

SK5002 ALGORITMA DAN RANCANGAN PERANGKAT LUNAK

Ujian Tengah Semester

Mohammad Rizka Fadhli

NIM: 20921004

14 October 2021

Contents

1	PENDAHULUAN	4
1.1	Bahasa Pemrograman yang Dipakai	4
1.2	<i>Libraries</i> R yang Digunakan	4
1.3	Program	4
1.4	Pembulatan	4
1.5	Program untuk Menggambar Fungsi	4
2	SOAL 1	6
	Algoritma Komputasi Numerik	6
2.1	Soal Utama	6
2.1.1	Sub Soal I	6
2.1.2	Jawaban Sub Soal I	6

List of Figures

1	Grafik $f(x,y)$	6
2	Area Luas	7
3	sumber: bragitoff.com	8

1 PENDAHULUAN

1.1 Bahasa Pemrograman yang Dipakai

Bahasa pemrograman yang digunakan pada tugas ini adalah **R** versi 4.1.1. Format tugas ini ditulis menggunakan *LaTeX R Markdown* di *software R Studio*.

1.2 *Libraries* R yang Digunakan

Berikut adalah beberapa *libraries* yang digunakan dalam mengerjakan dan menuliskan tugas ini:

1. `dplyr`: untuk *data carpentry*.
2. `ggplot2`: sebagai visualisasi data (grafik).

1.3 Program

Program yang digunakan untuk menjawab soal akan di kirimkan sebagai lampiran dan ditunjukkan dalam format *code markdown*. Agar bisa dieksekusi dengan baik, pastikan *libraries* yang terlibat sudah ter-*install* terlebih dahulu.

1.4 Pembulatan

Seluruh jawaban numerik akan ditampilkan menggunakan **delapan angka berarti**.

1.5 Program untuk Menggambar Fungsi

Berikut adalah program yang saya buat untuk menggambar fungsi:

```
gambar_grafik = function(x_lower,x_upper, # selang x
                          y_lower,y_upper, # selang y
                          delta, # selang gambar
                          f){ # fungsi f(x,y)

  # generate selang
  selang_x = seq(x_lower,x_upper,by = delta)
  selang_y = seq(y_lower,y_upper,by = delta)

  # menghitung (x,y) yang memenuhi f(x,y) = 1
  df =
    # mengeluarkan semua kombinasi yang mungkin dari selang
    expand.grid(selang_x,selang_y) %>%
```

```
as.data.frame() %>%  
# mengubah nama variabel menjadi x,y  
rename(x = Var1,  
        y = Var2) %>%  
# menghitung nilai f(x,y)  
mutate(f = f(x,y)) %>%  
# hanya mengambil (x,y) yang memenuhi f(x,y) = 1  
filter(round(f,2) == 1)  
# membuat grafik  
df %>%  
  ggplot(aes(x,y)) +  
  geom_point(size = .1,  
             color = "steelblue") +  
  theme_minimal() +  
  geom_vline(xintercept = 0,color = "black") +  
  geom_hline(yintercept = 0,color = "black") +  
  labs(title = "Grafik f(x,y)",  
        caption = "Digambar dengan R\n20921004@mahasiswa.itb.ac.id")  
}
```

2 SOAL 1

Algoritma Komputasi Numerik

2.1 Soal Utama

Diketahui sebuah fungsi:

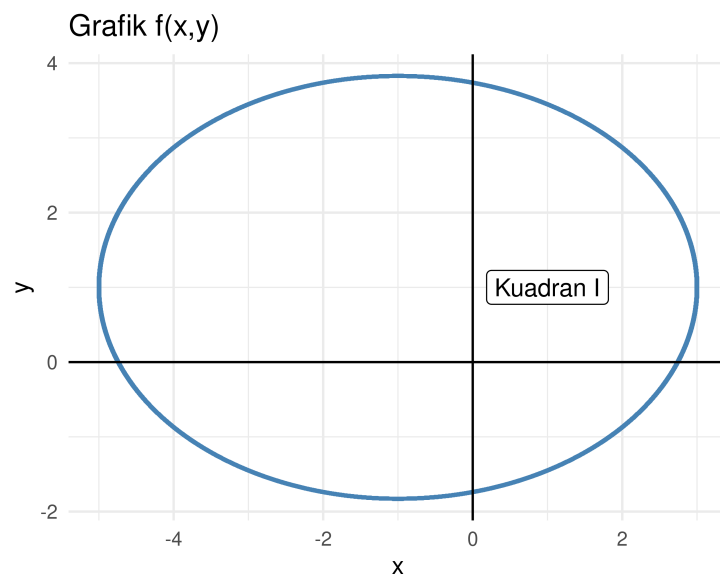
$$\frac{(x+1)^2}{16} + \frac{(y-1)^2}{8} = 1$$

2.1.1 Sub Soal I

Gambarlah fungsi tersebut. Hitunglah luas area di bawah kurva pada kuadran pertama untuk nilai $x \in [0, \sqrt{14} - 1]$ dengan metode partisi trapesium.

2.1.2 Jawaban Sub Soal I

Gambar Fungsi Berikut adalah gambar fungsi yang saya buat dengan **R**.



Digambar dengan R
20921004@mahasiswa.itb.ac.id

Figure 1: Grafik $f(x,y)$

Sekarang kita akan menghitung luas area pada kuadran I di selang $x \in [0, \sqrt{(14) - 1}]$. Saya akan gambarkan selang tersebut dengan garis merah sebagai berikut:

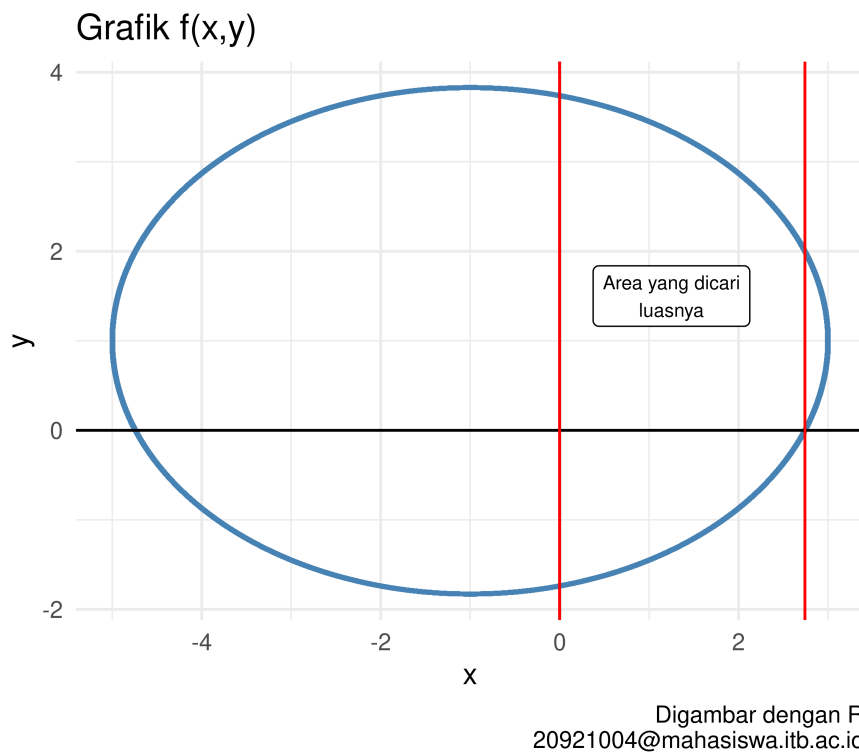


Figure 2: Area Luas

Mengubah Fungsi Untuk memudahkan, kita perlu memodifikasi fungsi $f(x, y)$ ke dalam bentuk $y = g(x)$ yang lebih sederhana.

$$y = 1 + \sqrt{8 - \frac{(x+1)^2}{2}}$$

Karena kita akan menghitung luas area di kuadran I, maka nilai akar yang dihasilkan kita akan ambil hanya yang bernilai **positif** saja.

Luas Area di Bawah Kurva Ide dasar untuk menghitung luas area di bawah kurva adalah:

$$L = \text{alas} \times \text{tinggi}$$

Pada partisi trapesium, tinggi yang akan digunakan adalah: $\text{tinggi} = \frac{f(x_1) + f(x_2)}{2}$

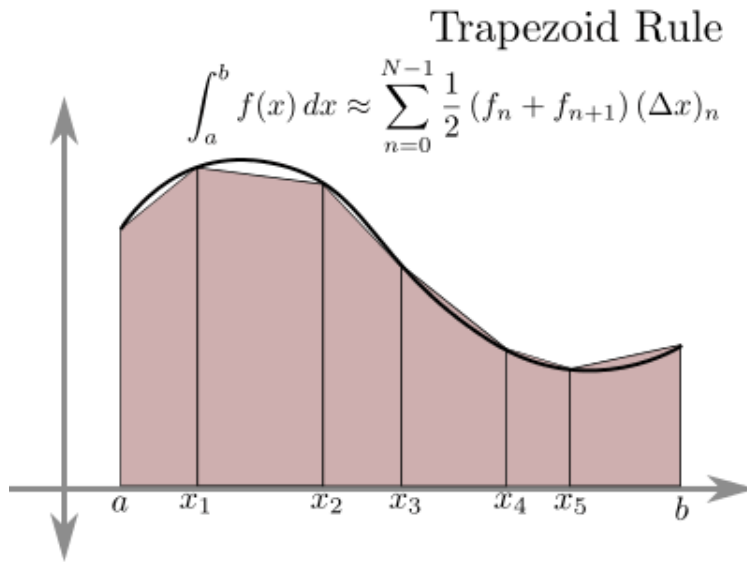


Figure 3: sumber: bragitoff.com

Pada metode trapesium ini, penentuan berapa banyak selang akan mempengaruhi seberapa akurat hasilnya.

Barikut adalah program **luastrap** yang saya buat di **R**:

```
luas_trap = function(x0, # titik awal
                     xn, # titik akhir
                     n,  # banyak selang
                     f){ # fungsi y = f(x)

  # menghitung delta x
  h = (xn - x0) / n
  # menghitung f di x0
  f0 = f(x0)
  # selang pertama
  i = 1
  k = x0 + i*h
  fn = f(k)
  integration = (f0+fn)/2
  # iterasi untuk selang berikutnya hingga selesai
  for(i in 2:n){
```



```
f0 = fn
k = x0 + i*h
fn = f(k)
temp = (f0+fn)/2
integration = integration + temp
}
integration = integration * h
return(integration)
}
```

Sekarang kita akan bandingkan hasilnya untuk berbagai banyak selang.

Table 1: Hasil Perhitungan Luas

n banyak selang	Luas aproksimasi
10	8.64494288
50	8.65494631
100	8.65526330
200	8.65534260
1000	8.65536798
2500	8.65536887
5000	8.65536899
100000	8.65536904
250000	8.65536904
500000	8.65536904
750000	8.65536904
1000000	8.65536904

Terlihat bahwa semakin banyak selangnya, hasilnya konvergen ke suatu nilai yang sama yakni: **8.65536904**.