

# SISTEM KEMUDI

KENDARAAN OTOMATIS PADA GAME 3D

**ARIANTO** ANGGORO

**AWAL** RAIS SANUBARI

**DESTI** MINA RAHAYU

**GEMA** WAHYU SAPUTRA

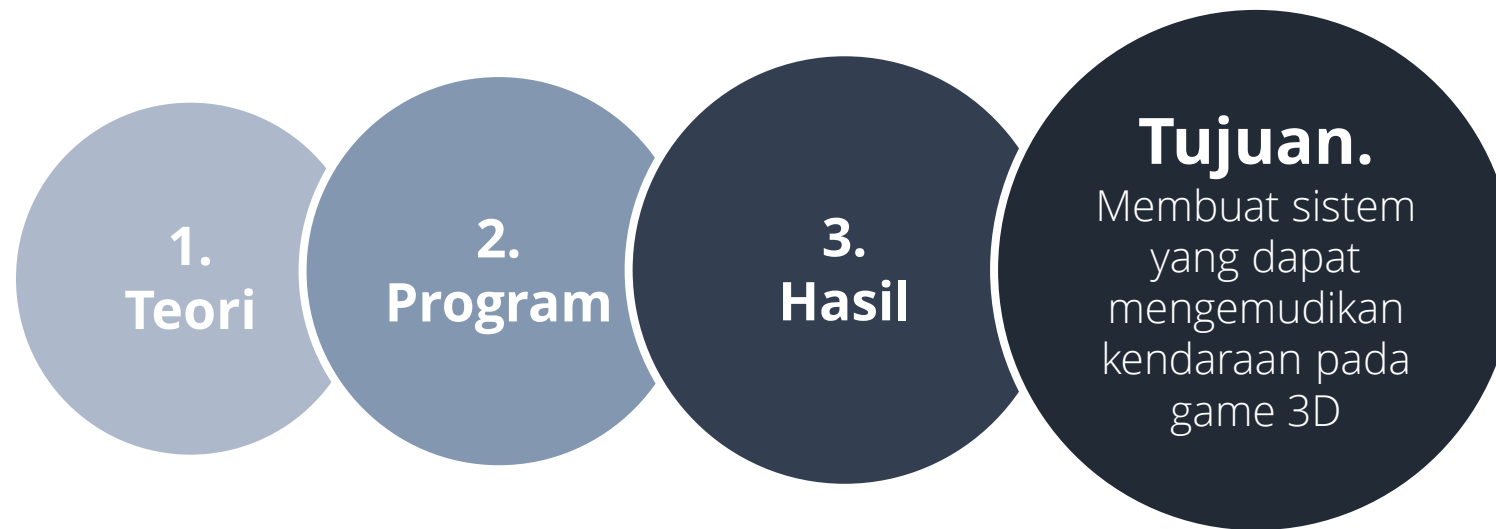
**KUNCORO** TRIANDONO MUKTI

**ILHAM** MAJID RABBANI



# PENDAHULUAN

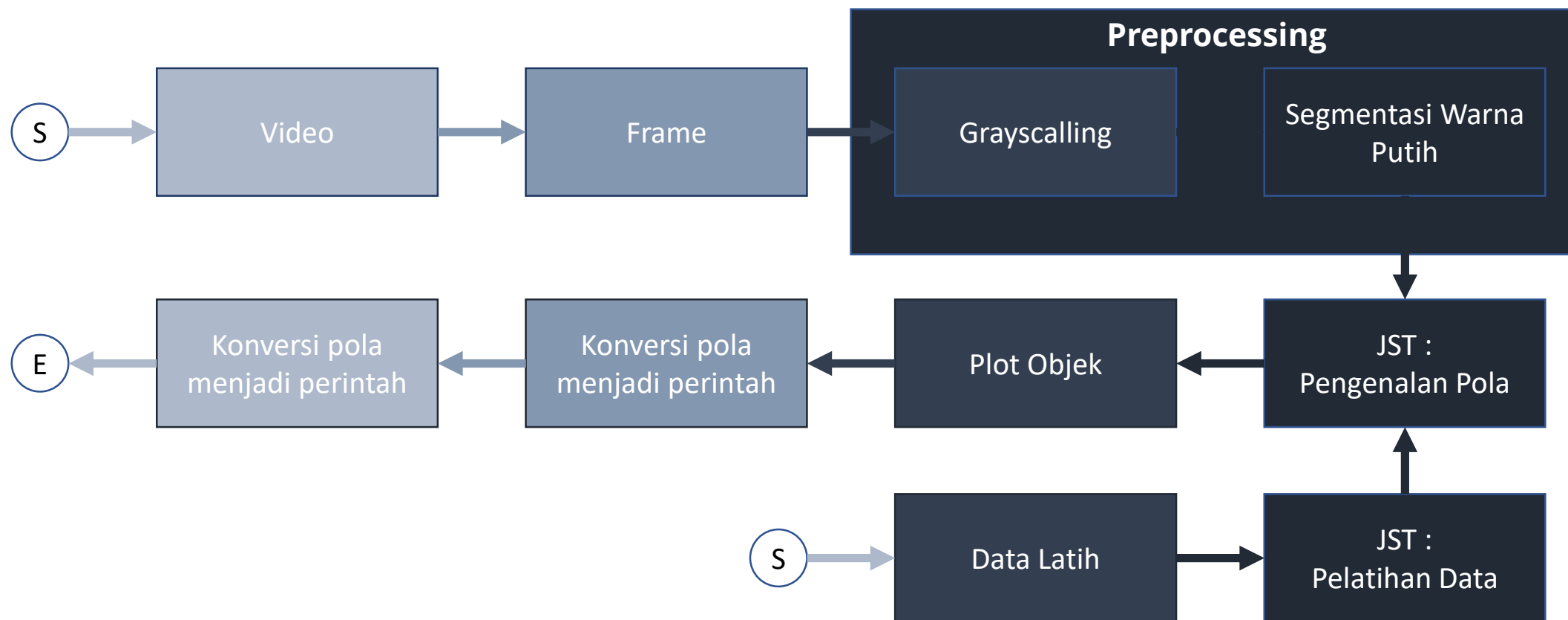
Apa yang akan dibahas





# PENDAHULUAN

Rancangan Umum Sistem

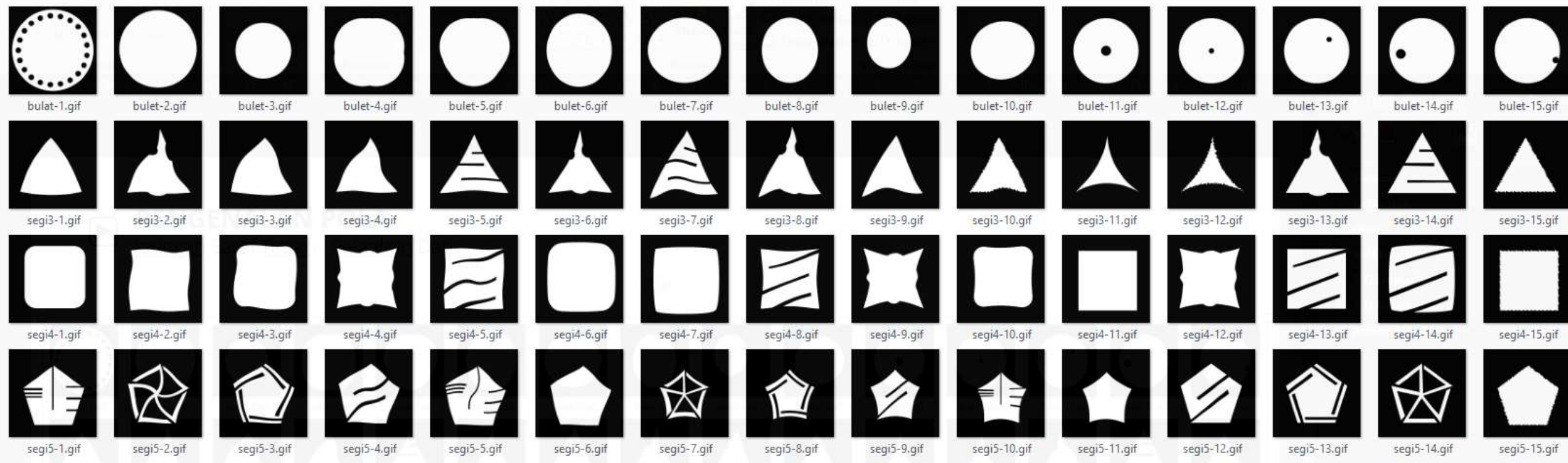




# PENGENALAN POLA

Jaringan Saraf Tiruan : Pelatihan Data

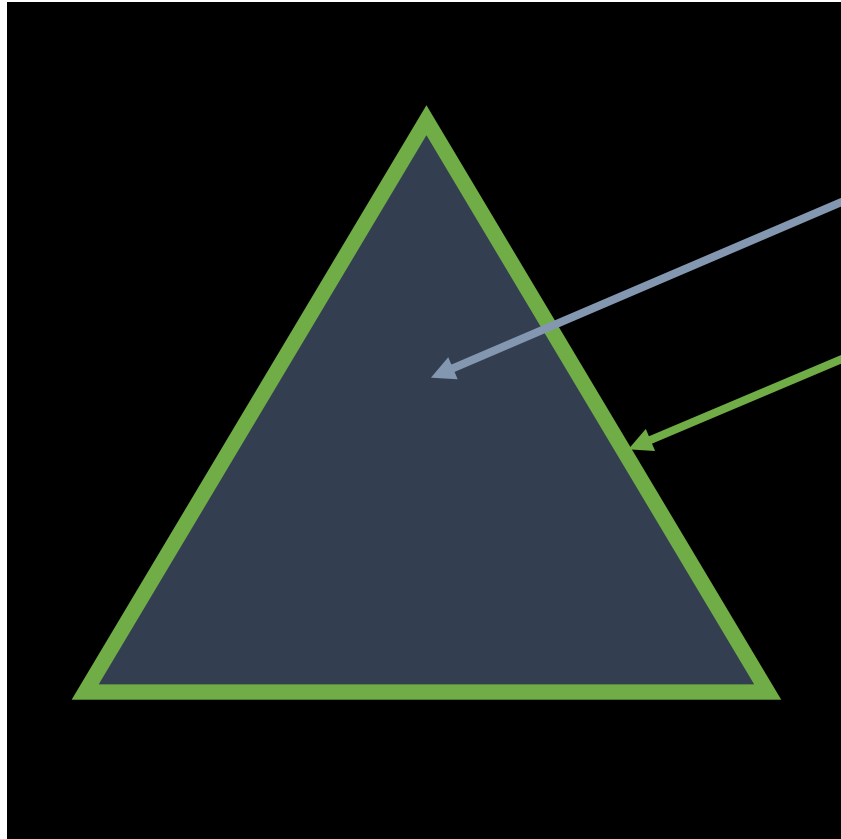
Kami menggunakan data latih berupa gambar dengan 4 pola berbeda dengan jumlah 64 file yang berwarna hitam putih.





# PENGENALAN POLA

Jaringan Saraf Tiruan : Pelatihan Data : Input



area =

88218

perimeter =

1.3310e+03

metric =

0.6258

eccentricity =

0.2207

```
2 % DATA LATIH
3 % Bulat = 0
4 % Segitiga = 3
5 % Segiempat = 4
6 % Segilima = 5
7
8 % Membaca seluruh Citra sebagai data latih
9 folder_citra = 'data latih';
10 namafile = dir(fullfile(folder_citra, '*.gif'));
11 total_citra = numel(namafile);
12
13 % Menentukan jumlah array setiap variabel
14 area = zeros(1,total_citra);
15 perimeter = zeros(1,total_citra);
16 metric = zeros(1,total_citra);
17 eccentricity = zeros(1,total_citra);
18
19 % Preprocessing
20 for n = 1:total_citra
21     nama= fullfile(folder_citra, namafile(n).name);
22     citra = logical(imread(nama));
23     citra = bwconvhull(citra,'objects');
24     % Mendapatkan Area, Perimeter, Eccentricity
25     stats = regionprops(citra,'Area','Perimeter','Eccentricity');
26     area(n) = stats.Area;
27     perimeter(n) = stats.Perimeter;
28     % METRIC
29     % merupakan nilai perbandingan antara luas dan keliling objek.
30     % Metric memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk
31     % memanjang/mendekati bentuk garis lurus, nilai metricnya mendekati
32     % angka 0, sedangkan objek yang berbentuk bulat/lingkaran,
33     % nilai metricnya mendekati angka 1.
34     metric(n) = 4*pi*area(n)/(perimeter(n)^2);
35     % ECCENTRICITY
36     % merupakan nilai perbandingan antara jarak foci ellips
37     % minor dengan foci ellips mayor suatu objek. Eccentricity memiliki
38     % rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang/
39     % mendekati bentuk garis lurus, nilai eccentricitynya mendekati angka 1
40     % sedangkan objek yang berbentuk bulat/lingkaran, nilai eccentricitynya
41     % mendekati angka 0.
42     eccentricity(n) = stats.Eccentricity;
43 end
```



# PENGENALAN POLA

Jaringan Saraf Tiruan : Pelatihan Data : Target

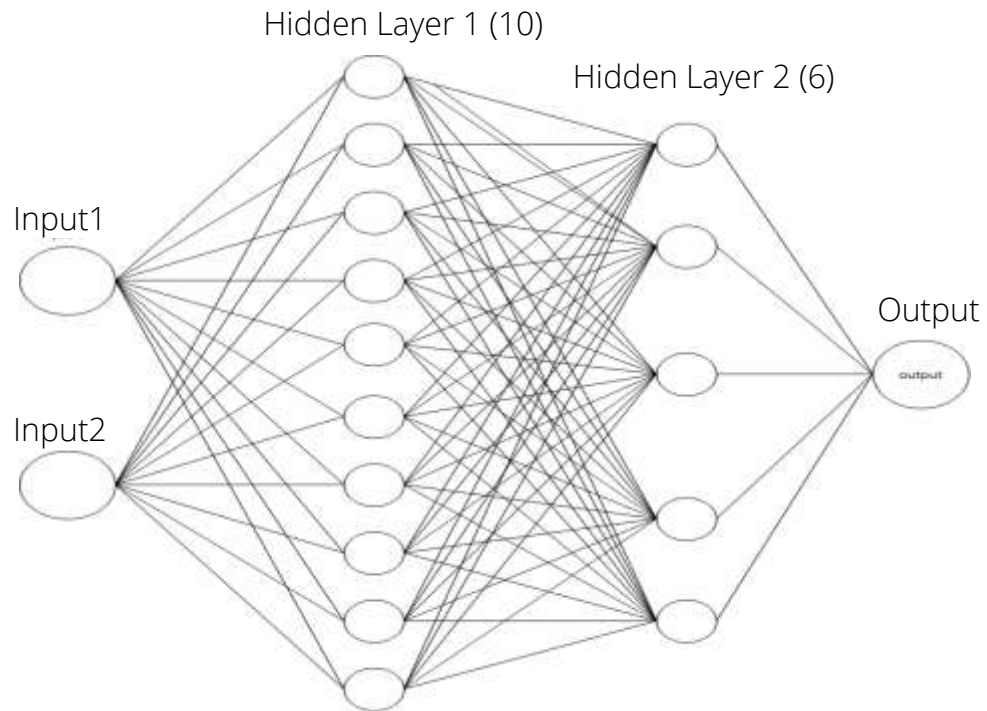
Pola	Urutan File	Target
Lingkaran	1-16	0
Segitiga	17-32	3
Segiempat	33-48	4
Segilima	49-64	5

```
47 % Menentukan target untuk NN
48 - target = zeros(1,64);
49 % Menentukan target file 1 - 16 merupakan 0 (Bulat)
50 - target(:,1:16) = 0;
51 % Menentukan target file 17 - 32 merupakan 3 (Segitiga)
52 - target(:,17:32) = 3;
53 % Menentukan target file 33 - 48 merupakan 4 (Segiempat)
54 - target(:,33:48) = 4;
55 % Menentukan target file 49 - 64 merupakan 5 (Segilima)
56 - target(:,49:64) = 5;
```



# PENGENALAN POLA

Jaringan Saraf Tiruan : Pelatihan Data : Struktur Jaringan



```
58 % Neural Network
59 - net = newff(input,target,[10 6],{'logsig','logsig'},'trainlm');
60 - net.trainParam.epochs = 100;
61 - net.trainParam.goal = 1e-5;
62 - net = train(net,input,target);
63 - output = round(sim(net,input));
64
65 % Menyimpan data NN kedalam file net.mat
66 - save net.mat net
67
68 % Menentukan tingkat Akurasi
69 - [m,Target] = find(output==target);
70 - Akurasi = sum(m)/total_images*100
```

## Command Window

```
>> cvTrain

Akurasi =

    46.6667

ans =

    46.6667

>> cvTrain;

Akurasi =

    96.6667
```



## PREPROCESSING

Konversi ke skala keabuan



200	240	255
0	0	255
20	250	240

R

236	240	255
0	0	255
234	167	240

G

200	140	255
0	0	255
20	210	240

B

$$GS = \frac{R * Wr + G * Wg + B * Wb}{3}$$

$$GS(1,1) = \frac{200 + 236 + 200}{3} = 212$$







## PREPROCESSING

Konversi ke skala keabuan

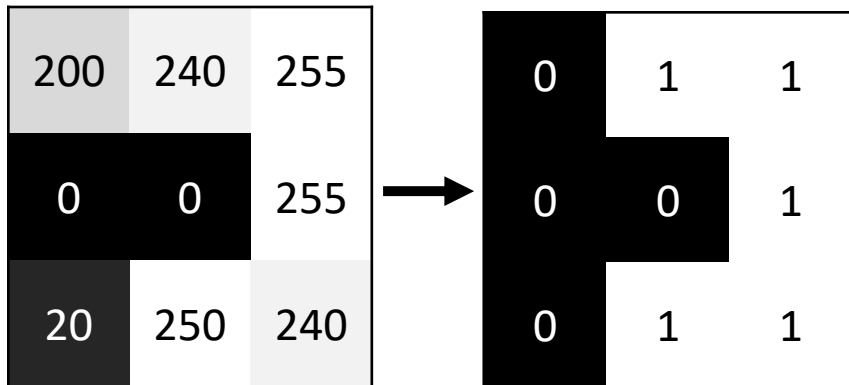




## PREPROCESSING

Segmentasi warna Putih

Tahap ini dilakukan guna mendapatkan pixel berwarna putih dengan nilai **threshold** 250, pixel yang bernilai 0-5 akan diubah menjadi pixel berwarna putih 1 selain itu berwarna hitam 0.

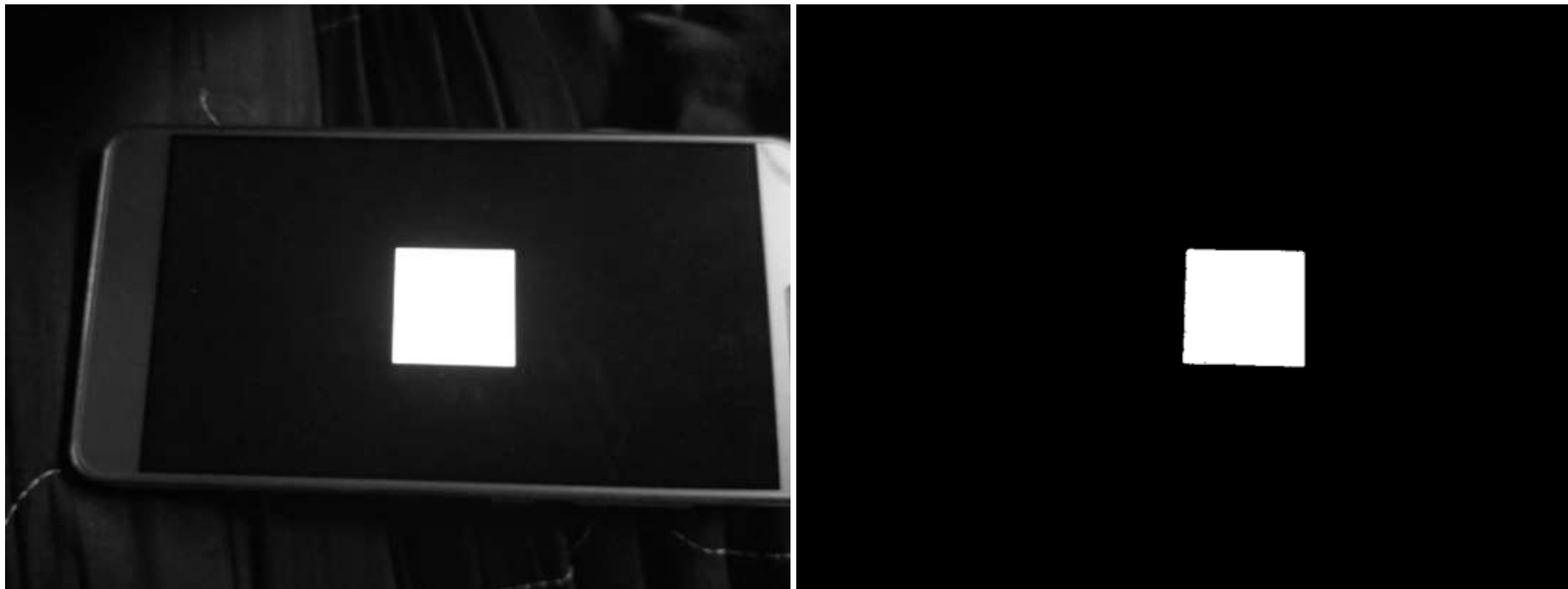


```
1 function output = cvFindBW(x)
2     % Mendapatkan Baris, Kolom dan Channel
3     [b,k,c] = size(x);
4     % Menyiapkan citra output sesuai dengan uk citra input
5     output = zeros(b,k);
6     for i = 1:b
7         for j = 1:k
8             % Mencari warna hitam pada Channel RGB
9             % untuk putih mendekati 255, hitam 0
10            if x(i,j) >= 250
11                % Jika terdapat warna hitam maka (x,y) akan diberi warna Putih
12                output(i,j) = 1;
13            else
14                % Selain itu bernilai 0
15                output(i,j) = 0;
16            end
17        end
18    end
```



## PREPROCESSING

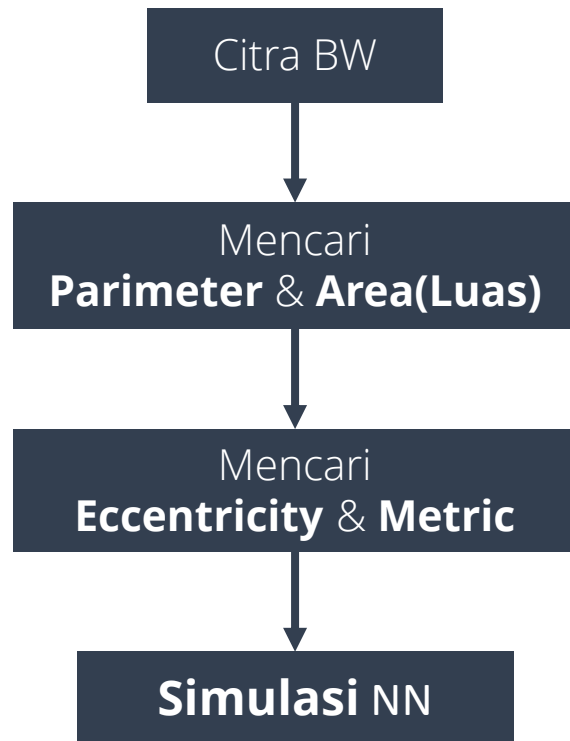
Segmentasi warna Putih





# PENGENALAN POLA

Jaringan Saraf Tiruan : Pengujian

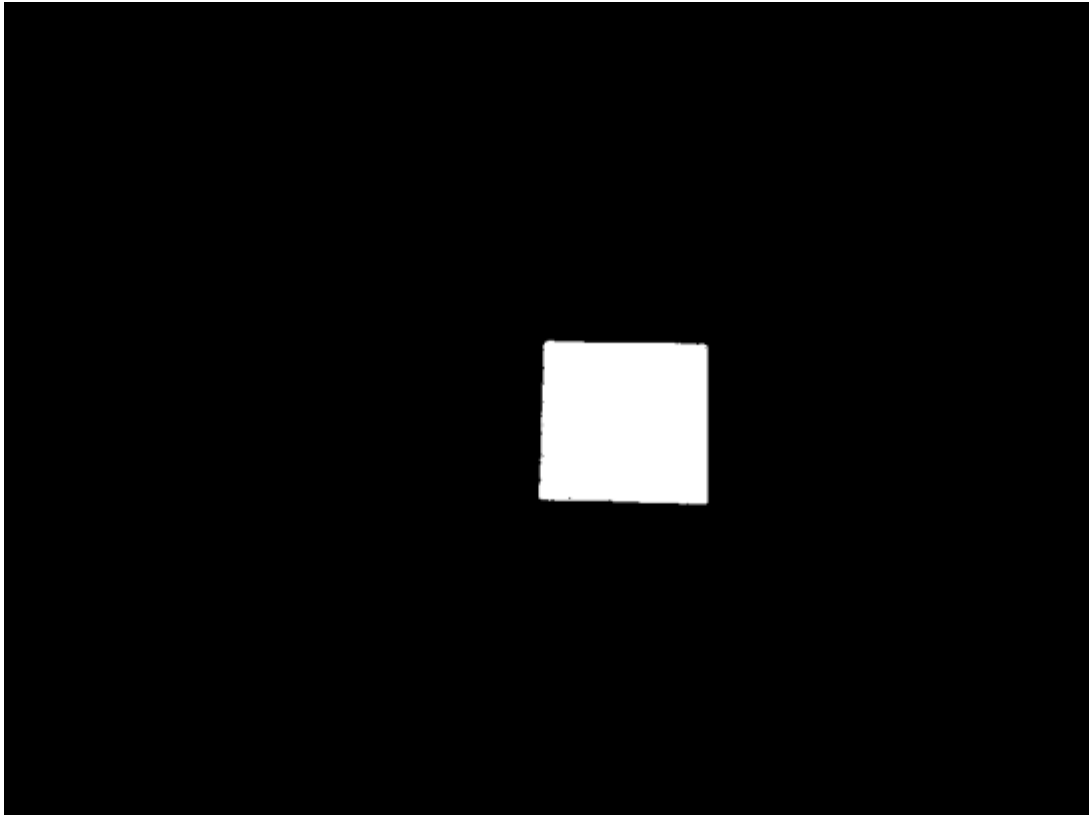


```
2 % Membuat kondisi saat gambar tanpa objek
3 - if data(:, :, :) == 0
4 -     pola = 6; % Nilai random selain kode untuk bentuk pola
5 - else
6     % Menggunakan data sebagai citra masukkan
7 -     citra = logical(data);
8 -     citra = bwconvhull(citra, 'objects');
9     % Mendapatkan nilai Area, Perimeter & Eccentricity
10 -    stats = regionprops(citra, 'Area', 'Perimeter', 'Eccentricity');
11 -    area = stats.Area;
12 -    perimeter = stats.Perimeter;
13 -    metric = (4*pi*area) ./ (perimeter.^2);
14 -    eccentricity = stats.Eccentricity;
15
16     % Memasukkan data inputan
17 -    input = [metric;eccentricity];
18     % Meload variabel net
19 -    load net
20     % Mendapatkan Output
21 -    pola = round(sim(net,input));
22 - end
```



# PENGENALAN POLA

Jaringan Saraf Tiruan : Pengujian



## Command Window

```
>> a = imread('Pola/7939.JPG');  
>> gs = cvGS(a,1,1,1);  
>> bw = cvFindBW(gs);  
>> pola = cvPR(bw);  
>> pola
```

```
pola =
```

```
4
```



## KONVERSI

Konversi angka menjadi teks

```
Command Window

pola =

     4

>> cvCnv(pola)

pola =

'Segiempat'
```

```
Editor - E:\My Work\Program\Automatic Self Driving Patter R

1 function cvCnv(x)
2 % Konversi Pola angka menjadi Teks
3 % Bulat = 0 = Kanan
4 % Segitiga = 3 = Maju
5 % Segiempat = 4 = Kiri
6 % Segilima = 5 = Mundur
7 if x == 0
8     pola = 'Bulat'
9 elseif x == 3
10    pola = 'Segitiga'
11 elseif x == 4
12    pola = 'Segiempat'
13 elseif x == 5
14    pola = 'Segilima'
15 else
16    pola = 'Tidak terdeteksi'
17 end
```



## PLOT OBJEK

Menandai objek yang merupakan pola

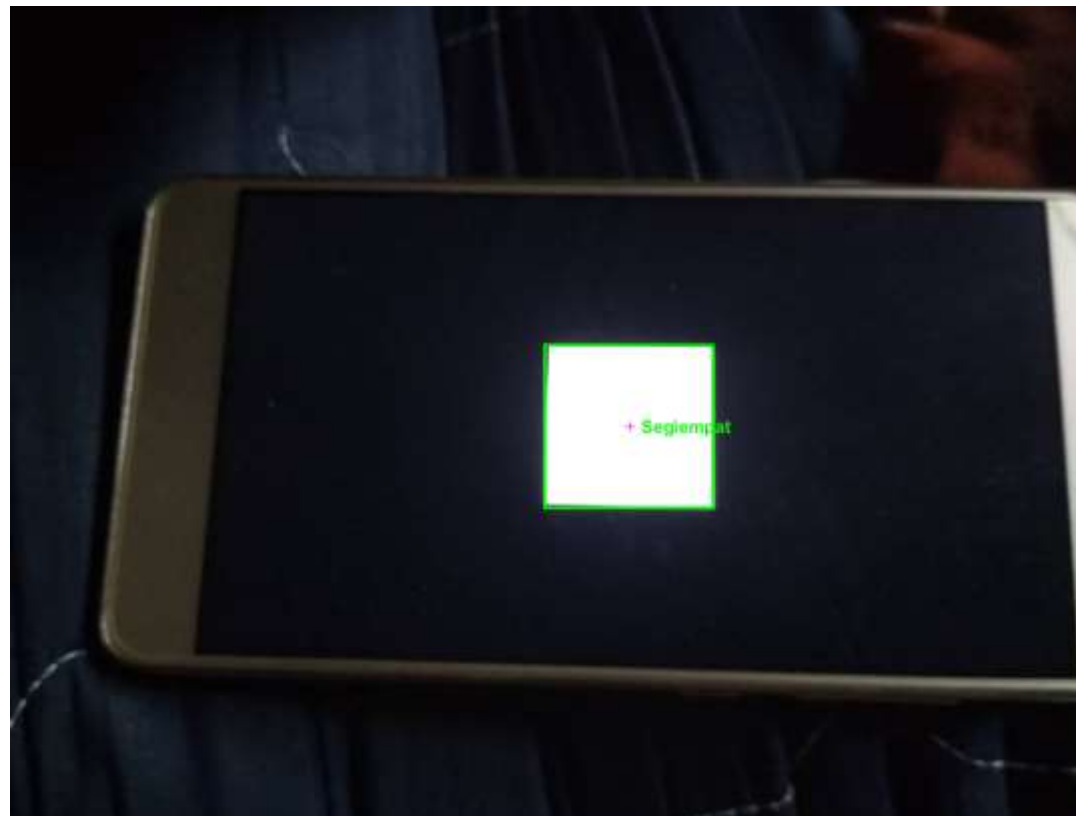
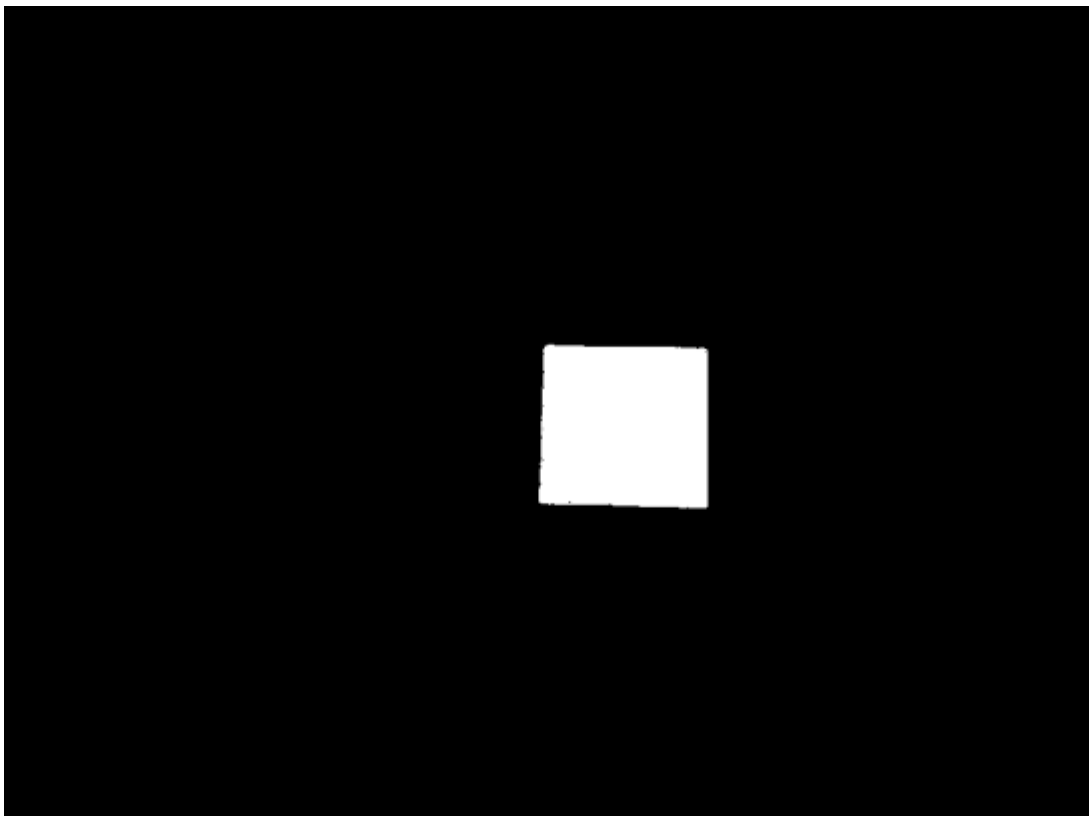


```
8 % Label all the connected components in the image.
9 - label = bwlabel(x, 8);
10
11 % Here we do the image blob analysis.
12 % We get a set of properties for each labeled region.
13 - objek = regionprops(label, 'BoundingBox', 'Centroid');
14 - centroids = cat(2, objek.Centroid);
15
16 % Display the image
17 - imshow(citra)
18
19 - hold on
20 %This is a loop to bound the blue objects in a rectangular box.
21 - for n = 1:length(objek)
22 -     rectangle('Position', objek(n).BoundingBox, 'EdgeColor', 'g', 'LineWidth', 2)
23 -     plot(centroids(:,1), centroids(:,2), '-m+')
24 -     txt = text(centroids(:,1)+15, centroids(:,2), sprintf('%s', pola));
25 -     set(txt, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'green');
26 - end
```



## PENGENALAN POLA

Jaringan Saraf Tiruan : Pengujian







# KONVERSI

## Konversi pola menjadi perintah

Parameter	W	A	S	D
0 (Kiri)	v	v	-	-
3 (Maju)	v	-	-	-
4 (Kanan)	v	-	-	v
5 (Mundur)	-	-	v	-

### Command Window

Elapsed time is 0.437915 seconds.

pola =

'Tidak terdeteksi'

Elapsed time is 0.426574 seconds.

[illegible] $f_x$ 

```

1 function key = cvKey(x)
2     import java.awt.Robot;
3     import java.awt.event.*;
4     key = Robot();
5     % key.setAutoDelay(1000);
6
7     key.keyRelease(KeyEvent.VK_A);
8     key.keyRelease(KeyEvent.VK_S);
9     key.keyRelease(KeyEvent.VK_D);
10    key.keyRelease(KeyEvent.VK_W);
11
12    if x == 4 % Kiri
13        key.keyRelease(KeyEvent.VK_S);
14        key.keyRelease(KeyEvent.VK_D);
15        key.keyPress(KeyEvent.VK_W);
16        key.keyPress(KeyEvent.VK_A);
17    elseif x == 5 % Mundur
18        key.keyRelease(KeyEvent.VK_A);
19        key.keyRelease(KeyEvent.VK_D);
20        key.keyRelease(KeyEvent.VK_W);
21        key.keyPress(KeyEvent.VK_S);
22    elseif x == 0 % Kanan
23        key.keyRelease(KeyEvent.VK_S);
24        key.keyRelease(KeyEvent.VK_A);
25        key.keyPress(KeyEvent.VK_W);
26        key.keyPress(KeyEvent.VK_D);
27    else
28        % Maju
29        key.keyRelease(KeyEvent.VK_S);
30        key.keyRelease(KeyEvent.VK_D);
31        key.keyRelease(KeyEvent.VK_A);
32        key.keyPress(KeyEvent.VK_W);
33    end

```



## KESIMPULAN

Kesimpulan

### Fungsional

Setelah melakukan beberapa percobaan, program ini dapat dikatakan berhasil, seperti yang terlihat dalam simulasi. mobil dapat bergerak maju apabila diberi pola segitiga, dan mundur apabila diberi pola segilima. adapun berbelok ke arah kiri apabila diberi pola kotak segiempat, dan berbelok ke arah kanan apabila diberi pola lingkaran.

### Sistem

**Preprocessing,** Pencahayaan sangat mempengaruhi hasil segmentasi warna.

**JST,** semakin banyak data latih pada proses pelatihan maka semakin besar akurasi yang dihasilkan dan menghasilkan pengenalan pola yang lebih baik lagi.

# Terimakasih

**ARIANTO** ANGGORO  
**AWAL** RAIS SANUBARI  
**DESTI** MINA RAHAYU  
**GEMA** WAHYU SAPUTRA  
**KUNCORO** TRIANDONO MUKTI  
**ILHAM** MAJID RABBANI