Tugas 4 Praktikum Fisika Komputasi

Integral Numerik

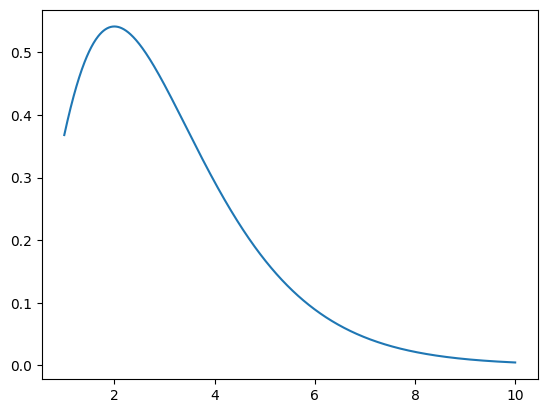
Moch. Alldho Candra Ramadhan

(1227030020)

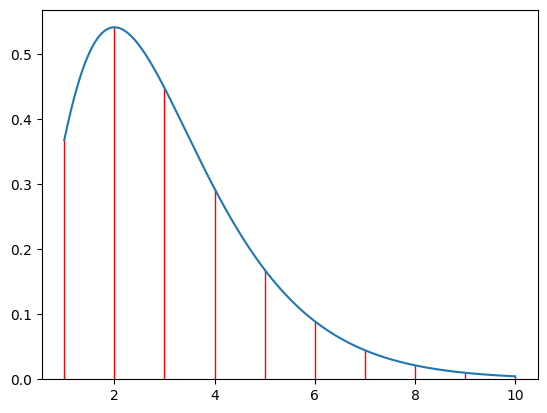
Lampiran

|  |
| --- |
| # Mengimport Library  import numpy as np  import matplotlib.pyplot as plt  # Integral  def func(x):                    # Nama Fungsi      return (x\*\*2) \* np.exp(-x)  # fungsi yang akan diintegralkan  a = 1.0                         # Batas bawah  b = 10.0                        # Batas atas  # Metode Trapezoid  n = 10                          # Jumlah grid  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)  1.8025164595168974 |

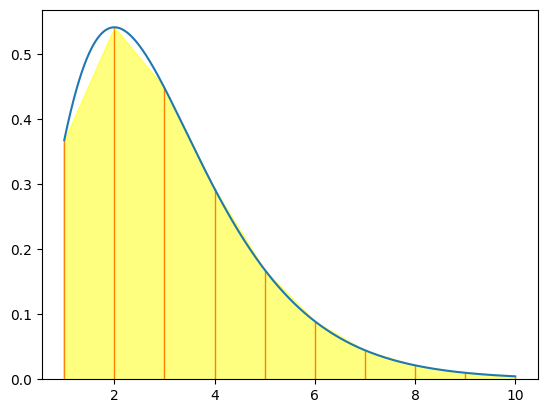
|  |
| --- |
| xp =np.linspace(a, b, 1000)  plt.plot(xp, func(xp))  plt.show() |



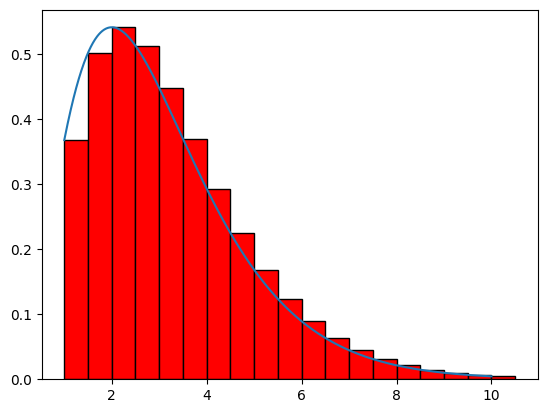
|  |
| --- |
| xp = np.linspace(a, b, 1000)  plt.plot(xp, func(xp))  for i in range(n):      plt.bar(x[i], func(x[i]), align='edge', width=0.000001, edgecolor='red')  plt.show() |



|  |
| --- |
| xp = np.linspace(a, b, 1000)  plt.plot(xp, func(xp))  for i in range(n):      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() |

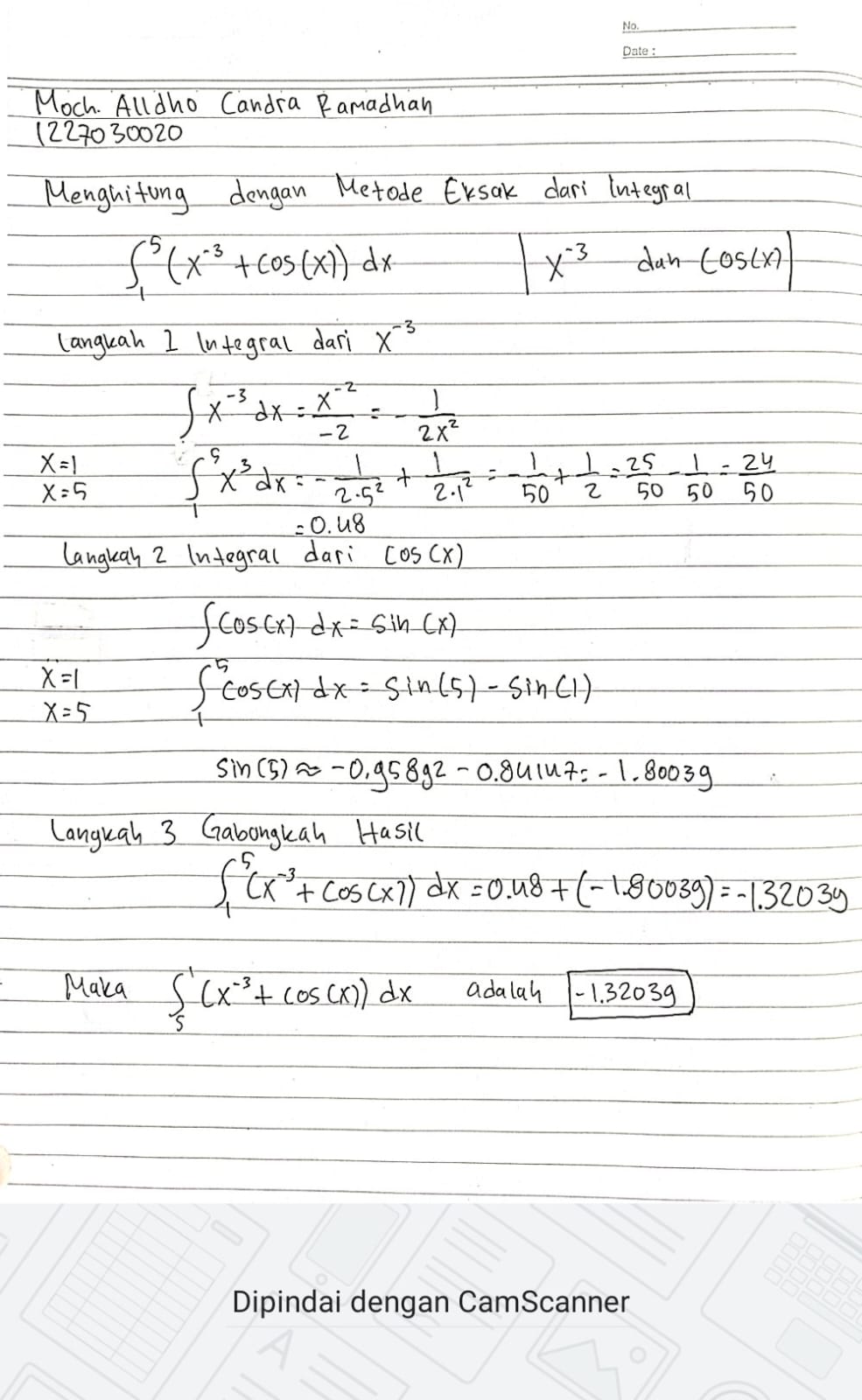


|  |
| --- |
| import matplotlib.pyplot as plt  import numpy as np  # Fungsi yang akan diintegralkan  def func(x):      return (x\*\*2) \* np.exp(-x)  # Fixed function definition  # Batas integrasi  a = 1.0  # Batas bawah  b = 10.0 # Batas atas  n = 19  # Jumlah grid, harus ganjil untuk metode Simpson  # Pastikan n ganjil untuk metode Simpson  if n % 2 == 0:      n += 1  # Jika n genap, tambah 1 agar menjadi ganjil  x = np.linspace(a, b, n)  # Membuat grid untuk x  dx = (x[-1] - x[0]) / (n - 1)  # Lebar tiap grid  # Menghitung Integral menggunakan metode Simpson  hasil = func(x[0]) + func(x[-1])  # Tambah f(a) dan f(b)  # Untuk indeks ganjil  for i in range(1, n-1, 2):      hasil += 4 \* func(x[i])  # Untuk indeks genap  for i in range(2, n-1, 2):      hasil += 2 \* func(x[i])  hasil = dx / 3 \* hasil  # Faktor dx/3  # Visualisasi grafik dan bar  xp = np.linspace(a, b, 1000)  # Grid untuk plot halus  plt.plot(xp, func(xp))  # Plot kurva fungsi  # Plot bar untuk grid Simpson  for i in range(n):      plt.bar(x[i], func(x[i]), align='edge', width=dx, color='red', edgecolor='black')  plt.show()  # Cetak |



**Soal no 1**

a). Metode Eksak

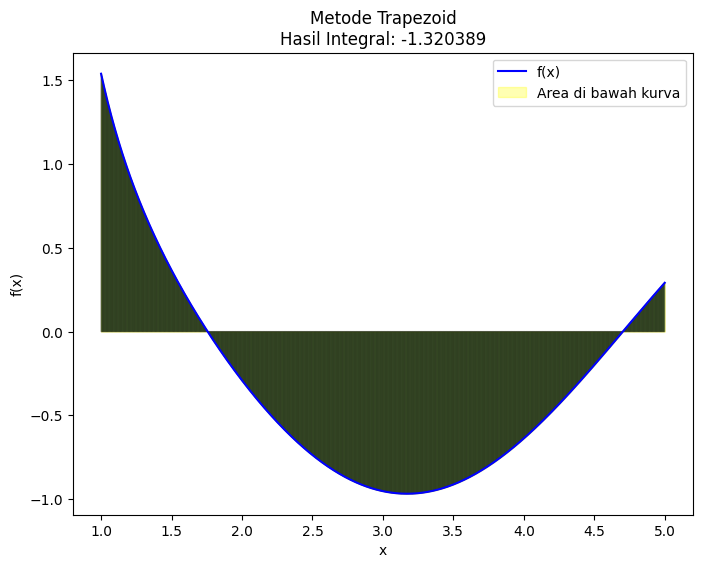


b). Metode Trapezoid

Kode pemrograman

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Definisikan fungsi  def f(x):      return x\*\*(-3) + np.cos(x)  # Batas integral dan jumlah partisi  a = 1  b = 5  n = 1000  # Jumlah partisi  # Buat array x dan y  x = np.linspace(a, b, n+1)  y = f(x)  # Implementasi metode trapezoid  h = (b - a) / n  integral\_trapezoid = (h / 2) \* (y[0] + 2 \* np.sum(y[1:n]) + y[n])  # Plot untuk metode Trapezoid  x\_fine = np.linspace(a, b, 1000)  y\_fine = f(x\_fine)  plt.figure(figsize=(8, 6))  plt.plot(x\_fine, y\_fine, label="f(x)", color="blue")  plt.fill\_between(x, 0, y, color='yellow', alpha=0.3, label='Area di bawah kurva')  # Plot area untuk metode Trapezoid  for i in range(n):      plt.fill([x[i], x[i], x[i+1], x[i+1]], [0, y[i], y[i+1], 0], color='green', edgecolor='black', alpha=0.3)  plt.title(f"Metode Trapezoid\nHasil Integral: {integral\_trapezoid:.6f}")  plt.xlabel('x')  plt.ylabel('f(x)')  plt.legend()  # Tampilkan grafik  plt.show() |

Grafik Metode Trapezoid

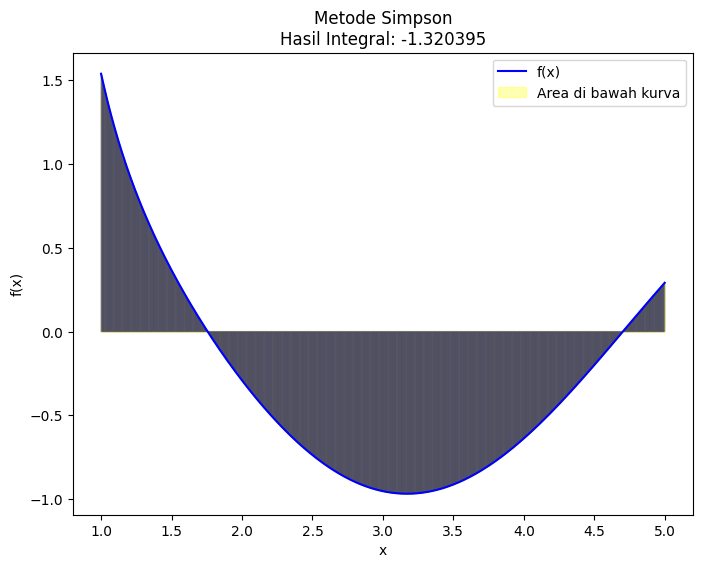


c). Metode Simpson

Kode Pemrograman

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  from scipy.integrate import simps  # Definisikan fungsi  def f(x):      return x\*\*(-3) + np.cos(x)  # Batas integral dan jumlah partisi  a = 1  b = 5  n = 1000  # Pastikan n genap untuk metode Simpson  # Buat array x dan y  x = np.linspace(a, b, n+1)  y = f(x)  # Implementasi metode Simpson  integral\_simpson = simps(y, x)  # Plot untuk metode Simpson  x\_fine = np.linspace(a, b, 1000)  y\_fine = f(x\_fine)  plt.figure(figsize=(8, 6))  plt.plot(x\_fine, y\_fine, label="f(x)", color="blue")  plt.fill\_between(x, 0, y, color='yellow', alpha=0.3, label='Area di bawah kurva')  # Plot area untuk metode Simpson  for i in range(0, n, 2):      plt.fill([x[i], x[i], x[i+2], x[i+2]], [0, y[i], y[i+2], 0], color='blue', edgecolor='black', alpha=0.3)  plt.title(f"Metode Simpson\nHasil Integral: {integral\_simpson:.6f}")  plt.xlabel('x')  plt.ylabel('f(x)')  plt.legend()  # Tampilkan grafik  plt.show() |

Grafik Metode Simpson



**Soal no 2**

Hasil dari setiap metode dimulai dari metode eksak, metode trapezoid dan metode simpson ialah sama dengan metode perhitungan eksak manual ada perhitungan yang di bulatkan, sedangkan untuk metode trapezoid ini mengaproksimasi kurva dengan yang di pakai serangkaian trapezium dalam penggunaanya, lalu untuk metode simpson ialah menggunakan pendekatan parabola (kuadratik) untuk mendekati fungsi. Maka dengan ketiga metode tersebut dapat di ketahui hasil integralnya sama yaitu -1,320395.

**Soal no 3**

Metode eksak paling efektif digunakan karena fungsi dapat diintegralkan secara analitik dan manual, karena memberikan hasil presisi tanpa kode program serta aproksimasi. Namun, untuk fungsi yang lebih kompleks atau tidak dapat diselesaikan secara manual, metode numerik trapezoid dan Simpson lebih praktis. Metode trapezoid lebih sederhana dan cepat, tetapi kurang akurat karena hanya menghampiri kurva dengan garis lurus, sementara metode Simpson lebih akurat dengan galat yang lebih kecil karena menggunakan parabola untuk mendekati kurva. Jadi untuk situasi akurasi lebih penting, metode Simpson lebih efektif, sedangkan metode trapezoid lebih cocok untuk cepat.