

# **Tugas Kelompok**

## **Pengolahan Citra IF-604**

### **Deteksi Plat Nomor Kendaraan**



INSTITUT  
TEKNOLOGI  
HARAPAN  
BANGSA

*Veritas vos liberabit*

Disusun oleh :

1119010 - Neelson William

1119036 - Hans Patrick

1119039 - Pierre Sebastian

1119045 - Rafael Christo Gracia

1119049 - Andreas Stefanus Sibarani

**Departemen Informatika**  
**Institut Teknologi Harapan Bangsa**  
**Jalan Dipatiukur No. 80-84, Bandung**  
**2022**

## 1. Topik

Topik tugas besar yang kami angkat yaitu “Deteksi Plat Nomor Kendaraan”. Deteksi Plat Nomor Kendaraan adalah pendeteksi karakter pada gambar plat nomor kendaraan dengan menggunakan beberapa library. Pendeteksian dilakukan berdasarkan contour yang dideteksi dengan menggunakan library tertentu.

## 2. Batasan Masalah

Dalam Tugas ini, kami akan membatasi masalah yang dikerjakan antara lain :

- Mengimplementasikan proses pengolahan citra pada kehidupan sehari - hari
- Memanfaatkan Library yang ada untuk digunakan menyelesaikan masalah yang ada

## 3. Metode

- Metode Grayscale

Pada Metode ini, gambar Plat nomor kendaraan akan dibuat menjadi grayscale. Metode ini bertujuan untuk menghasilkan informasi yang memudahkan untuk mendeteksi fitur yang ada.

```
gray_image = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
```

✓ 0.4s

Metode ini dapat dijalankan dengan menggunakan library openCV, dan menghasilkan gambar seperti berikut :



- Metode Bilateral Filter

Pada Metode ini Bilateral Filter digunakan untuk menghaluskan gambar dan mengurangi noise. Kami memilih Bilateral Filter untuk smoothing citra dikarenakan menurut :

<https://repositori.usu.ac.id/handle/123456789/18131#:~:text=Bilateral%20Filter%20adalah%20teknik%20untuk,filter%20spasial%20merupakan%20fungsi%20Gaussian.>

Bilateral Filter menghasilkan citra yang lebih baik dibandingkan dengan Gaussian Filter namun Bilateral Filter membutuhkan waktu pemrosesan yang lebih lama dibandingkan dengan Gaussian Filter.

```
# Noise reduction dan Smoothing Image
gray_image = cv.bilateralFilter(gray_image, 11, 17, 17)
```

Bilateral Filter mengandung argument diameter setiap pixel sekitarnya. Pada gambar diatas bernilai 11. Kami mengambilnya dikarenakan menurut kami dengan diameter 11 kami dapat menghasilkan gambar yang lebih baik dibandingkan dengan nilai yang lain. Untuk 2 argument terakhir adalah sigmaColor dan sigmaSpace. Semakin besar nilai dari sigmaColor maka warna akan semakin bercampur satu sama lain. Sedangkan sigmaSpace, semakin besar nilainya maka semakin besar pixel yang bercampur.

Kami memilih nilai kecil dikarenakan kami mengasumsikan gambar yang kami inputkan adalah gambar dengan kualitas baik.



Berikut adalah hasil dari proses smoothing image dengan metode Bilateral Filter.

- Metode Canny EdgeDetection

Kami memilih metode Canny Edge Detection dikarenakan menurut : [https://docs.opencv.org/3.4/da/d5c/tutorial\\_canny\\_detector.html](https://docs.opencv.org/3.4/da/d5c/tutorial_canny_detector.html) , Canny menghasilkan error rate yang kecil

Dengan metode ini kami mendeteksi bagian tepi dari sebuah foto yang sebelumnya sudah diberi filter, sehingga akan terlihat bentuk karakter yang diambil tepiannya.

```
# Edge Detection dengan metode Canny
edged = cv.Canny(smoothing_image, 100, 200) |
```

Argumen yang kami pakai pada pemanggilan function Canny dari library openCV adalah low\_threshold = 100 dan upper\_threshold adalah 200. Kami memilih nilai tersebut dikarenakan nilai tersebut menghasilkan gambar dengan Edged yang optimal.



Berikut adalah hasil pemrosesan edge detection menggunakan metode Canny.

- Contours Detection

Disini kami menggunakan contours untuk mencari tepian dari plat nomor yang akan dideteksi. Nanti akan didapatkan keypoint yang akan digunakan untuk mendapatkan location pojok dari plat nomor. Bentuk output location seperti ini:

```
In [33]: location
Out[33]: array([[467, 139]],
               [[ 61, 141]],
               [[ 64, 252]],
               [[459, 266]]], dtype=int32)
```

Titik kanan bawah = 467,139

Titik kanan atas = 459, 266

Titik kiri bawah = 61, 141

Titik kiri atas = 64, 252

- Masking

Location dari proses contour akan digunakan untuk masking. Dimana Citra di luar dari titik yang telah berada pada variable Location akan diberikan masking.

```
mask = np.zeros(gray_image.shape, np.uint8)
new_image = cv.drawContours(mask, [location], 0,255, -1)
new_image = cv.bitwise_and(image, image, mask=mask)
```

✓ 0.9s

Dalam hal ini bertujuan untuk memudahkan pendeteksian character yang ada pada Citra.



Berikut adalah hasil dari proses masking berdasarkan variabel Location.

- Easyocr

Easyocr merupakan salah satu algoritma untuk mengkonversi gambar menjadi string, yang pada kasus kami akan mengubah gambar nomor polisi dan mendeteksi nomor polisi tersebut menjadi string.

```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

Berikut untuk membaca dengan basis bahasa inggris dan di-read textnya dengan fungsi .readtext(). Yang akan menghasilkan:

```
[([0, 9], [43, 9], [43, 77], [0, 77]), '0', 0.8425513997019927),
([89, 0], [251, 0], [251, 91], [89, 91]), '1019', 0.9999896883964539),
([307, 0], [407, 0], [407, 91], [307, 91]), 'SD', 0.9993249585329087),
([135, 83], [247, 83], [247, 119], [135, 119]),
'05 . 17',
0.7656812718064526)]
```

Dan untuk menyusun dan menghasilkan text yang telah dikonversi dari easyocr, akan menggunakan fungsi pengulangan seperti berikut.

```
index = 0
text = ""
for value in result:
    text += result[index][-2] + " "
    index+=1
font = cv.FONT_HERSHEY_SIMPLEX
print(text)
res = cv.rectangle(image, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
plt.imshow(cv.cvtColor(res, cv.COLOR_BGR2RGB))
```

Menghasilkan text berikut:

```
0 1019 SD 05 . 17
```

Dan berikut untuk menunjukkan deteksi nomor polisi.



#### 4. Pengujian

Dengan Threshold min = 100 dan max = 200 pada Canny Edge Detection

Gambar asal	Hasil masking	Plat yang terdeteksi	Hasil Konversi
			IDB 12 00
			0 1019 SD 05 . 17
			B
			6
			B
			0



Dengan Threshold min = 200 dan max = 300 pada Canny Edge Detection

Gambar asal	Hasil masking	Plat yang terdeteksi	Hasil Konversi
			IDB 12 00
			0 1019 SD 05 . 17
			B
			6
			0
			0

Terdapat sedikit peningkatan pendeteksian pada test pengujian ke 5 dimana yang sebelumnya '0' terrecognize sebagai 'B' menjadi '0'.



#### Analisa Kesalahan :

- Menurut kami tingkat akurasi pengukuran location menurun pada saat ada banyak objek yang terdapat pada gambar.
- Character recognition yang dipakai masih memakai library. Pada saat ada character atau angka yang mirip atau sedikit terhapus pada platnya tidak akan terbaca.
- Recognition juga dipengaruhi oleh kontras warna yang ada pada plat dan lingkungan sekitarnya.
- Hasil recognition dipengaruhi oleh bentuk plat yang kurang menyerupai persegi panjang. Pada beberapa kasus uji ada beberapa asset image dengan plat yang terdeteksi sebagai persegi panjang dengan garis putus - putus pada tahap edge detection.
- Pengukuran akurasi masih belum ditampilkan sebagai persentase.

#### 5. Kesimpulan

Dari hasil pengujian yang telah kami lakukan, dapat disimpulkan bahwa plat nomor kendaraan dapat dideteksi dengan menggunakan pengolahan citra, yang dibantu oleh library - library yang mendukung proses pendeteksian plat nomor kendaraan. Hasil yang didapat dipengaruhi oleh beberapa faktor. Dari hasil yang kami dapatkan tingkat akurasi masih bisa dibilang cukup kecil. Pengubahan parameter threshold yang digunakan sedikit meningkatkan pendeteksian citra yang terdeteksi.

#### 6. Daftar Referensi

<https://pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/>  
[https://www.youtube.com/watch?v=NApYP\\_5wIKY](https://www.youtube.com/watch?v=NApYP_5wIKY)