

LAMPIRAN

1. SOURCE CODE

1.1 Binerisasi Aksara Jawa

```
% ===== AKSARA IMAGE BINARIZATION =====
% Batch conversion of Aksara Jawa images to binary
% Run directly in MATLAB Command Window

% Initialize
clear; clc; close all;
disp('Starting Aksara Jawa image processing...');

% Image list (all 160 Aksara Jawa characters)
images = {
    'ha.jpg', 'ha (1).jpg', 'ha (2).jpg', 'ha (3).jpg', 'ha
(4).jpg', 'ha (5).jpg', 'ha (6).jpg', 'ha (7).jpg', ...
    'na.jpg', 'na (1).jpg', 'na (2).jpg', 'na (3).jpg', 'na
(4).jpg', 'na (5).jpg', 'na (6).jpg', 'na (7).jpg', ...
    'ca.jpg', 'ca (1).jpg', 'ca (2).jpg', 'ca (3).jpg', 'ca
(4).jpg', 'ca (5).jpg', 'ca (6).jpg', 'ca (7).jpg', ...
    'ra.jpg', 'ra (1).jpg', 'ra (2).jpg', 'ra (3).jpg', 'ra
(4).jpg', 'ra (5).jpg', 'ra (6).jpg', 'ra (7).jpg', ...
    'ka.jpg', 'ka (1).jpg', 'ka (2).jpg', 'ka (3).jpg', 'ka
(4).jpg', 'ka (5).jpg', 'ka (6).jpg', 'ka (7).jpg', ...
    'da.jpg', 'da (1).jpg', 'da (2).jpg', 'da (3).jpg', 'da
(4).jpg', 'da (5).jpg', 'da (6).jpg', 'da (7).jpg', ...
    'ta.jpg', 'ta (1).jpg', 'ta (2).jpg', 'ta (3).jpg', 'ta
(4).jpg', 'ta (5).jpg', 'ta (6).jpg', 'ta (7).jpg', ...
    'sa.jpg', 'sa (1).jpg', 'sa (2).jpg', 'sa (3).jpg', 'sa
(4).jpg', 'sa (5).jpg', 'sa (6).jpg', 'sa (7).jpg', ...
    'wa.jpg', 'wa (1).jpg', 'wa (2).jpg', 'wa (3).jpg', 'wa
(4).jpg', 'wa (5).jpg', 'wa (6).jpg', 'wa (7).jpg', ...
    'la.jpg', 'la (1).jpg', 'la (2).jpg', 'la (3).jpg', 'la
(4).jpg', 'la (5).jpg', 'la (6).jpg', 'la (7).jpg', ...
    'pa.jpg', 'pa (1).jpg', 'pa (2).jpg', 'pa (3).jpg', 'pa
(4).jpg', 'pa (5).jpg', 'pa (6).jpg', 'pa (7).jpg', ...
    'dha.jpg', 'dha (1).jpg', 'dha (2).jpg', 'dha (3).jpg', 'dha
(4).jpg', 'dha (5).jpg', 'dha (6).jpg', 'dha (7).jpg', ...
    'ja.jpg', 'ja (1).jpg', 'ja (2).jpg', 'ja (3).jpg', 'ja
(4).jpg', 'ja (5).jpg', 'ja (6).jpg', 'ja (7).jpg', ...
    'ya.jpg', 'ya (1).jpg', 'ya (2).jpg', 'ya (3).jpg', 'ya
(4).jpg', 'ya (5).jpg', 'ya (6).jpg', 'ya (7).jpg', ...
    'nya.jpg', 'nya (1).jpg', 'nya (2).jpg', 'nya (3).jpg', 'nya
(4).jpg', 'nya (5).jpg', 'nya (6).jpg', 'nya (7).jpg', ...
    'ma.jpg', 'ma (1).jpg', 'ma (2).jpg', 'ma (3).jpg', 'ma
(4).jpg', 'ma (5).jpg', 'ma (6).jpg', 'ma (7).jpg', ...
    'ga.jpg', 'ga (1).jpg', 'ga (2).jpg', 'ga (3).jpg', 'ga
(4).jpg', 'ga (5).jpg', 'ga (6).jpg', 'ga (7).jpg', ...
    'ba.jpg', 'ba (1).jpg', 'ba (2).jpg', 'ba (3).jpg', 'ba
(4).jpg', 'ba (5).jpg', 'ba (6).jpg', 'ba (7).jpg', ...
```

```

    'tha.jpg','tha (1).jpg','tha (2).jpg','tha (3).jpg','tha
(4).jpg','tha (5).jpg','tha (6).jpg','tha (7).jpg',...
    'nga.jpg','nga (1).jpg','nga (2).jpg','nga (3).jpg','nga
(4).jpg','nga (5).jpg','nga (6).jpg','nga (7).jpg'
};

% Create output directory
if ~exist('binary_results','dir')
    mkdir('binary_results');
end

% Process each image
success_count = 0;
total_images = length(images);

for i = 1:total_images
    try
        % 1. Read and convert to grayscale
        img = imread(images{i});
        if size(img,3) == 3
            gray = rgb2gray(img);
        else
            gray = img;
        end

        % 2. Otsu thresholding
        level = graythresh(gray);
        binary = imbinarize(gray,level);

        % 3. Save binary image
        [~,name,ext] = fileparts(images{i});
        imwrite(binary,['binary_results/' name ' ' ext]);

        % 4. Display progress
        fprintf('%03d/%d: %s (Threshold=%.2f)\n',...
            i,total_images,images{i},level*255);
        success_count = success_count + 1;

    catch ME
        warning('Error processing %s: %s',images{i},ME.message);
    end
end

% Summary
fprintf('\n===== PROCESSING COMPLETE =====\n');
fprintf('Successfully processed: %d/%d
images\n',success_count,total_images);
fprintf('Binary images saved in:
%s\n',fullfile(pwd,'binary_results'));

```

1.2 Deteksi Tepi Prewitt

```
% ===== DETEKSI TEPI PREWITT UNTUK 160 AKSARA JAWA =====
% Oleh: [Nama Anda]
% Skripsi: [Judul Skripsi]

clear; clc; close all;

% Daftar semua file gambar (pastikan nama file sesuai)
files = {
    'ha.jpg','ha (1).jpg','ha (2).jpg','ha (3).jpg','ha
(4).jpg','ha (5).jpg','ha (6).jpg','ha (7).jpg',...
    'na.jpg','na (1).jpg','na (2).jpg','na (3).jpg','na
(4).jpg','na (5).jpg','na (6).jpg','na (7).jpg',...
    'ca.jpg','ca (1).jpg','ca (2).jpg','ca (3).jpg','ca
(4).jpg','ca (5).jpg','ca (6).jpg','ca (7).jpg',...
    'ra.jpg','ra (1).jpg','ra (2).jpg','ra (3).jpg','ra
(4).jpg','ra (5).jpg','ra (6).jpg','ra (7).jpg',...
    'ka.jpg','ka (1).jpg','ka (2).jpg','ka (3).jpg','ka
(4).jpg','ka (5).jpg','ka (6).jpg','ka (7).jpg',...
    'da.jpg','da (1).jpg','da (2).jpg','da (3).jpg','da
(4).jpg','da (5).jpg','da (6).jpg','da (7).jpg',...
    'ta.jpg','ta (1).jpg','ta (2).jpg','ta (3).jpg','ta
(4).jpg','ta (5).jpg','ta (6).jpg','ta (7).jpg',...
    'sa.jpg','sa (1).jpg','sa (2).jpg','sa (3).jpg','sa
(4).jpg','sa (5).jpg','sa (6).jpg','sa (7).jpg',...
    'wa.jpg','wa (1).jpg','wa (2).jpg','wa (3).jpg','wa
(4).jpg','wa (5).jpg','wa (6).jpg','wa (7).jpg',...
    'la.jpg','la (1).jpg','la (2).jpg','la (3).jpg','la
(4).jpg','la (5).jpg','la (6).jpg','la (7).jpg',...
    'pa.jpg','pa (1).jpg','pa (2).jpg','pa (3).jpg','pa
(4).jpg','pa (5).jpg','pa (6).jpg','pa (7).jpg',...
    'dha.jpg','dha (1).jpg','dha (2).jpg','dha (3).jpg','dha
(4).jpg','dha (5).jpg','dha (6).jpg','dha (7).jpg',...
    'ja.jpg','ja (1).jpg','ja (2).jpg','ja (3).jpg','ja
(4).jpg','ja (5).jpg','ja (6).jpg','ja (7).jpg',...
    'ya.jpg','ya (1).jpg','ya (2).jpg','ya (3).jpg','ya
(4).jpg','ya (5).jpg','ya (6).jpg','ya (7).jpg',...
    'nya.jpg','nya (1).jpg','nya (2).jpg','nya (3).jpg','nya
(4).jpg','nya (5).jpg','nya (6).jpg','nya (7).jpg',...
    'ma.jpg','ma (1).jpg','ma (2).jpg','ma (3).jpg','ma
(4).jpg','ma (5).jpg','ma (6).jpg','ma (7).jpg',...
    'ga.jpg','ga (1).jpg','ga (2).jpg','ga (3).jpg','ga
(4).jpg','ga (5).jpg','ga (6).jpg','ga (7).jpg',...
    'ba.jpg','ba (1).jpg','ba (2).jpg','ba (3).jpg','ba
(4).jpg','ba (5).jpg','ba (6).jpg','ba (7).jpg',...
    'tha.jpg','tha (1).jpg','tha (2).jpg','tha (3).jpg','tha
(4).jpg','tha (5).jpg','tha (6).jpg','tha (7).jpg',...
    'nga.jpg','nga (1).jpg','nga (2).jpg','nga (3).jpg','nga
(4).jpg','nga (5).jpg','nga (6).jpg','nga (7).jpg'
};

% Buat folder penyimpanan
```

```

if ~exist('hasil_deteksi_tepi', 'dir')
    mkdir('hasil_deteksi_tepi');
end

% Kernel Prewitt
prewitt_x = [-1 0 1; -1 0 1; -1 0 1]; % Deteksi tepi vertikal
prewitt_y = [-1 -1 -1; 0 0 0; 1 1 1]; % Deteksi tepi horizontal

% Proses semua gambar
for i = 1:length(files)
    try
        % 1. Baca gambar biner
        bw_img = imread(files{i});

        % 2. Pastikan gambar biner (hitam-putih)
        if size(bw_img, 3) == 3
            bw_img = rgb2gray(bw_img) > 0.5;
        else
            bw_img = bw_img > 0.5;
        end

        % 3. Hitung gradien Prewitt
        Gx = imfilter(double(bw_img), prewitt_x, 'same');
        Gy = imfilter(double(bw_img), prewitt_y, 'same');
        edge_mag = sqrt(Gx.^2 + Gy.^2);

        % 4. Normalisasi dan konversi ke uint8
        edge_norm = uint8(255 * mat2gray(edge_mag));

        % 5. Simpan hasil
        [~, name, ext] = fileparts(files{i});
        imwrite(edge_norm, fullfile('hasil_deteksi_tepi', ['',
name, ext]]));

        % 6. Tampilkan progress
        fprintf('%03d/%d: %s diproses (Intensitas tepi rata-rata:
%.2f)\n', ...
            i, length(files), files{i}, mean(edge_mag(:)));

    catch ME
        warning('Gagal memproses %s: %s', files{i}, ME.message);
    end
end

disp('=== PROSES SELESAI ===');
disp(['Hasil disimpan di folder: ' fullfile(pwd,
'hasil_deteksi_tepi')]);

```

1.3 Resize 30x30

```
% Daftar lengkap file dataset
fileList = {
    'ha.jpg', 'ha (1).jpg', 'ha (2).jpg', 'ha (3).jpg', 'ha
(4).jpg', 'ha (5).jpg', 'ha (6).jpg', 'ha (7).jpg', ...
    'na.jpg', 'na (1).jpg', 'na (2).jpg', 'na (3).jpg', 'na
(4).jpg', 'na (5).jpg', 'na (6).jpg', 'na (7).jpg', ...
    'ca.jpg', 'ca (1).jpg', 'ca (2).jpg', 'ca (3).jpg', 'ca
(4).jpg', 'ca (5).jpg', 'ca (6).jpg', 'ca (7).jpg', ...
    'ra.jpg', 'ra (1).jpg', 'ra (2).jpg', 'ra (3).jpg', 'ra
(4).jpg', 'ra (5).jpg', 'ra (6).jpg', 'ra (7).jpg', ...
    'ka.jpg', 'ka (1).jpg', 'ka (2).jpg', 'ka (3).jpg', 'ka
(4).jpg', 'ka (5).jpg', 'ka (6).jpg', 'ka (7).jpg', ...
    'da.jpg', 'da (1).jpg', 'da (2).jpg', 'da (3).jpg', 'da
(4).jpg', 'da (5).jpg', 'da (6).jpg', 'da (7).jpg', ...
    'ta.jpg', 'ta (1).jpg', 'ta (2).jpg', 'ta (3).jpg', 'ta
(4).jpg', 'ta (5).jpg', 'ta (6).jpg', 'ta (7).jpg', ...
    'sa.jpg', 'sa (1).jpg', 'sa (2).jpg', 'sa (3).jpg', 'sa
(4).jpg', 'sa (5).jpg', 'sa (6).jpg', 'sa (7).jpg', ...
    'wa.jpg', 'wa (1).jpg', 'wa (2).jpg', 'wa (3).jpg', 'wa
(4).jpg', 'wa (5).jpg', 'wa (6).jpg', 'wa (7).jpg', ...
    'la.jpg', 'la (1).jpg', 'la (2).jpg', 'la (3).jpg', 'la
(4).jpg', 'la (5).jpg', 'la (6).jpg', 'la (7).jpg', ...
    'pa.jpg', 'pa (1).jpg', 'pa (2).jpg', 'pa (3).jpg', 'pa
(4).jpg', 'pa (5).jpg', 'pa (6).jpg', 'pa (7).jpg', ...
    'dha.jpg', 'dha (1).jpg', 'dha (2).jpg', 'dha (3).jpg', 'dha
(4).jpg', 'dha (5).jpg', 'dha (6).jpg', 'dha (7).jpg', ...
    'ja.jpg', 'ja (1).jpg', 'ja (2).jpg', 'ja (3).jpg', 'ja
(4).jpg', 'ja (5).jpg', 'ja (6).jpg', 'ja (7).jpg', ...
    'ya.jpg', 'ya (1).jpg', 'ya (2).jpg', 'ya (3).jpg', 'ya
(4).jpg', 'ya (5).jpg', 'ya (6).jpg', 'ya (7).jpg', ...
    'nya.jpg', 'nya (1).jpg', 'nya (2).jpg', 'nya (3).jpg', 'nya
(4).jpg', 'nya (5).jpg', 'nya (6).jpg', 'nya (7).jpg', ...
    'ma.jpg', 'ma (1).jpg', 'ma (2).jpg', 'ma (3).jpg', 'ma
(4).jpg', 'ma (5).jpg', 'ma (6).jpg', 'ma (7).jpg', ...
    'ga.jpg', 'ga (1).jpg', 'ga (2).jpg', 'ga (3).jpg', 'ga
(4).jpg', 'ga (5).jpg', 'ga (6).jpg', 'ga (7).jpg', ...
    'ba.jpg', 'ba (1).jpg', 'ba (2).jpg', 'ba (3).jpg', 'ba
(4).jpg', 'ba (5).jpg', 'ba (6).jpg', 'ba (7).jpg', ...
    'tha.jpg', 'tha (1).jpg', 'tha (2).jpg', 'tha (3).jpg', 'tha
(4).jpg', 'tha (5).jpg', 'tha (6).jpg', 'tha (7).jpg', ...
    'nga.jpg', 'nga (1).jpg', 'nga (2).jpg', 'nga (3).jpg', 'nga
(4).jpg', 'nga (5).jpg', 'nga (6).jpg', 'nga (7).jpg'
};

% Buat folder output jika belum ada
outputDir = 'resized_30x30';
if ~exist(outputDir, 'dir')
    mkdir(outputDir);
end

% Proses resize untuk semua file
```

```

for i = 1:length(fileList)
    try
        % Baca gambar
        imgPath = fileList{i};
        img = imread(imgPath);

        % Konversi ke grayscale jika perlu
        if size(img, 3) == 3
            img = rgb2gray(img);
        end

        % Resize ke 30x30 dengan metode bicubic
        imgResized = imresize(img, [30 30], 'bicubic');

        % Simpan gambar hasil resize
        [~, name, ext] = fileparts(imgPath);
        outputPath = fullfile(outputDir, [name ext]);
        imwrite(imgResized, outputPath);

        % Tampilkan konfirmasi
        fprintf('Berhasil resize: %s -> %s\n', imgPath,
            outputPath);
    catch ME
        warning('Gagal memproses %s: %s', imgPath, ME.message);
    end
end

% Verifikasi hasil
resizedFiles = dir(fullfile(outputDir, '*.jpg'));
fprintf('\n=== HASIL RESIZE ===\n');
fprintf('Total file diproses: %d\n', length(fileList));
fprintf('Total file berhasil: %d\n', length(resizedFiles));
fprintf('Folder output: %s\n', fullfile(pwd, outputDir));

% Contoh cek ukuran 5 file pertama
fprintf('\nContoh ukuran file hasil:\n');
for i = 1:min(5, length(resizedFiles))
    imgInfo = imfinfo(fullfile(outputDir, resizedFiles(i).name));
    fprintf('%s: %dx%d piksel\n', resizedFiles(i).name,
        imgInfo.Height, imgInfo.Width);
end

```

1.4 Arsitektur JST Perceptron

```

%% 1. Membersihkan workspace dan menutup semua gambar

clear all; close all; clc; % Membersihkan environment MATLAB

%% 2. Memuat dan mempersiapkan dataset
% Daftar aksara Jawa
aksara = {'ha', 'na', 'ca', 'ra', 'ka', 'da', 'ta', 'sa', 'wa',
'la', ...
          'pa', 'dha', 'ja', 'ya', 'nya', 'ma', 'ga', 'ba', 'tha',
'nga'};
numClasses = length(aksara); % 20 kelas aksara
samplesPerClass = 8; % Sekarang ada 8 sampel per kelas

% Inisialisasi variabel untuk data dan label
data = [];
labels = [];

% Loop melalui setiap aksara dan setiap sampel
for i = 1:numClasses
    for j = 0:samplesPerClass-1 % Mulai dari 0 karena format
filename
        % Membuat nama file
        if j == 0
            filename = [aksara{i} '.jpg'];
        else
            filename = [aksara{i} ' (' num2str(j) ').jpg'];
        end

        % Membaca dan memproses gambar
        img = imread(filename);
        if size(img, 3) == 3
            img = rgb2gray(img); % Konversi ke grayscale jika perlu
        end

        % Normalisasi dan flatten gambar ke vektor
        imgVector = double(img(:)) / 255;

        % Menyimpan data dan label
        data = [data imgVector];
        labels = [labels; i];
    end
end

% Transpose data untuk format yang benar (samples x features)
data = data';

%% 3. Pembagian dataset (70% training, 30% testing)
rng(42); % Seed untuk reproduktibilitas
indices = randperm(size(data, 1));
trainSize = round(0.7 * size(data, 1));

```

```

trainData = data(indices(1:trainSize), :);
trainLabels = labels(indices(1:trainSize));
testData = data(indices(trainSize+1:end), :);
testLabels = labels(indices(trainSize+1:end));

%% 4. Membuat arsitektur Perceptron
inputSize = size(trainData, 2); % Jumlah input = jumlah pixel
outputSize = numClasses; % Jumlah output = jumlah kelas

% Membuat perceptron dengan fungsi aktivasi hardlim
net = perceptron;
net = configure(net, trainData', dummyvar(trainLabels)');

% Menampilkan arsitektur
view(net)

%% 5. Pelatihan model dengan parameter yang disesuaikan
% Konversi label ke one-hot encoding
trainTargets = dummyvar(trainLabels)';

% Parameter pelatihan yang diperbarui
net.trainParam.epochs = 5000; % Meningkatkan jumlah epoch
net.trainParam.show = 50; % Menampilkan progress setiap 50 epoch
net.trainParam.lr = 0.001; % Learning rate yang lebih kecil
net.trainParam.goal = 1e-6; % Target error yang lebih ketat

% Melatih jaringan
[net, tr] = train(net, trainData', trainTargets);

%% 6. Evaluasi model - DIUBAH
% Prediksi pada data testing
testOutputs = net(testData');
[~, predictedLabels] = max(testOutputs);

% Pastikan predictedLabels tidak melebihi numClasses
predictedLabels = min(predictedLabels, numClasses); % Batasi maksimal
predictedLabels = max(predictedLabels, 1); % Batasi minimal

%% 7. Analisis performa - DIUBAH
% Confusion matrix
figure;
plotconfusion(dummyvar(testLabels)', testOutputs);
title('Confusion Matrix - Klasifikasi Aksara Jawa');

% Menghitung precision, recall, F1-score
confMat = confusionmat(testLabels, predictedLabels', 'Order',
1:numClasses);

% Handle kasus pembagi 0
precision = diag(confMat)./(sum(confMat,2) + eps);
recall = diag(confMat)./(sum(confMat,1)' + eps);
f1Scores = 2*(precision.*recall)./(precision+recall + eps);

```

```

%% Hitung dan tampilkan metrik untuk SEMUA 20 kelas - DIUBAH
fprintf('\n=== METRIK UNTUK SEMUA 20 AKSARA JAWA ===\n');
fprintf('%-6s | %-10s | %-10s | %-10s\n', 'Aksara', 'Precision',
'Recall', 'F1-Score');
fprintf('-----\n');
for i = 1:numClasses
    % Pastikan tidak ada NaN
    if isnan(f1Scores(i)) || isinf(f1Scores(i))
        f1Scores(i) = 0;
    end

    fprintf('%-6s | %-10.2f | %-10.2f | %-10.2f\n', ...
        aksara{i}, precision(i), recall(i), f1Scores(i));
end

% Hitung akurasi
correctPredictions = sum(predictedLabels' == testLabels);
accuracy = correctPredictions / numel(testLabels);

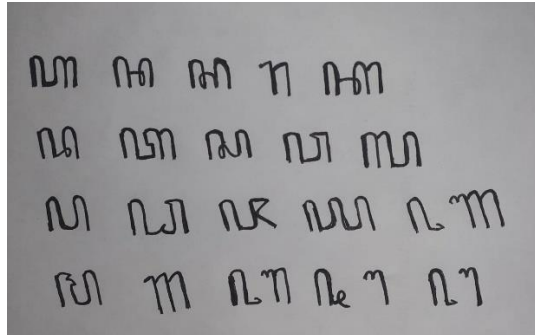
% Tambahkan summary
fprintf('\n=== RINGKASAN ===\n');
fprintf('Akurasi Model: %.2f%%\n', accuracy * 100);
fprintf('Rata-rata F1-Score: %.2f\n', mean(f1Scores));

%% 8. Menyimpan model - DIUBAH
try
    save('perceptron_model_v2.mat', 'net', 'aksara', 'tr', '-
v7.3');
    fprintf('\nModel berhasil disimpan sebagai
"perceptron_model_v2.mat"\n');
catch ME
    fprintf('\nGagal menyimpan model. Error: %s\n', ME.message);
    fprintf('Menyimpan dengan alternatif...\n');

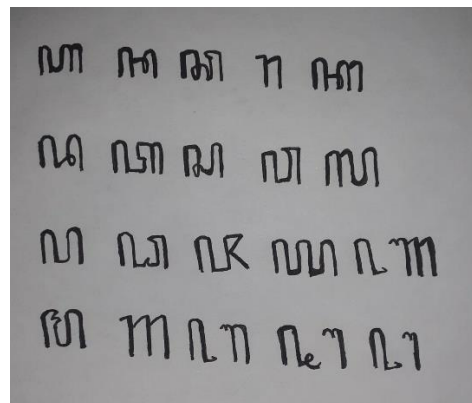
    % Alternatif 1: Simpan tanpa 'tr'
    try
        save('perceptron_model_v2_simple.mat', 'net', 'aksara');
        fprintf('Model alternatif berhasil disimpan tanpa training
record\n');
    catch
        % Alternatif 2: Simpan dengan format berbeda
        save('perceptron_model_v2_backup.mat', 'net', 'aksara', '-
ascii');
        fprintf('Model berhasil disimpan dalam format ASCII\n');
    end
end
end

```

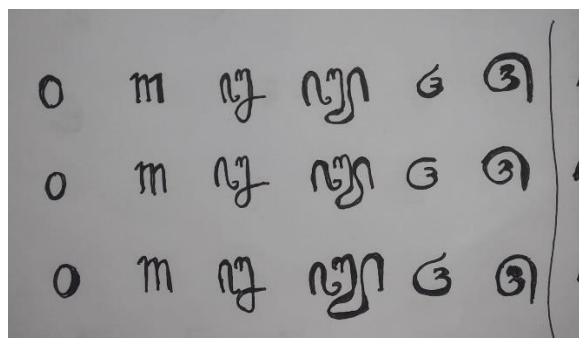
2. HASIL KEGIATAN PENGUMPULAN DATA



Gambar 1. Tulisan tangan aksara jawa



Gambar 2. Tulisan tangan aksara jawa



Gambar 3. Tulisan tangan aksara jawa

RIWAYAT HIDUP



Khoirun Nisa dilahirkan di Blitar, Jawa Timur pada 21 April 2001. Sebagai anak sulung dari dua bersaudara, ia merupakan putri dari pasangan Khoirul Huda dan Nikmatul Imdad. Riwayat pendidikan formalnya dimulai pada tahun 2008 ketika menempuh pendidikan dasar di MI Maftahul Ulum Karangsono Kanigoro Blitar hingga menyelesaikannya pada tahun 2014. Jenjang pendidikan menengah pertama ditempuh di SMP Negeri 1 Kanigoro Blitar dari tahun 2014 hingga 2017. Selanjutnya, penulis melanjutkan pendidikan kejuruan di SMK Negeri 2 Blitar selama tiga tahun (2017-2020). Pada tahun 2020, penulis berhasil lolos seleksi tes masuk Universitas Islam Balitar di Program Studi Teknik Informatika.