nClass, 1);
    detail = struct();
    detail.logPrior = zeros(nClass, 1);
    detail.constTerm = zeros(nClass, 1);
    detail.quadTerm = zeros(nClass, 1);

    for k = 1:nClass
        mu = model.mu{k};
        sg = model.sigma{k};
        constTerm = -0.5 * sum(log(2 * pi * sg));
        quadTerm = -0.5 * sum(((x - mu).^2) ./ sg);
        logLikelihood = constTerm + quadTerm;
        logPrior = log(model.prior(k));
        logPosterior(k) = logPrior + logLikelihood;

        detail.logPrior(k) = logPrior;
        detail.constTerm(k) = constTerm;
        detail.quadTerm(k) = quadTerm;
    end

    tmp = exp(logPosterior - max(logPosterior));
    posterior = tmp / sum(tmp);
    [~, idx] = max(logPosterior);
    label = model.kelas(idx);
end

%% =============== FUNGSI UNTUK MENAMPILKAN PREVIEW & DIAGRAM HOG
===============
function tampilkanPreviewHOG(src, event)
    % Jika tabel diklik tapi meleset/kosong, abaikan
    if isempty(event.Indices)
```

```
        return;
    end

    % Ambil nomor baris yang diklik
    row = event.Indices(1, 1);

    % Pastikan baris yang diklik valid
    if row > length(testFiles)
        return;
    end

    % Ambil path gambar asli berdasarkan baris yang diklik
    imgPath = testFiles{row};
    img = imread(imgPath);

    % Pra-pemrosesan (Resize & Grayscale) persis seperti sistem utama
    imgResized = imresize(img, imgSize);
    if size(imgResized, 3) == 3
        imgGray = rgb2gray(imgResized);
    else
        imgGray = imgResized;
    end

    % Ambil ukuran Cell HOG yang sedang dipilih di dropdown GUI
    currentCellSize = ddHOG.Value;

    % EKSTRAKSI HOG: Ambil FITUR (Angka untuk diagram) & VISUALISASI
(Untuk gambar)
    [feat, hogVis] = extractHOGFeatures(im2single(imgGray), 'CellSize',
currentCellSize);

    % Ambil nama file dari tabel untuk dijadikan judul
    namaFile = src.Data{row, 1};

    % Buka jendela Pop-up Figure baru (Ukurannya dilebarkan jadi 900
untuk 2 gambar)
    fPreview = figure('Name', ['Analisis HOG Lengkap - ' namaFile], ...
                      'NumberTitle', 'off', ...
                      'Position', [250 200 950 450], ... % Dibuat
memanjang ke samping
                      'Color', 'white');

    % ================= BAGIAN KIRI: GAMBAR + JARING HOG
=================
    subplot(1, 2, 1);
    imshow(imgResized);
    hold on;
    plot(hogVis); % Menimpa jaring HOG
    title(sprintf('Visualisasi Spatial HOG\nFile: %s', namaFile),
'Interpreter', 'none', 'FontSize', 11);
    hold off;

    % ================= BAGIAN KANAN: DIAGRAM BATANG (HISTOGRAM)
=================
```

```
    subplot(1, 2, 2);
    % Membuat diagram batang dari array fitur. EdgeColor 'none' agar
tidak terlalu hitam
    bar(feat, 'FaceColor', [0.1 0.5 0.8], 'EdgeColor', 'none');

    % Mengatur tampilan grafik
    xlim([0 length(feat)]); % Rentang sumbu X (misal 0 sampai 1764)
    grid on; % Nyalakan garis bantu

    % Memberi label pada sumbu
    xlabel(sprintf('Indeks Fitur (Dimensi ke-1 sampai %d)',
length(feat)), 'FontWeight', 'bold');
    ylabel('Nilai / Magnitudo', 'FontWeight', 'bold');
    title(sprintf('Diagram Histogram Fitur HOG\nPanjang Fitur: %d',
length(feat)), 'FontSize', 11);
end

end


    %% ===== RENAME DATA LATIH =====
    kelas = dir(latihPath);
    kelas = kelas([kelas.isdir] & ~startsWith({kelas.name},'.'));

    for k = 1:length(kelas)
        className = kelas(k).name;
        classPath = fullfile(latihPath,className);

        files = [ ...
            dir(fullfile(classPath,'*.jpg')); ...
            dir(fullfile(classPath,'*.jpeg')); ...
            dir(fullfile(classPath,'*.png')) ];

        for i = 1:length(files)
            oldName = files(i).name;
            [~,~,ext] = fileparts(oldName);

            newName = sprintf('%s_%03d%s',className,i,ext);

            movefile(fullfile(classPath,oldName),...
                     fullfile(classPath,newName));
        end
    end

    %% ===== RENAME DATA UJI =====
    filesUji = [ ...
        dir(fullfile(ujiPath,'*.jpg')); ...
        dir(fullfile(ujiPath,'*.jpeg')); ...
        dir(fullfile(ujiPath,'*.png')) ];

    for i = 1:length(filesUji)
        oldName = filesUji(i).name;
        [~,~,ext] = fileparts(oldName);

        newName = sprintf('uji_%03d%s',i,ext);
```

```
        movefile(fullfile(ujiPath,oldName),...
                 fullfile(ujiPath,newName));
    end

    uialert(fig,'Rename data latih & uji berhasil','SUKSES');
end
end
```