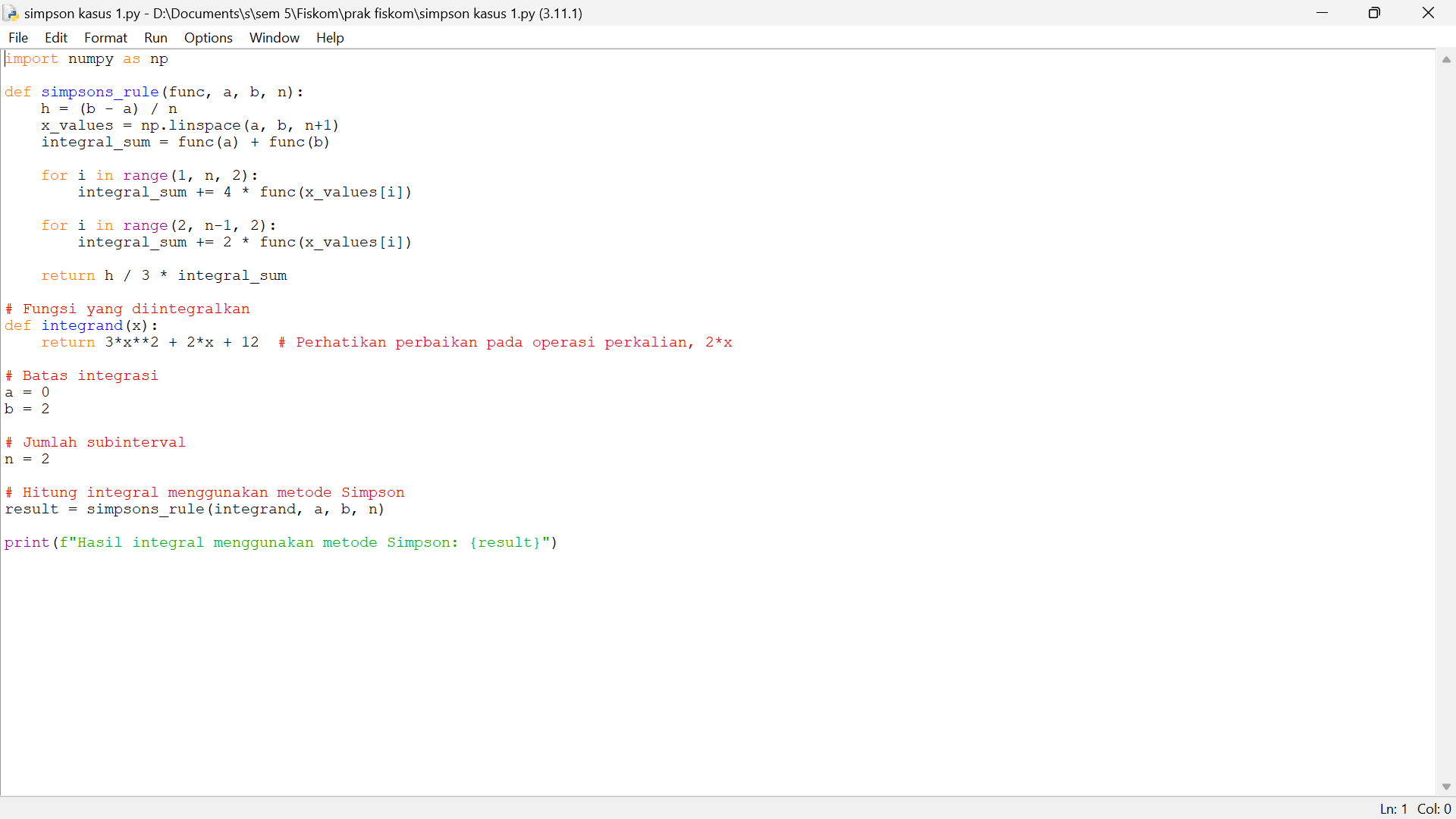
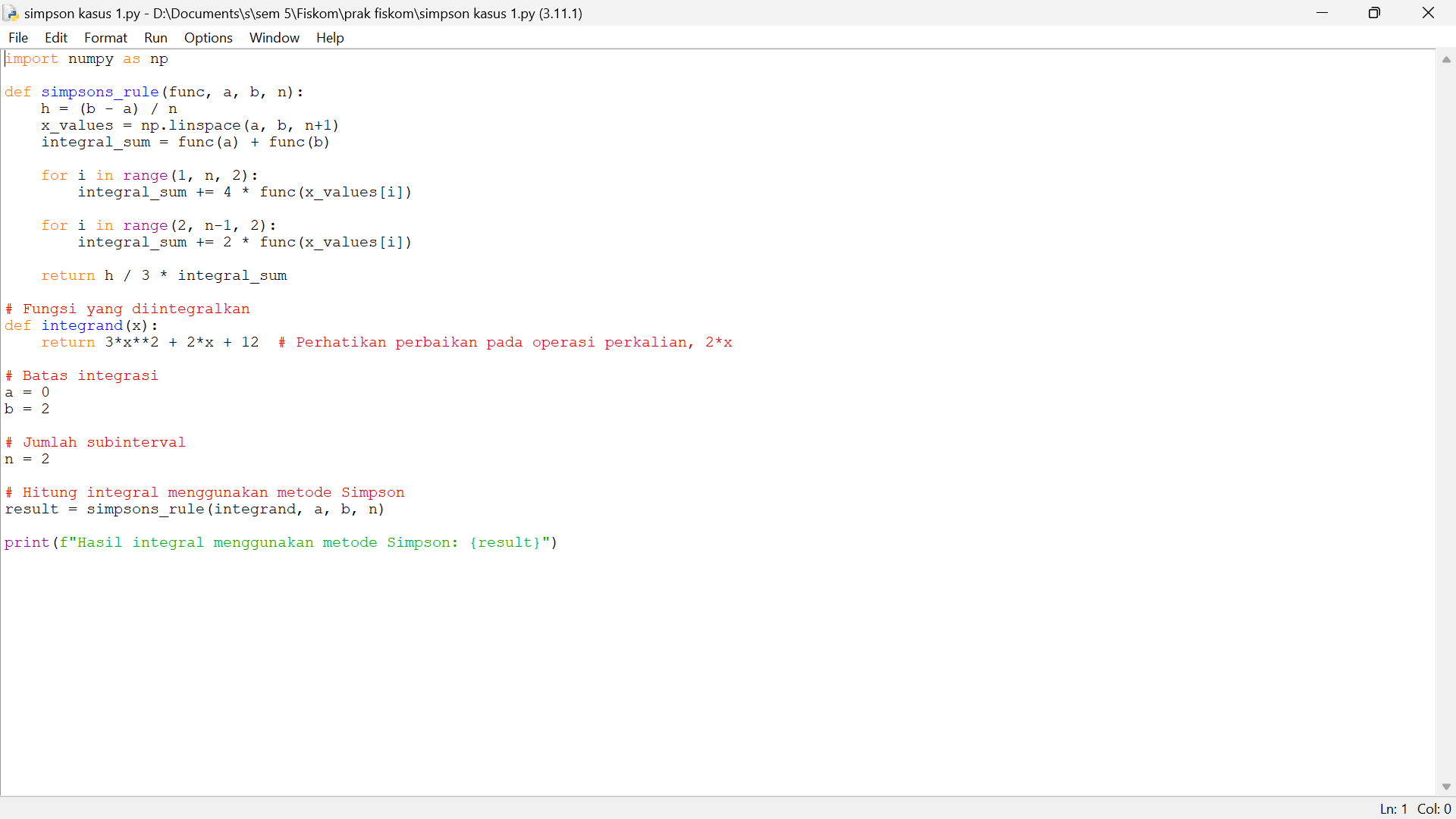
Penjelasan Kasus 1

Kode tersebut adalah implementasi dari metode Simpson untuk menghitung integral numerik dari sebuah fungsi dalam interval tertentu. Di sini adalah penjelasan dari setiap bagian kode tersebut:



* `import numpy as np`: Mengimpor modul NumPy yang digunakan untuk operasi numerik dan array.



* `def simpsons\_rule(func, a, b, n):`: Mendefinisikan fungsi `simpsons\_rule` yang akan menghitung integral menggunakan metode Simpson. Fungsi ini menerima empat argumen:

- `func`: Fungsi yang ingin diintegrasikan.

- `a` dan `b`: Batas bawah dan atas dari interval integrasi.

- `n`: Jumlah subinterval yang digunakan dalam metode Simpson.

* Dalam fungsi `simpsons\_rule`:

- `h = (b - a) / n`: Menghitung lebar setiap subinterval.

- `x\_values = np.linspace(a, b, n+1)`: Membuat array nilai \( x \) untuk setiap titik pada interval \( [a, b] \) dengan jumlah \( n+1 \) titik secara merata.

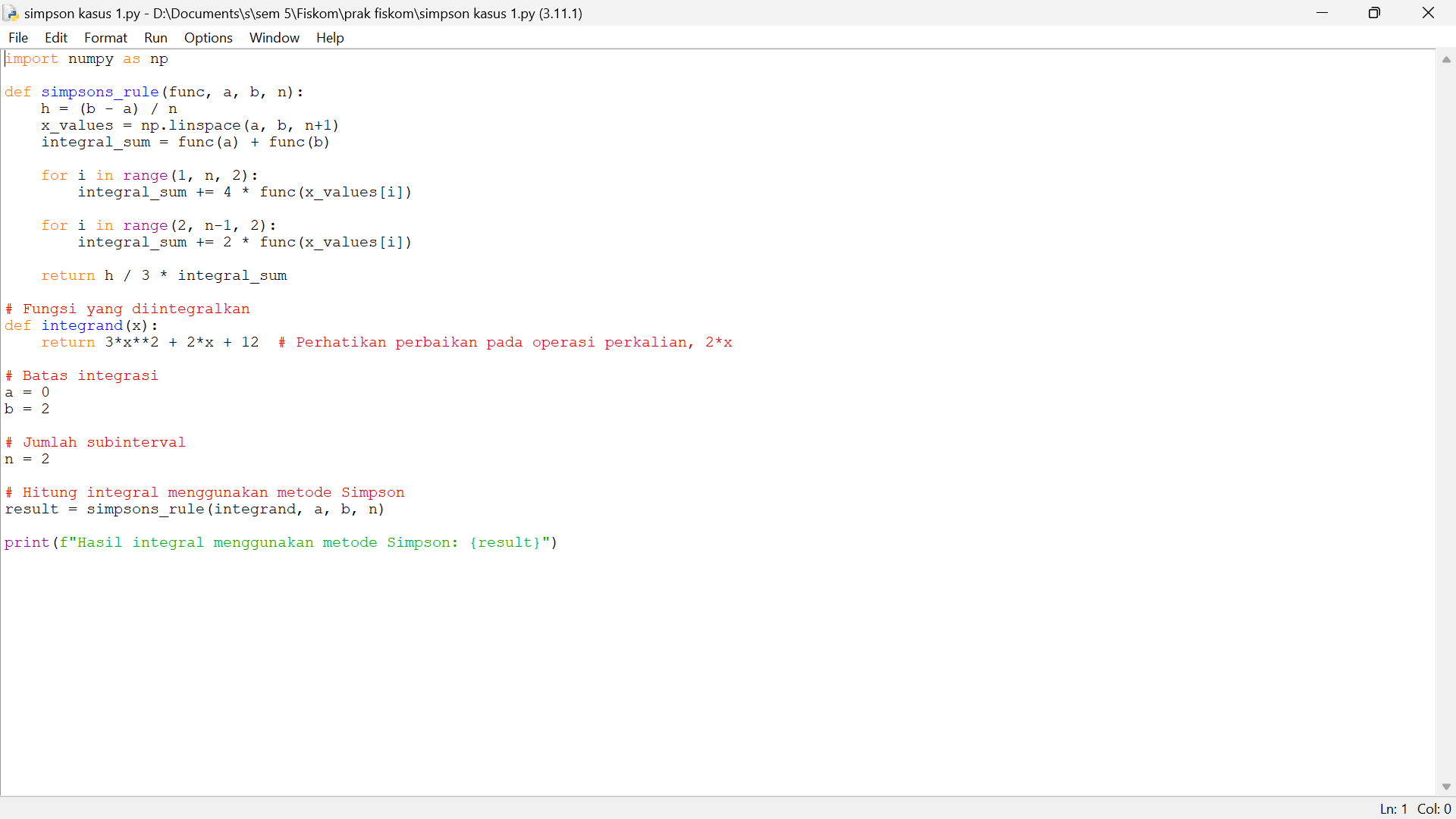
- `integral\_sum = func(a) + func(b)`: Menambahkan nilai fungsi pada batas bawah dan atas ke dalam hasil integral.

* Terdapat dua loop `for`:

- Loop pertama (`for i in range(1, n, 2):`) menghitung kontribusi dari titik-titik dengan indeks ganjil dalam metode Simpson (menggunakan faktor 4).

- Loop kedua (`for i in range(2, n-1, 2):`) menghitung kontribusi dari titik-titik dengan indeks genap dalam metode Simpson (menggunakan faktor 2).

* Hasil integral dikembalikan dengan rumus metode Simpson: `return h / 3 \* integral\_sum`.



* `def integrand(x):`: Ini adalah definisi fungsi `integrand(x)` yang merepresentasikan fungsi yang akan diintegralkan. Di sini, fungsi tersebut adalah \(3x^2 + 2x + 12\).
* `a = 0` dan `b = 2`: Merupakan batas bawah dan atas dari interval yang ingin diintegrasikan.
* `n = 2`: Jumlah subinterval yang akan digunakan dalam metode Simpson.
* `result = simpsons\_rule(integrand, a, b, n)`: Memanggil fungsi `simpsons\_rule` dengan argumen yang telah ditentukan sebelumnya untuk menghitung integral fungsi `integrand` dari \(0\) hingga \(2\) menggunakan metode Simpson dengan \(2\) subinterval.
* `print(f"Hasil integral menggunakan metode Simpson: {result}")`: Mencetak hasil integral yang telah dihitung menggunakan metode Simpson.

**Penjelasan Kasus 2**

Tentu, berikut adalah penjelasan mengenai kode yang telah diberikan:



* `import numpy as np`: Ini adalah pernyataan untuk mengimpor modul NumPy ke dalam program Python. NumPy adalah pustaka yang digunakan untuk melakukan operasi numerik, seperti array, operasi matematika, dan lainnya. Penggunaan `as np` membuat kita dapat menggunakan alias `np` untuk memanggil fungsi-fungsi dari NumPy.



* `def simpsons\_rule(func, a, b, n):`: Ini adalah definisi fungsi `simpsons\_rule` yang mengimplementasikan metode Simpson untuk menghitung integral numerik dari suatu fungsi. Fungsi ini menerima argumen:

- `func`: Fungsi yang ingin diintegralkan.

- `a` dan `b`: Batas bawah dan atas dari interval integrasi.

- `n`: Jumlah subinterval yang digunakan dalam metode Simpson.

* Dalam fungsi `simpsons\_rule`:

- `h = (b - a) / n`: Menghitung lebar setiap subinterval.

- `x\_values = np.linspace(a, b, n+1)`: Membuat array nilai x untuk setiap titik pada interval \( [a, b] \) dengan jumlah `n+1` titik secara merata.

- `integral\_sum = func(a) + func(b)`: Menambahkan nilai fungsi pada batas bawah dan atas ke dalam hasil integral.

* Kemudian terdapat dua loop `for`:

- Loop pertama (`for i in range(1, n, 2):`) menghitung kontribusi dari titik-titik dengan indeks ganjil dalam metode Simpson (menggunakan faktor 4).

- Loop kedua (`for i in range(2, n-1, 2):`) menghitung kontribusi dari titik-titik dengan indeks genap dalam metode Simpson (menggunakan faktor 2).

* Hasil integral dikembalikan dengan rumus metode Simpson: `return h / 3 \* integral\_sum`.



* `def integrand(x):`: Ini adalah definisi fungsi `integrand(x)` yang merepresentasikan fungsi yang akan diintegralkan. Di sini, fungsi tersebut adalah \(\frac{{x^2}}{3} + 2x + 12\).
* `a = 0` dan `b = 2`: Ini adalah batas bawah dan atas dari interval yang ingin diintegrasikan.
* `n = 2`: Ini adalah jumlah subinterval yang akan digunakan dalam metode Simpson.
* `result = simpsons\_rule(integrand, a, b, n)`: Memanggil fungsi `simpsons\_rule` dengan argumen yang telah ditentukan sebelumnya untuk menghitung integral fungsi `integrand` dari \(0\) hingga \(2\) menggunakan metode Simpson dengan \(2\) subinterval.
* `print(f"Hasil integral menggunakan metode Simpson: {result}")`: Mencetak hasil integral yang telah dihitung menggunakan metode Simpson.