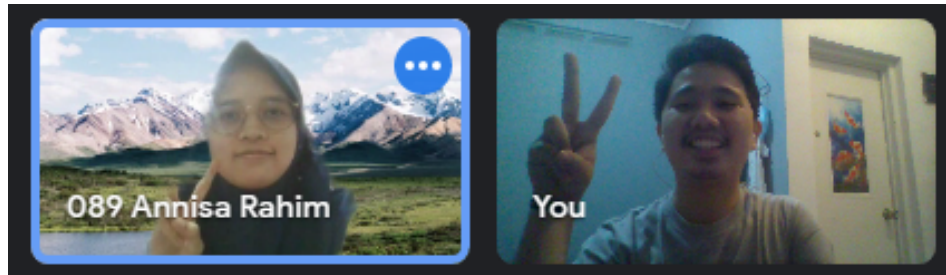


LAPORAN TUGAS 4

IF4073 Interpretasi dan Pengolahan Citra

Implementasi *Automatic Plate Number Recognition* Menggunakan MATLAB



Disusun oleh

| | |
|-------------------------|----------|
| Annisa Rahim | 13518089 |
| Stefanus Gusega Gunawan | 13518149 |

Tanggal Pengumpulan

Senin, 25 April 2022

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022**

Daftar Isi

| | |
|---|-----------|
| Daftar Isi | 1 |
| Kode Program MATLAB | 2 |
| Pembuatan Template | 2 |
| Deteksi Plat (Plate Detection) | 3 |
| Segmentasi Karakter (Character Segmentation) | 6 |
| Pengenalan Karakter (Character Recognition) | 8 |
| Screenshot Antarmuka Program | 10 |
| Contoh Hasil Eksekusi Program | 11 |
| Tidak Secara Langsung Gambar Plat | 11 |
| Secara Langsung Citra Plat | 11 |
| Analisis Hasil Pengenalan | 12 |
| Hasil Pengenalan dan Akurasi untuk Citra Plat Secara Langsung | 12 |
| Hasil Pengenalan dan Akurasi untuk Citra Plat Secara Tidak Langsung | 14 |
| Analisis | 16 |
| Pranala Kode Program pada GitHub | 17 |
| Referensi | 18 |

Kode Program MATLAB

Pada tugas ini, kelompok kami mengimplementasikan aplikasi *Automatic Plate Number Recognition* sederhana dengan menggunakan teknik-teknik *image processing*. Proses pengenalan plat nomor pada implementasi kami secara umum dibagi menjadi tiga tahap, yaitu deteksi plat (*plate detection*), segmentasi karakter (*character segmentation*), dan pengenalan karakter (*character recognition*). Lalu, dalam proses pengenalan karakter dilakukan *template matching*, sehingga sebelum membuat sistemnya, perlu mengekstrak *template* dari karakter-karakter yang mungkin pada plat nomor.

Berikut ini akan dijelaskan implementasi dari masing-masing tahapan pada pengenalan plat nomor yang kami buat, mulai dari pembuatan *template* hingga *character recognition*.

1. Pembuatan *Template*

Pembuatan *template* dilakukan dalam *script* **createTemplates.m**. Berikut rincian implementasinya.

createTemplates.m

```
% This script is used to create a new template

% The template folder
templateDir = '../templates';

% Alphabets
A = imread(fullfile(templateDir, 'A.bmp')); B =
imread(fullfile(templateDir, 'B.bmp')); C = imread(fullfile(templateDir,
'C.bmp'));
D = imread(fullfile(templateDir, 'D.bmp')); E =
imread(fullfile(templateDir, 'E.bmp')); F = imread(fullfile(templateDir,
'F.bmp'));
G = imread(fullfile(templateDir, 'G.bmp')); H =
imread(fullfile(templateDir, 'H.bmp')); I = imread(fullfile(templateDir,
'I.bmp')); I2 = imread(fullfile(templateDir, 'I2.bmp'));
J = imread(fullfile(templateDir, 'J.bmp')); K =
imread(fullfile(templateDir, 'K.bmp')); L = imread(fullfile(templateDir,
'L.bmp'));
M = imread(fullfile(templateDir, 'M.bmp')); N =
imread(fullfile(templateDir, 'N.bmp')); O = imread(fullfile(templateDir,
'O.bmp'));
P = imread(fullfile(templateDir, 'P.bmp')); Q =
imread(fullfile(templateDir, 'Q.bmp')); R = imread(fullfile(templateDir,
'R.bmp'));
S = imread(fullfile(templateDir, 'S.bmp')); T =
```

```

imread(fullfile(templateDir, 'T.bmp')); U = imread(fullfile(templateDir,
'U.bmp'));
V = imread(fullfile(templateDir, 'V.bmp')); W =
imread(fullfile(templateDir, 'W.bmp')); X = imread(fullfile(templateDir,
'X.bmp'));
Y = imread(fullfile(templateDir, 'Y.bmp')); Z =
imread(fullfile(templateDir, 'Z.bmp'));

% Nonnegative Integers
one = imread(fullfile(templateDir, '1.bmp')); one2 =
imread(fullfile(templateDir, '12.bmp')); two =
imread(fullfile(templateDir, '2.bmp'));
three = imread(fullfile(templateDir, '3.bmp')); four =
imread(fullfile(templateDir, '4.bmp'));
five = imread(fullfile(templateDir, '5.bmp')); six =
imread(fullfile(templateDir, '6.bmp'));
seven = imread(fullfile(templateDir, '7.bmp')); eight =
imread(fullfile(templateDir, '8.bmp'));
nine = imread(fullfile(templateDir, '9.bmp')); zero =
imread(fullfile(templateDir, '0.bmp')); zero2 =
imread(fullfile(templateDir, '02.bmp'));

% Creating Array for Alphabets
letter = [A B C D E F G H I J K L M N O P Q R S T U V W X Y Z];

% Creating Array for Numbers
number = [one one2 two three four five six seven eight nine zero zero2];

newTemplates = [letter number];
save ('newTemplates', 'newTemplates')
clear all

```

Pertama-tama semua citra *template* akan dibaca, lalu dimasukkan ke dalam *array* sesuai jenis *template*, yaitu huruf atau angka. Perlu diketahui setiap *template* merupakan matriks berukuran sama (42×24) dan merupakan foto biner. Kemudian, semua *template* dijadikan satu dan disimpan pada variabel *newTemplates*, sehingga menghasilkan *file* *newTemplates.mat*, yang merupakan *dump* dari variabel yang sudah dibuat pada *script* ini.

2. Deteksi Plat (*Plate Detection*)

Program dapat menerima masukan berupa citra kendaraan. Langkah yang dilakukan adalah dengan mendeteksi bagian citra yang memuat plat nomor kendaraan sebagai *region of interest* (ROI). Berikut rincian implementasinya.

detectPlate.m

```

function plateImg = detectPlate(imgIn)
%detectPlate - Finds where the plate number is

```

```

%
% Syntax: plateImg = detectPlate(imgIn)
%
% This function takes an image as input and then return a greyscale
of plate image

% Convert image to grayscale first.
imgray = im2gray(imgIn);

[h, w] = size(imgray);

diskSECoeff = max(1, round(1/30 * h));

% Apply contrast enhancement using histogram equalization
imhisteq = histogram_equalization(imgray);

% Morphological image opening
SE = strel('disk', diskSECoeff);
openedImg = imopen(imhisteq, SE);

% Subtract operation
subtracted = imsubtract(imhisteq, openedImg);

% Image binarization using Otsu's method
imbin = imbinarize(subtracted);

% Edge detection using Sobel
% imedge = edge(imbin, 'sobel');
imedge = detect_edge(imbin, 'sobel', [], [], [], [], [], []);

% Perform image dilation
dilationSE = strel('line', 4, 45);
dilated = imdilate(imedge, dilationSE);

% Fill the holes
filled = imfill(dilated, 'holes');

% Morphological image opening onto the filled image
openingSE2 = strel('disk', round(diskSECoeff / 2));
openedImg2 = imopen(filled, openingSE2);

% Erode the opened image
eroded = imerode(openedImg2, dilationSE);

regions = regionprops(eroded, 'BoundingBox', 'Area', 'Image');

boundingBox = getLargestBB(regions);

imcr = imcrop(imgray, boundingBox);

```

```

croppedbin = imbinarize(imcr);

% Get regions/bounding boxes
% If the white part dominates, then apply negative filter to make
background black
% Then apply connected component analysis and bounding box analysis
onto it
if bwarea(croppedbin) > bwarea(~croppedbin)
    segments = regionprops(~croppedbin, 'BoundingBox', 'Area',
'Image');
else
    segments = regionprops(croppedbin, 'BoundingBox', 'Area',
'Image');
end

finalBB = getLargestBB(segments);

plateImg = imcrop(imcr, finalBB);
end

```

Fungsi ini menerima input citra kendaraan. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Mengubah citra ke grayscale menggunakan `im2gray`.
2. Melakukan *contrast enhancement* menggunakan *histogram equalization* untuk mempertajam gambar dan menghilangkan sedikit derau yang membuat kualitas gambar terlihat rendah.
3. Melakukan operasi *morphological image opening* dengan menggunakan *disc shaped structuring element* pada citra yang sudah dilakukan *contrast enhancement*.
4. Melakukan operasi pengurangan citra dengan mengurangi citra yang sudah dilakukan operasi *morphological opening* dengan citra yang sudah dilakukan *contrast enhancement*, sehingga plat nomor bisa di-*highlight*.
5. Melakukan *binarization* atau membuat citra yang sudah dilakukan pengurangan menjadi citra biner dengan menggunakan metode Otsu.
6. Melakukan *edge detection* pada citra biner dengan memanfaatkan operator Sobel.
7. Melakukan operasi *dilation* pada citra hasil *edge detection*, lalu mengisi celah-celah yang masih ada dengan menggunakan fungsi `imfill`.
8. Melakukan operasi *morphological image opening* dengan jari-jari *structuring element* yang lebih kecil untuk mendeteksi kandidat *region* plat.
9. Melakukan operasi *morphological erode* untuk deteksi lebih dekat pada kandidat plat.
10. Melakukan *bounding box analysis* dengan mengambil *bounding box* yang memiliki luas paling besar dan nilai *width* lebih besar dari nilai *height*. Citra yang terdapat pada *bounding box* tersebut akan di-*crop*.

11. Melakukan *image binarization* pada citra hasil *cropping*, lalu dilakukan pengubahan warna *background*. Pada implementasi kami, *background* harus memiliki warna hitam, sehingga jika pada kenyataannya lebih banyak warna putih, maka dilakukan *negative filter* sehingga warna hitam tetap memiliki luas paling besar.
12. Melakukan *bounding box analysis* lagi untuk lebih mempersempit *region* sehingga menjadi benar-benar hanya citra sebuah plat.

3. Segmentasi Karakter (*Character Segmentation*)

Setelah didapat citra yang berisikan hanya plat saja, maka akan dilakukan penentuan mana sajakah bagian dari plat tersebut yang merupakan karakter-karakter utama, yaitu bagian atas dari plat. Implementasinya pada MATLAB terdefinisi pada fungsi di bawah ini.

segmentLetter.m

```
function segments = segmentLetter(imgIn)
    %segmentLetter - Return segments from the plate image
    %
    % Syntax: segments = segmentLetter(imgIn)

    % Apply binarization towards the image
    imbin = imbinarize(imgIn);

    % Get the size of the image
    [h, w] = size(imbin);

    % Clean the small objects that have pixel area less than 0.1% of
    image area
    imclean = bwareaopen(imbin, round(0.001 * h * w));

    % Get regions/bounding boxes
    % If the white part dominates, then apply negative filter to make
    background black
    % Then apply connected component analysis and bounding box analysis
    onto it
    if bwarea(imclean) > bwarea(~imclean)
        segments = regionprops(~imclean, 'BoundingBox', 'Area', 'Image');
    else
        segments = regionprops(imclean, 'BoundingBox', 'Area', 'Image');
    end
end
```

Pertama-tama, dilakukan *image binarization* pada citra masukan. Lalu, dilakukan pembersihan citra dengan menggunakan operasi *opening* pada citra, dengan menentukan bahwa bagian citra yang memiliki luas lebih kecil dari 0,1% dari luas citra akan dianggap *small objects*. Kemudian citra yang sudah

dibersihkan dari *small objects* akan dilakukan *connected component analysis* dan *bounding box analysis*, untuk menghasilkan *region-region* yang merupakan kandidat karakter. Lalu, fungsi ini digunakan pada *main program* sebagai berikut.

```
function letters = getOutput(app, im)
    % If not a plate, detect the plate first
    if strcmp(app.AlreadyPlateSwitch.Value, 'No')
        plateIm = detectPlate(im);
        imagesc(app.BoundingBoxImageAxes, plateIm);
    % Else apply grayscaling directly
    else
        plateIm = im2gray(im);
        imagesc(app.BoundingBoxImageAxes, im);
    end

    segments = segmentLetter(plateIm);
    letters = [];

    [h, ~] = size(plateIm);

    for k = 1:length(segments)

        ow = length(segments(k).Image(1, :));
        oh = length(segments(k).Image(:, 1));
        ratio = ow/oh;

        % The height of bounding box should be in range
        (0.26*h, 0.8*h)
        % and its weight should be less than or equal with
        its height
        if oh < (0.8 * h) && oh > (0.26 * h) && ow <= oh
            && ratio > 0.1
                ratio = ow/oh;
                thisBB = segments(k).BoundingBox;
                rectangle(app.BoundingBoxImageAxes,
                    'Position', [thisBB(1), thisBB(2), thisBB(3), thisBB(4)], ...
                        'EdgeColor', 'g', 'LineWidth', 2)

                thisLetter = segments(k).Image;

                detect = detectLetter(thisLetter);
                letters = [letters detect];
            end
        end
    end
end
```


end

Proses penentuan apakah *region* yang dihasilkan merupakan karakter adalah jika *region* tersebut memenuhi kondisi sebagai berikut.

- Tinggi dari *region* berada pada rentang $(0.26 \times h, 0.8 \times h)$.
- Lebar dari *region* lebih kecil atau sama dengan tingginya. Hal ini dikarenakan sifat dari sebuah karakter huruf dan angka tidak mungkin *stretched* secara horizontal.
- Perbandingan lebar dari *region* dan tingginya lebih dari 0,1. Hal ini bertujuan untuk menghindari *region* yang terlalu sempit, yang tidak mungkin merupakan sebuah karakter.

4. Pengenalan Karakter (*Character Recognition*)

Setelah mendapatkan segmentasi *bounding box* per karakter, dilakukan pengenalan tiap karakter yang didapat dengan menggunakan *template matching*. *Template* yang digunakan adalah yang telah di-*generate* dalam langkah pembuatan template pada **createTemplates.m**. Rincian implementasinya adalah sebagai berikut.

detectLetter.m

```
function letter = detectLetter(imgIn)
% Detect Letter using template matching
%

% Load templates
load newTemplates;
tempLabel = [
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'I', 'J', 'K', ...
    'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', ...
    'X', 'Y', 'Z', ...
    '1', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '0'
];
col = 24;

% resize input image
imgIn = imresize(imgIn, [42 24]);

rec=[];

for n=1:39
    % access one template
    temp = imbinarize(newTemplates(:, ((n - 1) * col) + 1:n * col));

    % find correlation
    cor = corr2(temp, imgIn);
```

```

        % append correlation
        rec = [rec cor];
    end

    % find index with maximum correlation
    ind = find(rec == max(rec));

    % find label
    letter = tempLabel(ind);
end

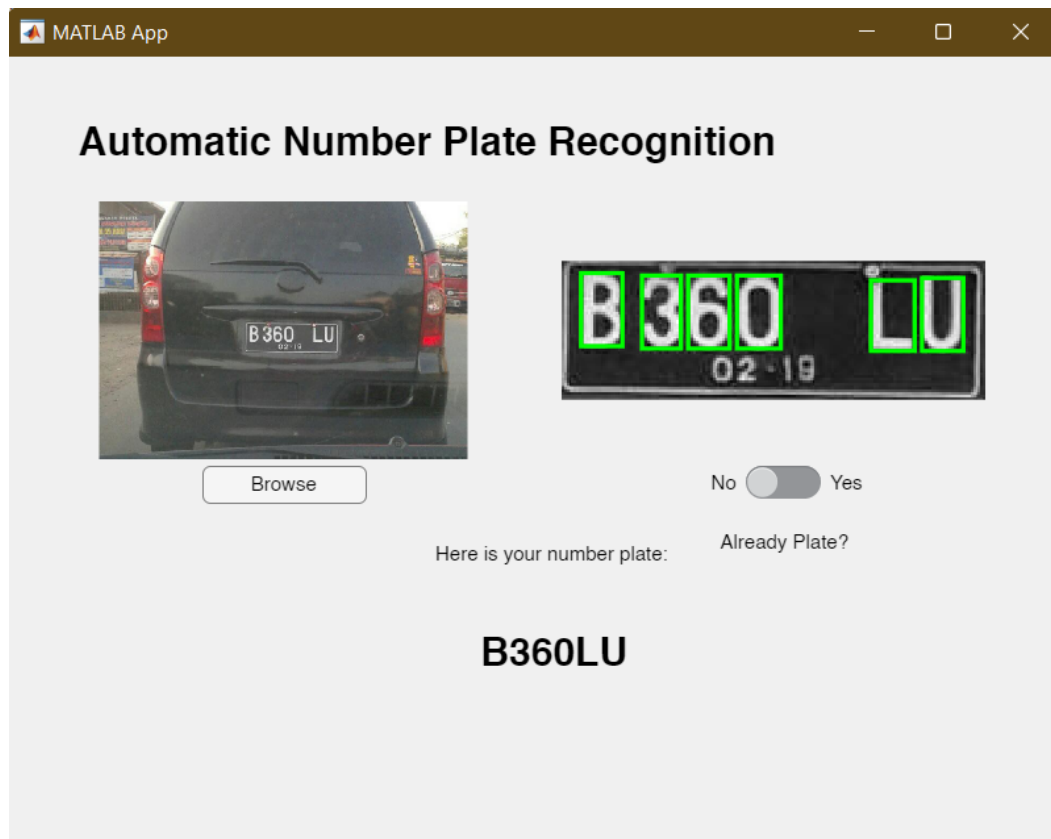
```

Fungsi ini menerima masukan citra satu buah karakter hasil segmentasi karakter menggunakan *bounding box*, dan melakukan langkah-langkah berikut:

1. *Load* template dari file newTemplates.mat hasil createTemplates
2. Mengubah ukuran input citra menjadi matriks berukuran 42×24, sesuai dengan ukuran tiap karakter pada *template*.
3. Melakukan pencocokan dengan semua karakter pada *template*, dengan menghitung korelasi antara citra masukan dengan semua *template*. Kedua citra sudah melewati *thresholding* sehingga merupakan citra biner.
4. Mengambil label dari *template* yang memiliki korelasi tertinggi (paling mirip) dengan citra masukan.
5. Mengembalikan label hasil prediksi.

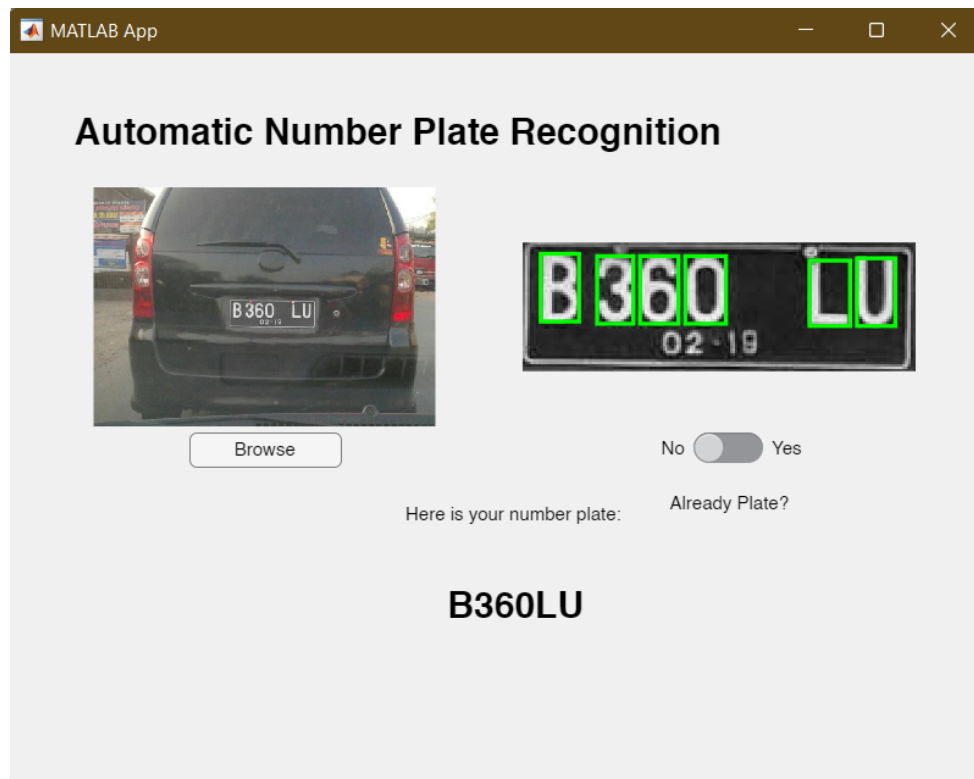
Screenshot Antarmuka Program

Berikut *screenshot* yang menunjukkan tampilan GUI dari aplikasi yang dibuat.

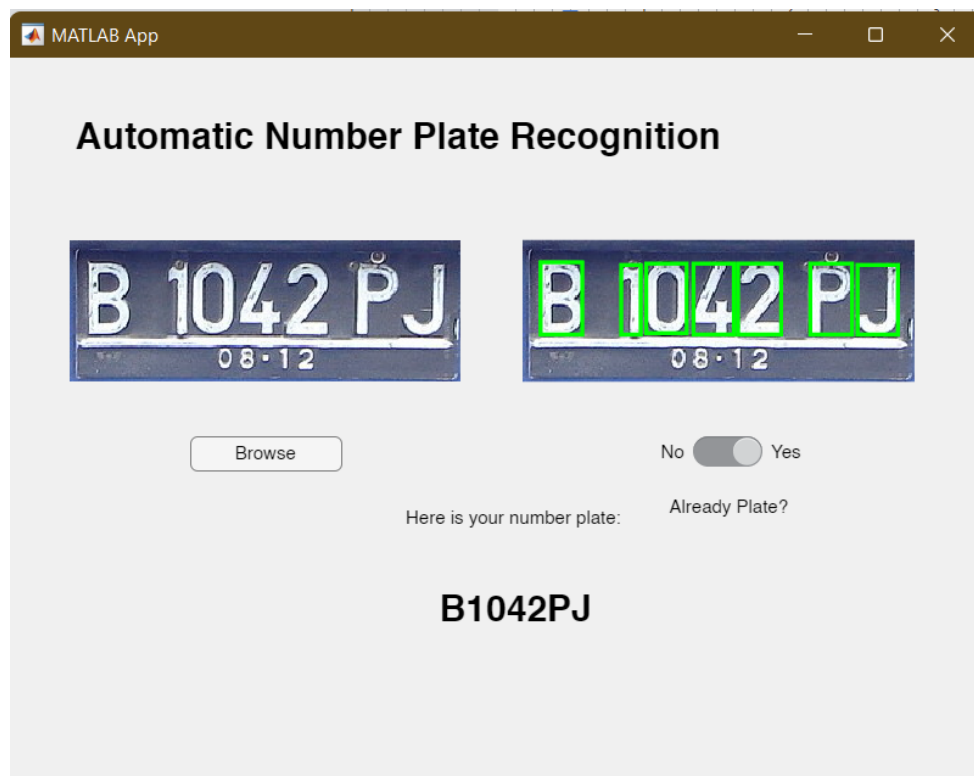


Contoh Hasil Eksekusi Program

1. Tidak Secara Langsung Gambar Plat



2. Secara Langsung Citra Plat



Analisis Hasil Pengenalan




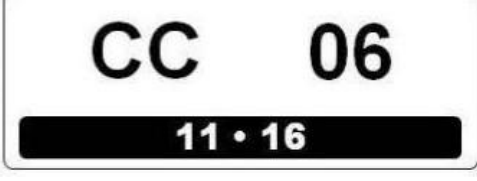
Berikut ini adalah hasil pengenalan yang dilakukan oleh aplikasi yang kami buat dan hasil perhitungan karakter yang mengalami *insertion error* (*I*), *deletion error* (*D*), dan *substitution error* (*S*) sesuai rumus dari *Character Error Rate* (*CER*). Jumlah karakter pada plat dinyatakan dengan variabel *N*. Rumus *CER* dan akurasi berdasarkan *CER* didefinisikan sebagai berikut.

$$CER = \frac{I+D+S}{N}$$

$$acc = 1 - CER$$


Hasil pengenalan ini dibagi menjadi 2 tabel. Tabel pertama menggunakan citra yang bukan merupakan plat asli, sedangkan tabel kedua menggunakan citra yang merupakan plat asli.

1. Hasil Pengenalan dan Akurasi untuk Citra Plat Secara Langsung

| No | Citra Asli | Hasil Prediksi | Perhitungan |
|----|---|----------------|---|
| 1 |  | A888AH | I = 0, D = 0, S = 0, N = 6 Akurasi = 100% |
| 2 |  | B212JI | I = 0, D = 0, S = 0, N = 6 Akurasi = 100% |
| 3 |  | L4821UJ | I = 0, D = 0, S = 0, N = 6 Akurasi = 100% |
| 4 |  | CC06 | I = 0, D = 0, S = 0, N = 4 Akurasi = 100% |

Tabel 1. Hasil pengenalan dengan citra plat yang tidak asli.

| No | Citra Asli | Hasil Prediksi | Perhitungan |
|----|---|----------------|--|
| 1 |  | D6349TAP | $I = 0, D = 0, S = 1, N = 8$ Akurasi = 87,5% |
| 2 |  | B9320VUA | $I = 0, D = 0, S = 0, N = 8$ Akurasi = 100% |
| 3 |  | B1042PJ | $I = 0, D = 0, S = 0, N = 7$ Akurasi = 100% |
| 4 |  | AC5T40VAA | $I = 0, D = 0, S = 2, N = 9$ Akurasi = 77,78% |
| 5 |  | AB1895KA | $I = 0, D = 0, S = 0, N = 8$ Akurasi = 100% |
| 6 |  | L34838X | $I = 0, D = 0, S = 1, N = 7$ Akurasi = 87,5% |
| 7 |  | L3368K | $I = 0, D = 0, S = 0, N = 6$ Akurasi = 100% |
| 8 |  | B360LU | $I = 0, D = 0, S = 0, N = 6$ Akurasi = 100% |

| | | | |
|----|---|------------|---|
| 9 |  | MH12DE1433 | I = 0, D = 0, S = 0, N = 10 Akurasi = 100% |
| 10 |  | B6970Z | I = 0, D = 0, S = 1, N = 6 Akurasi = 83,33% |

Tabel 2. Hasil pengenalan dengan citra plat yang asli.

Dari Tabel 1 dan Tabel 2 dapat dihitung akurasi rata-rata tiap tabel sebagai berikut.

$$mean(acc_1) = 100\%$$


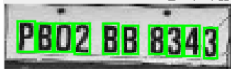












$$mean(acc_2) \approx 93,61\%$$

Jika hasil dari Tabel 1 dan Tabel 2 digabungkan, akan didapat akurasi rata-rata sebagai berikut.

$$mean(acc_{1,2}) \approx 95,44\%$$

2. Hasil Pengenalan dan Akurasi untuk Citra Plat Secara Tidak Langsung

| No | Citra Asli | Hasil Segmentasi | Hasil Prediksi | Perhitungan |
|----|---|---|----------------|---|
| 1 |  |  | B360LU | I = 0, D = 0, S = 0, N = 6 Akurasi = 100% |
| 2 |  |  | A875ML | I = 0, D = 0, S = 2, N = 6 Akurasi = 66,67% |
| 3 |  |  | AA189378 | I = 0, D = 0, S = 2, N = 8 Akurasi = 75% |

| | | | | |
|----|---|---|-------------------|--|
| 4 |  |  | P8D26B8343 | I = 0, D = 0, S = 3, N = 10 Akurasi = 70% |
| 5 |  |  | EK090CN | I = 0, D = 0, S = 0, N = 7 Akurasi = 100% |
| 6 |  |  | MH12DE1433 | I = 0, D = 0, S = 0, N = 10 Akurasi = 100% |
| 7 |  |  | FMP0900404 | I = 1, D = 1, S = 2, N = 10 Akurasi = 60% |
| 8 |  |  | ARYAN1 | I = 0, D = 0, S = 0, N = 6 Akurasi = 100% |
| 9 |  |  | B3ACH | I = 0, D = 0, S = 0, N = 5 Akurasi = 100% |
| 10 |  |  | ENWDVID | I = 0, D = 0, S = 1, N = 7 Akurasi = 85,71% |

Tabel 3. Hasil pengenalan dengan citra plat tidak langsung.

Berikut ini adalah akurasi rata-rata dari Tabel 3.

$$mean(acc_3) \approx 85.74\%$$

3. Analisis

Hasil akurasi rata-rata *CER* untuk citra plat secara langsung adalah sebesar 95,44% untuk keseluruhan kasus plat asli dan tidak asli dan 93,61% untuk kasus hanya plat asli. Kemudian, untuk citra plat secara tidak langsung sebesar 85,74%. Angka akurasi ini sudah cukup baik. Namun, masih terdapat kesalahan dalam mendeteksi karakter-karakter tertentu.

Kesalahan deteksi mayoritas terjadi pada bentuk karakter yang mirip, seperti 8-B-D, A-4, O-D-Q, 1-I, dan lain-lain. Kesalahan ini dapat terjadi karena teknik yang digunakan adalah *template matching*, sehingga huruf yang mirip dapat memiliki korelasi lebih tinggi daripada huruf yang seharusnya menjadi hasil deteksi. Untuk mengatasi hal ini, diperlukan kombinasi template yang beragam untuk setiap karakter agar berbagai kasus pada setiap karakter dapat teratasi dengan baik.

Selain itu, akurasi *CER* deteksi pada “citra plat secara tidak langsung” memiliki angka yang lebih kecil, karena terkadang pendeteksian platnya sendiri tidak benar-benar bisa mendeteksi platnya, baik karena tepi plat yang kurang jelas ataupun terdapat objek lain yang lebih dominan/mengganggu deteksi tepi. Walaupun tepat, tidak jarang juga daerah di luar plat masih ikut terdeteksi. Oleh karena itu, hasil deteksi untuk plat terkadang menghasilkan hasil segmentasi yang tidak ideal, seperti memiliki noise, ataupun hasil pemotongan gambarnya tidak lurus. Hal ini menyebabkan gangguan dalam mendeteksi karakter berdasarkan hasil segmentasi plat, karena fungsi `segmentLetter` mengasumsikan *input image* yang merupakan gambar citra yang sangat rapi dan tidak ada bagian citra yang di luar ROI (*plate* itu sendiri).

Pranala Kode Program pada GitHub

Kode program dapat diakses pada [pranala berikut ini](#).

Pembagian tugas adalah sebagai berikut.

| | |
|----------|--|
| 13518089 | <ul style="list-style-type: none">- Mengimplementasikan fungsi detectLetter- Memodifikasi fungsi createTemplates- Refactor fungsi detect_edge- Refactor fungsi histogram_equalization |
| 13518149 | <ul style="list-style-type: none">- Mengimplementasikan fungsi createTemplates- Mengimplementasikan fungsi detectPlate- Refactor fungsi getLargestBB- Mengimplementasikan GUI |

Referensi

- Citra diambil dari
 - [Berkas:Plat Nomor Nganjuk \(3 Huruf\).jpg - Wikipedia bahasa Indonesia, ensiklopedia bebas](#)
 - [Jual Sticker Huruf Plat Nomor Mobil Scotlight - - Super01 Grosir | Tokopedia](#)
 - [Menyerap Informasi Dari Plat Nomor Polisi - Citizen6 Liputan6.com](#)
 - [Vehicle registration plates of Indonesia - Wikipedia](#)
 - [License Plates of Indonesia \(worldlicenseplates.com\)](#)
 - [Vehicle registration plates of Indonesia - Wikiwand](#)
 - [Car plate ANPR LPR AHD cameras for number plates | DSE CCTV surveillance Italy EU](#)
 - [License Plate Detection And Recognition Using OpenCv And Pytesseract | Engineering Education \(EngEd\) Program | Section Preparation Instruction \(mecs-press.org\)](#)
 - [230064741.pdf \(core.ac.uk\)](#)
 - <https://semuatentangprovinsi.blogspot.com/2016/04/kode-plat-nomor-kendaraan-setiap.html>