

Laporan Tugas Besar Pemodelan dan Simulasi

IF-42-GAB



Aldiyan Farhan Nugroho

(1301180344)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2022**

1. Ketentuan Tugas

Pembuatan simulasi kemacetan kendaraan pada jalur lingkaran yang dibuat untuk mengetahui bagaimana pergerakan volume kendaraan didalam jalan tersebut. Dibuatlah simulasi dengan ketentuan :

- Kendaraan hanya dapat bergerak maju, dan demi keamanan tidak boleh melebihi kecepatan tertentu v_{max} .
- Pengendara memperhatikan jarak dengan mobil di depan untuk menghindari tabrakan.
- Kecenderungan pengendara untuk menginjak rem yang menyebabkan mobil melakukan perlambatan mengikuti peluang tertentu.

Dengan aturan yang disebutkan, di implementasikan parameter untuk pemodelan dan simulasi traffic flow dengan parameter tetap sebagai berikut

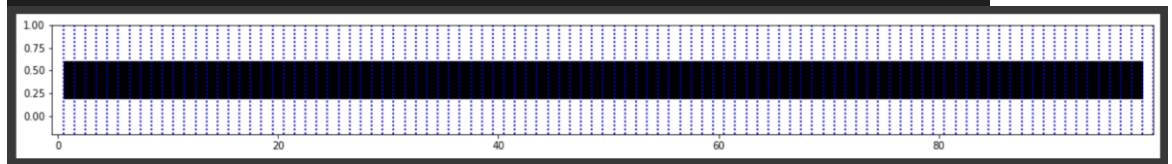
$M = 100$
 $p = 0.3$
 $v_0 = 0$
 $d = 2$
jumlah kendaraan $N = 40$
 $t_{max} = 1000$, $v_{max} = 5$

2. Kodingan

File kodingan dapat di cek langsung pada link [berikut](#)

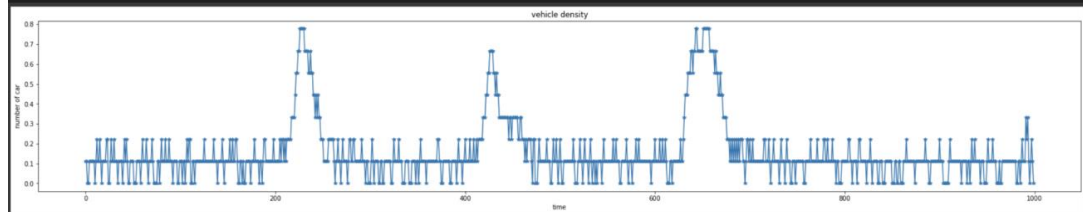
a. Pembuatan Jalur

```
14 def bundar():
15     [ax.axvline(x=k+0.5, color='b', linestyle='dotted') for k in range(0,M+1)]
16     ax.fill_between(np.linspace(0.5,M-1.5,10), 0.2, 0.6, color='black')
17     ax.set_xlim(-0.5,M-0.5)
18     ax.set_ylim(-0.2,1.)
19
20 jenis lintasan = 'bunderan'
21 if jenis lintasan == 'bunderan':
22     bundar()
23     line = [(0,0.4)]
24     [line.append((k,0.8)) for k in range(M)]
25     line.append((M-1,0.4))
26     [line.append((k,0)) for k in range(M)[::-1]]
27     bundar()
```



b. Perhitungan kepadatan mobil

```
1 density = []
2 for i in range(tmax):
3     pos = update(i)
4     density.append(np.count_nonzero((pos < 90) & (pos > 80))/(90-81))
5 plt.figure(figsize=(30,5))
6 plt.plot(range(tmax), density,'*-')
7 plt.title('vehicle density')
8 plt.xlabel('time')
9 plt.ylabel('number of car')
10 plt.show()
```



c. Fuction update posisi

```
34 def update(k):
35     for l in range(N):
36         pos0 = pos[0]
37         if (l == (N-1)):
38             dist = (pos0-pos[l])%len(line)
39         else:
40             dist = (pos[l+1]-pos[l])%len(line)
41
42         if np.random.uniform() < p:
43             vel[l] = max(min(vel[l]+1,vmax,dist-1)-1,0)
44         else:
45             vel[l] = min(vel[l]+1,vmax,dist-1)
46         dispS = (pos[l] + vel[l])%len(line)
47         door = True
48         if (posawal[l] >= 0 and door == True ):
49             posawal[l] = posawal[l] - dispS
50             vel[l]+=1
51         else:
52             door = False
53             pos[l] = dispS
54
55     return pos
```

d.

3. Hasil Simulasi

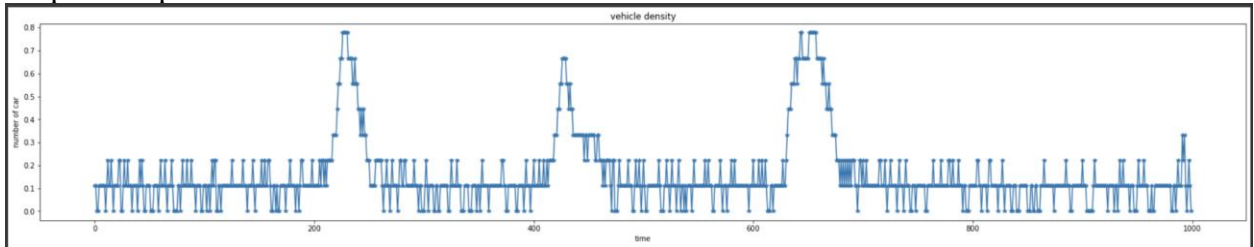
Dari parameter yang digunakan berikut menghasilkan nilai simulasi

```

1 # Parameter
2 M = 100
3 p = 0.3
4 v0 = 0
5 d = 2
6 N = 40
7 tmax = 1000
8 vmax = 5

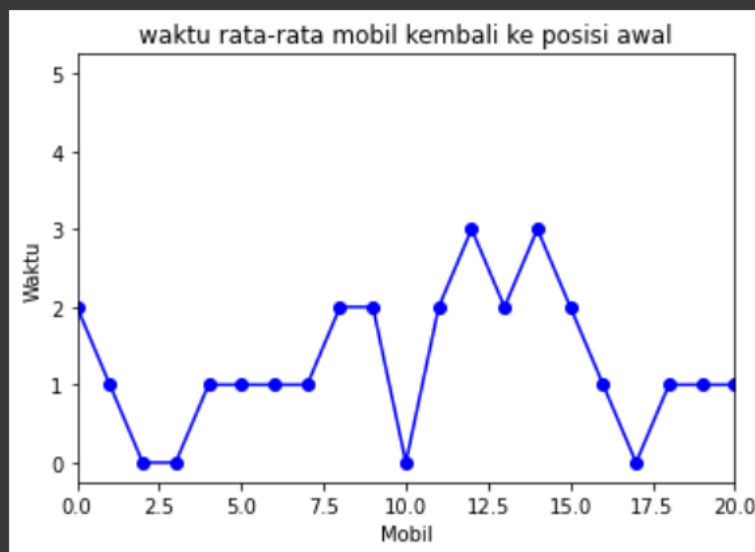
```

- a. Kepadatan per satuan waktu.



Dapat dilihat bahwa terdapat beberapa kepadatan selama simulasi berjalan, nilai 0 berarti lancar sedangkan nilai 1 berarti terhambat atau macet. Dari simulasi yang dilakukan dapat dilihat terjadi beberapa kepadatan yang mendekati angka 1 tepatnya di 0.8 yang menandakan kepadatan yang terjadi namun tidak sampai terhenti.

- b. Waktu rata-rata mobil kembali ke posisi awal.



Jumlah mobil : 40

Total waktu yang dibutuhkan seluruh mobil kembali ke posisi awal : 96.0 seconds

Rata rata waktu mobil kembali ke posisi awal : 2.4 seconds

- c. Hasil simulasi

Video dari simulasi yang telah dilakukan dapat dilihat di link [berikut](#)