PRAKTIKUM FISIKA KOMPUTASI

PENYELESAIAN INTEGRAL

Nama : Wira Satya Baladika

NIM : 1227030037

METODE TRAPEZOID DAN SIMPSON RULE

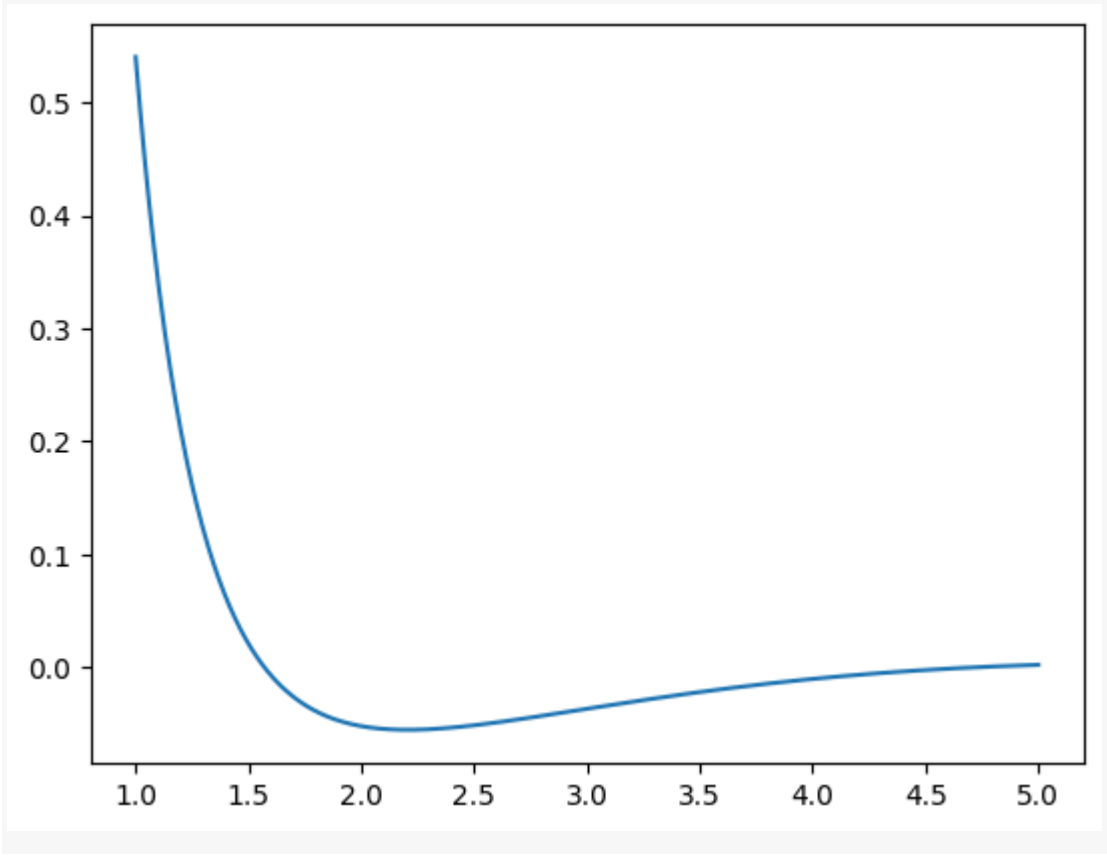
```
import numpy as np
import matplotlib.pyplot as plt
```

```
def func(x):
    return (x**-3)*np.cos(x)
a = 1.0
b = 5.0
```

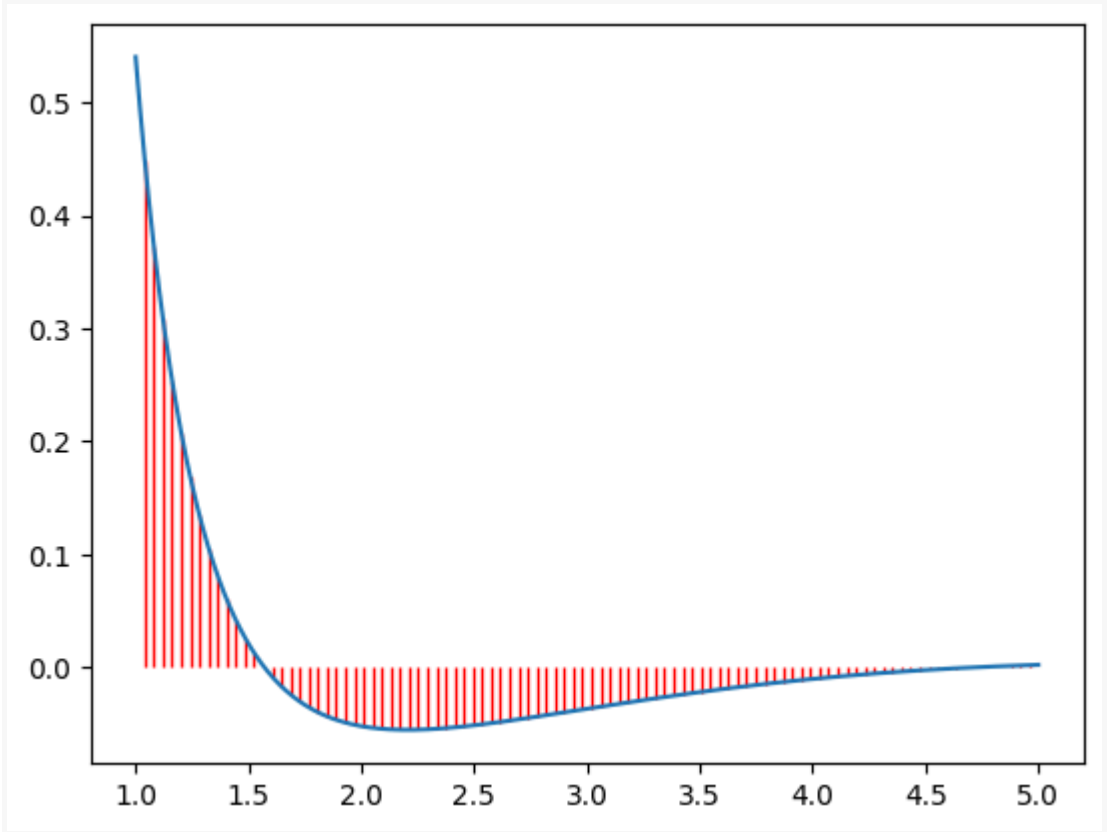
```
# Trapezoid
n = 100
dx = (b-a)/(n-1)
x = np.linspace(a,b,n)

sigma = 0
for i in range (1, n-1):
    sigma += func(x[i])

hasil = 0.5*dx*(func(x[0])+2*sigma+func(x[-1]))

print(hasil)
```

```
0.011902456567430l3
```

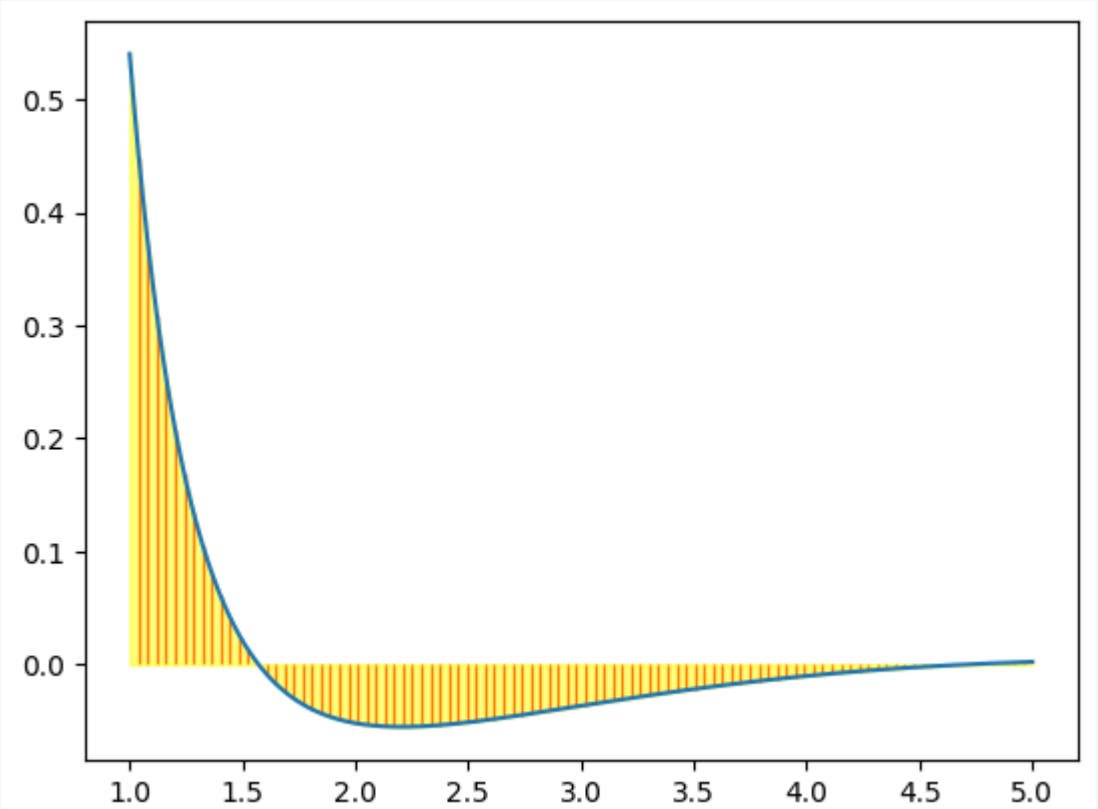
```
xp = np.linspace(a,b,1000)
plt.plot(xp, func(xp))
plt.show()
```




```
xp = np.linspace(a,b,1000)
plt.plot(xp, func(xp))

for i in range (1, n-1):
    plt.bar(x[i], func(x[i]), align = 'edge', width = 0.000001,
edgecolor='red')

plt.show()
```



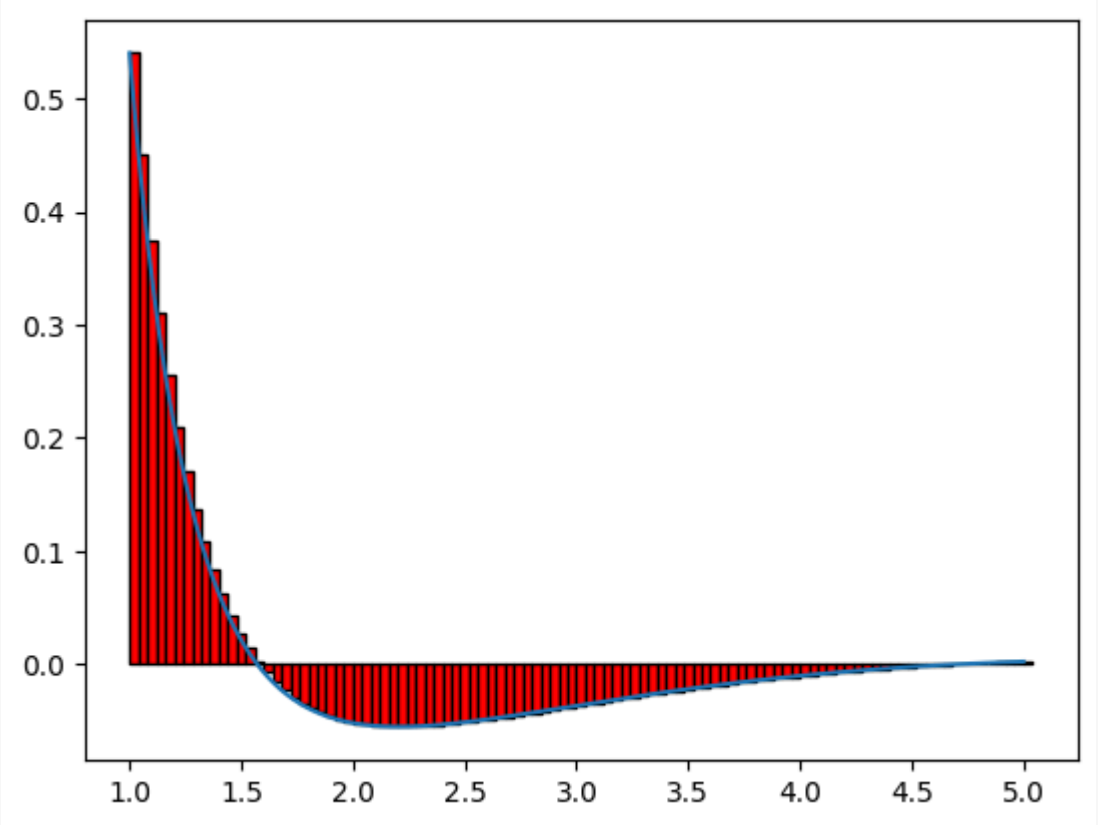

```
xp = np.linspace(a,b,1000)
plt.plot(xp, func(xp))

for i in range (1, n-1):
    plt.bar(x[i], func(x[i]), align = 'edge', width = 0.000001,
edgecolor='red')

plt.fill_between(x,func(x),color= 'yellow', alpha=0.5)

plt.show()
```




```
import numpy as np
import matplotlib.pyplot as plt

def func(x):
    return (x**-3) * np.cos(x)

a = 1.0
b = 5.0
n = 100
```

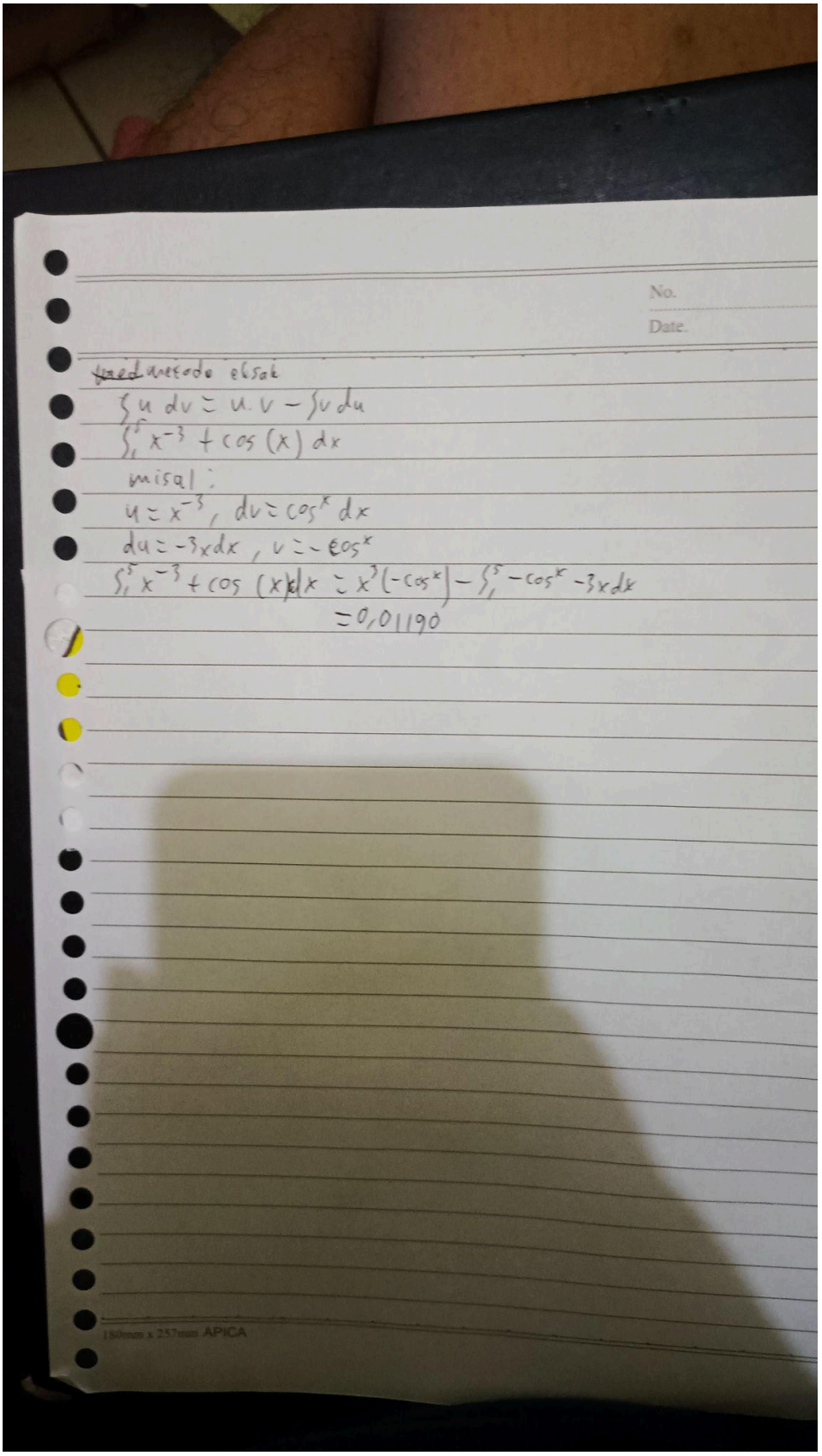
```
#Simpson rule
if n % 2 ==0:
    n += 1

x = np.linspace(a, b, n)
dx = (x[-1] - x[0]) / (n - 1)

hasil = func(x[0]) + func(x[-1])

for i in range(1, n-1, 2):
    hasil += 4 * func(x[i])
```

```
for i in range(2, n-2, 2):
    hasil += 2 * func(x[i])

hasil *= dx / 3

xp = np.linspace(a, b, 1000)
plt.plot(xp, func(xp))

for i in range (n):
    plt.bar(x[i], func(x[i]), align = 'edge', width = dx, color =
'red', edgecolor = 'black')
plt.show()
print(hasil)
```




```
0.011567629904671 75
```

METODE EKSAK



PENJELASAN

Dari hasil percobaan tersebut, penyelesaian soal integral terlihat sama dari menggunakan 3 metode yaitu eksak, trapezoid, dan simpson rule. Hal ini didapatkan sebuah kode program yang dirancang menyesuaikan rumus untuk penyelesaian soal integral yang ada pada diatas.