

Nurunnisa Fathanah Dz. S. B.
D121211002
Tugas 3 - Metode Komputasi Numerik
Chapter 2 - Integration

Nurunnisa Fathanah Dz. S. B.

D121211002

Metode Komputasi Numerik Kelas A

Buat Grid dan Points

1. Buat sebuah fungsi yang masukannya adalah a, b, dan n. a dan b adalah batas integrasi, n adalah jumlah grid yang akan dibuat. Keluaran fungsi ini adalah h (ukuran dari grid) dan titik x_i ($i = 0$ sampai $n + 1$, $x_{n+1} = b$)

```
def namafungsi(a,b,n):
```

```
....
```

```
return h,xi
```

```
[1]: def gridPoints(a,b,n):  
      h = (b-a)/(n)  
      xi=[]  
      for i in range(n+1):  
          point = a + i*h  
          xi.append(point)  
          #print(f'Ukuran dari grid adalah: {h}')  
          #print(f'Titik Xi adalah: {xi}')  
      return h, xi
```

```
[2]: #tes no.1  
a = 0  
b = 1  
n = 10  
h, xi = gridPoints(a,b,n)  
print("Ukuran dari grid adalah: ",h)  
print("Titik Xi adalah : ",xi)  
  
Ukuran dari grid adalah: 0.1  
Titik Xi adalah : [0.0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0.6000000000000001, 0.7000000000000001, 0.8, 0.9, 1.0]
```

Dibuat fungsi bernama gridPoints(a, b, n), yang memiliki tiga parameter: a, b, dan n. Fungsi ini digunakan untuk menghasilkan titik-titik grid antara a dan b dengan jumlah titik sebanyak n+1 yang sama terpisah.

Berikut penjelasan kode tersebut:

1. Menghitung ukuran langkah h antara dua titik berturut-turut pada grid menggunakan rumus $(b - a) / n$.
2. Membuat sebuah array kosong xi untuk menyimpan titik-titik grid.
3. Melakukan iterasi i dari 0 hingga n.
4. Menghitung titik grid ke-i menggunakan rumus $a + i * h$.
5. Menambahkan titik yang dihitung ke dalam array xi.
6. Mengembalikan ukuran langkah h dan array titik-titik grid xi.

Pada kode pengujian, nilai a = 0, b = 1, dan n = 10 digunakan untuk menghasilkan titik-titik grid. Nilai yang dikembalikan yaitu h dan xi kemudian dicetak.

```
[3]: h
```

```
[3]: 0.1
```

```
[4]: xi
```

```
[4]: [0.0,  
      0.1,  
      0.2,  
      0.30000000000000004,  
      0.4,  
      0.5,  
      0.6000000000000001,  
      0.7000000000000001,  
      0.8,  
      0.9,  
      1.0]
```

Keluaran dari kode pengujian tersebut akan menjadi:

Ukuran dari grid adalah: 0.1

Titik Xi adalah : [0.0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0.6000000000000001, 0.7000000000000001, 0.8, 0.9, 1.0]

Variabel h akan memiliki nilai 0.1, dan variabel xi akan berupa array yang berisi titik-titik grid [0.0, 0.1, 0.2, 0.30000000000000004, 0.4, 0.5, 0.6000000000000001, 0.7000000000000001, 0.8, 0.9, 1.0]

Hitung Segiempat Riemanns

2. Buat sebuah fungsi yang masukannya adalah x dan $f(x)$. Keluaran fungsi ini adalah y (yaitu panjang/tinggi segiempat Riemann di titik x), yaitu y_i ($i = 0$ sampai $n + 1$)

```
def namafungsi(y,fx):
```

```
....
```

```
return y
```

```
In [5]: def tinggiRiemanns(x, f):
        n = len(x) - 1
        yi = []
        for i in range(n):
            y = f(x[i])
            yi.append(y)
        return yi
```

Fungsi `tinggiRiemanns(x, f)` yang memiliki dua parameter: x dan f . Fungsi ini menghitung nilai fungsi f pada titik-titik x yang diberikan dan mengembalikan array nilai tersebut.

Berikut ini adalah penjelasan langkah demi langkah dari kode tersebut:

1. Menghitung jumlah subinterval n dengan mengambil panjang array x dikurangi 1.
2. Membuat sebuah array kosong yi untuk menyimpan nilai-nilai fungsi f pada setiap titik x .
3. Melakukan iterasi dari 0 hingga $n-1$.
4. Pada setiap iterasi, mengambil nilai x pada indeks i dan menghitung nilai fungsi y dengan memanggil fungsi f dengan argumen $x[i]$.
5. Menambahkan nilai y ke dalam array yi .
6. Mengembalikan array yi yang berisi nilai-nilai fungsi f pada setiap titik x .

Kode tersebut tidak menyertakan implementasi fungsi f yang digunakan untuk menghitung nilai fungsi pada setiap titik x .

```
[6]: #tes no.2

#fungsi
def f(x):
    return x**2

a = 0      #batas bawah
b = 1      #batas atas
n = 10     #jumlah grid

#hitung lebar dan titik grid
h, xi = gridPoints(a, b, n)

#hitung tinggi dari segi empat Riemanns
yi = tinggiRiemanns(xi, f)

print("Panjang/tinggi segiempat Riemann di titik Xi yaitu: ", yi)

Panjang/tinggi segiempat Riemann di titik Xi yaitu: [0.0, 0.010000000000000002, 0.04000000000000001, 0.09000000000000002, 0.16000000000000003, 0.25, 0.36000000000000001, 0.49000000000000001, 0.6400000000000001, 0.81]
```

Kemudian dilakukan pengujian dengan $a = 0$, $b = 1$, $n = 10$. Dan menyimpan nilai kembalian (*return value*) fungsi `grid` ke dalam h dan xi . Lalu menyimpan nilai kembalian (*return value*) `tinggiRiemanns` ke dalam variabel yi . Terakhir mencetak panjang/tinggi segiempat riemanns.

Hitung Riemanns Sum

3. Menggunakan kedua fungsi yang sudah dibuat, hitung Riemanns Sum untuk $\int_0^1 x^2$

```
[8]: def riemann_sum(f, a, b, n):  
      """  
      Menghitung Riemann sum dari fungsi f(x) pada interval [a,b]  
      menggunakan n subinterval.  
      """  
      #menghitung lebar dan titik grid  
      h, xi = gridPoints(a, b, n)  
      #menghitung tinggi/panjang grid  
      yi = tinggiRiemanns(xi, f)  
  
      #menghitung Riemann sum  
      riemann_sum = sum(h*yi[i] for i in range (n))  
  
      return riemann_sum
```

```
[9]: #tes no.3  
      #fungsi  
      def f(x):  
          return x**2  
  
      #menentukan batas integrasi  
      a = 0  
      b = 1  
  
      #menentukan jumlah subinterval  
      n = 10  
  
      # menghitung Riemann sum  
      riemann_sum_val = riemann_sum(f, a, b, n)  
  
      #menampilkan hasil  
      print("Riemann sum:", riemann_sum_val)
```

Riemann sum: 0.2850000000000001

Fungsi `riemann_sum(f, a, b, n)` yang diberikan menghitung Riemann sum dari fungsi $f(x)$ pada interval $[a, b]$ menggunakan n subinterval. Berikut ini adalah penjelasan langkah demi langkah dari kode tersebut:

1. Pertama-tama, fungsi ini membagi interval $[a, b]$ menjadi n subinterval dengan lebar yang sama. Hal ini dilakukan dengan menggunakan fungsi `gridPoints(a, b, n)` yang menghasilkan lebar (h) dari setiap subinterval dan titik-titik grid (xi) yang menandai batas-batas subinterval.
2. Selanjutnya, fungsi `riemann_sum` menghitung tinggi atau panjang setiap subinterval dengan menggunakan fungsi `tinggiRiemanns(xi, f)`. Fungsi ini memasukkan setiap titik grid xi ke dalam fungsi $f(x)$ untuk mendapatkan tinggi/panjang setiap subinterval (yi).
3. Terakhir, Riemann sum dihitung dengan menjumlahkan perkalian antara lebar setiap subinterval (h) dengan tinggi setiap subinterval (yi). Ini dilakukan dengan menggunakan perulangan `for` untuk mengakses setiap indeks i dalam rentang $(0, n)$. Hasil Riemann sum kemudian dikembalikan sebagai output dari fungsi.
4. Dengan menggunakan fungsi `riemann_sum`, kita dapat mengestimasi luas area di bawah kurva fungsi $f(x)$ pada interval $[a, b]$ dengan menggunakan metode Riemann sum. Semakin banyak subinterval yang digunakan (n yang lebih besar), estimasi luas area akan semakin akurat.

```
[9]: #tes no.3
      #fungsi
      def f(x):
          return x**2

      #menentukan batas integrasi
      a = 0
      b = 1

      #menentukan jumlah subinterval
      n = 10

      # menghitung Riemann sum
      riemann_sum_val = riemann_sum(f, a, b, n)

      #menampilkan hasil
      print("Riemann sum:", riemann_sum_val)

      Riemann sum: 0.2850000000000001
```

```
[10]: riemann_sum_val
```

```
[10]: 0.2850000000000001
```

Kemudian untuk pengujiannya, kita menetapkan fungsinya, menentukan batas integrasi, dan jumlah subinterval-nya atau ingin dibagi menjadi berapa grid. Setelah ini kita menyimpan nilai kembalian (*return value*) dari fungsi `riemann_sum` ke dalam `riemann_sum_val`. Terakhir, untuk melihat hasilnya, maka cetak nilai `riemann_sum_val`. Untuk soal ini, hasil riemann sum-nya adalah 0.285