# Modul Praktikum 8 Komputasi Statistik

**Numerical Integration 2 – Metode Simpson 3/8, Boole, dan Romberg**

Tim Dosen Komputasi Statistik

---

# Capaian Pembelajaran

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Menjelaskan ide dasar metode Simpson 3/8, Boole, dan Romberg.
2. Menurunkan rumus dan memahami bentuk komposit dari ketiga metode.
3. Mengimplementasikan algoritma tiap metode menggunakan bahasa R.
4. Membandingkan hasil aproksimasi terhadap nilai eksak integral.

# 1 Pendahuluan

Pada pertemuan sebelumnya (*Numerical Integration Part 1*) telah dibahas ide dasar aproksimasi integral menggunakan partisi selang $[a, b]$ dan metode:

- Riemann (kiri, kanan, tengah),
- Trapesium (sederhana dan komposit),
- Simpson 1/3.

Pada pertemuan ini kita melanjutkan ke metode dengan orde lebih tinggi, yaitu Simpson 3/8, Boole, dan Romberg. Metode-metode ini:

- menggunakan lebih banyak titik evaluasi fungsi,
- menghasilkan galat yang lebih kecil untuk lebar langkah $h$ yang sama,
- banyak digunakan dalam praktik integrasi numerik tingkat lanjut.

# 2 Dasar Teori

## 2.1 Metode Simpson 3/8

Simpson 3/8 Rule adalah metode Newton–Cotes dengan derajat tiga. Untuk satu panel, interval $[a, b]$ dibagi menjadi tiga sub-selang sama panjang:

$$h = \frac{b-a}{3}, \quad x_0 = a,\ x_1 = a+h,\ x_2 = a+2h,\ x_3 = b.$$

Rumus satu panel:

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right].$$

Untuk bentuk komposit, misalkan $[a, b]$ dibagi menjadi $n$ sub-selang dengan $n$ kelipatan 3 dan $h = \frac{b-a}{n}$, maka:

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}\left[f(x_0) + f(x_n) + 3\sum_{\substack{i=1\\ i\equiv 1,2 \pmod 3}}^{n-1} f(x_i) + 2\sum_{\substack{i=3\\ i\equiv 0 \pmod 3}}^{n-3} f(x_i)\right].$$

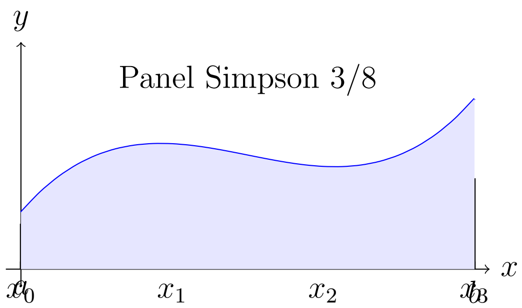


Figure 1: Ilustrasi satu panel Simpson 3/8 pada interval $[a, b]$.

## 2.2 Metode Boole

Boole's Rule adalah Newton–Cotes orde 4 yang menggunakan 5 titik dengan jarak sama $h = \frac{b-a}{4}$:

$$x_0 = a,\ x_1 = a + h,\ x_2 = a + 2h,\ x_3 = a + 3h,\ x_4 = b.$$

Rumus:

$$\int_a^b f(x)\,dx \approx \frac{2h}{45}\left[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\right].$$

Untuk bentuk komposit, interval $[a, b]$ dibagi menjadi $n$ sub-selang dengan $n$ kelipatan 4 dan $h = \frac{b-a}{n}$, dengan pola koefisien $7, 32, 12, 32, 7$ yang berulang.

## 2.3 Metode Romberg

Metode Romberg membangun tabel nilai aproksimasi integral berdasarkan aturan trapesium komposit dengan jumlah panel yang berlipat dua, lalu melakukan *Richardson Extrapolation.*

Misalkan $R_{k,1}$ adalah hasil aturan trapesium dengan $n = 2^{k-1}$ sub-selang:

$$R_{k,1} = T(h_k), \quad h_k = \frac{b-a}{2^{k-1}}.$$

Kemudian elemen-elemen lainnya dihitung dengan:

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}, \quad j = 2, \ldots, k.$$

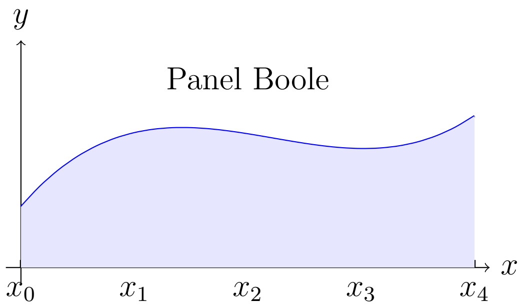


Figure 2: Ilustrasi satu panel Boole's Rule pada interval $[a, b]$.

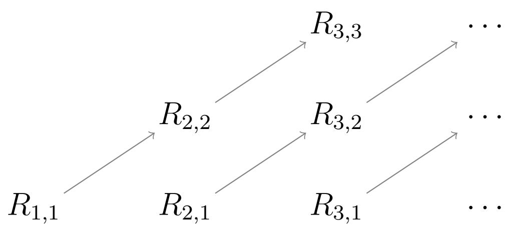


Figure 3: Skema tabel Romberg: setiap kolom menggunakan extrapolation dari kolom sebelumnya.

# 3 Algoritma Metode

## 3.1 Algoritma Simpson 3/8

**Algorithm 1** Algoritma Simpson 3/8

Input fungsi $f(x)$, batas $a, b$, jumlah sub-selang $n$ (kelipatan 3).
Hitung $h = \dfrac{b-a}{n}$.
Bentuk titik $x_i = a + ih$, untuk $i = 0, 1, \ldots, n$.
Hitung $f(x_i)$ untuk semua $i$.
Inisialisasi $S \leftarrow f(x_0) + f(x_n)$.
**for** $i = 1$ sampai $n - 1$ **do**
  **if** $i \equiv 0 \pmod 3$ **then**
    $S \leftarrow S + 2f(x_i)$
  **else**
    $S \leftarrow S + 3f(x_i)$
  **end if**
**end for**
Hitung aproksimasi integral: $I \leftarrow \dfrac{3h}{8} S$.
Kembalikan $I$.

## Contoh 1: Simpson 3/8

Hitung:

$$I_1 = \int_0^3 \left(e^{-x}\sin(x^2) + 3x^2\right)\,dx$$

dengan $n = 300$ menggunakan kode Simpson 3/8 di atas. Bandingkan hasil numerik dengan `integrate()` dan catat galat absolutnya.

## 3.2 Simpson 3/8

```r
################################################################
#        METODE SIMPSON 3/8
#        Numerical Integration Part 2 - Komputasi Statistik
################################################################

# Definisi fungsi (bisa diganti)
f <- function(x) exp(-x) * sin(x^2) + 3 * x^2

# Batas integrasi
a <- 0
b <- 3

# Jumlah sub-selang (harus kelipatan 3)
n <- 300

# Lebar langkah
h <- (b - a) / n

# Titik x dan nilai f(x)
x  <- numeric(n + 1)
fx <- numeric(n + 1)

for (i in 0:n) {
  x[i + 1]  <- a + i * h
  fx[i + 1] <- f(x[i + 1])
}

# Rumus Simpson 3/8 Komposit
S <- fx[1] + fx[n + 1]

for (i in 2:n) {
  if ((i - 1) %% 3 == 0) {
    S <- S + 2 * fx[i]
  } else {
    S <- S + 3 * fx[i]
  }
}

I_s38 <- (3 * h / 8) * S

```

```r
41 # Bandingkan dengan hasil bawaan R
42 I_builtin <- integrate(f, lower = a, upper = b)$value
43
44 cat("=====================================================\n")
45 cat("                 METODE SIMPSON 3/8\n")
46 cat("=====================================================\n")
47 cat("Jumlah sub-selang (n):", n, "\n")
48 cat("Hasil Simpson 3/8 Komposit :", I_s38, "\n")
49 cat("Hasil integrate() (built-in):", I_builtin, "\n")
50 cat("Galat absolut              :", abs(I_builtin - I_s38), "\n")
51 cat("=====================================================\n")
```

**Penjelasan Kode**

1. Baris 7: Mendefinisikan fungsi yang akan diintegralkan.
2. Baris 10–11: Menetapkan batas integrasi $a$ dan $b$.
3. Baris 14: Memilih jumlah sub-selang $n$ (harus kelipatan 3).
4. Baris 17: Menghitung panjang langkah $h$.
5. Baris 20–26: Membuat vektor titik $x_i$ dan nilai fungsi $f(x_i)$.
6. Baris 29: Inisialisasi jumlah $S$ dengan $f(x_0) + f(x_n)$.
7. Baris 31–37: Mengisi koefisien:
   - jika $(i-1) \mod 3 = 0 \to$ koefisien 2,
   - selain itu $\to$ koefisien 3.
8. Baris 39: Menghitung nilai aproksimasi menggunakan $(3h/8)S$.
9. Baris 41–51: Membandingkan hasil dengan fungsi bawaan R `integrate()`.

**Catatan:** Metode Simpson 3/8 memiliki orde galat $O(h^4)$, sehingga semakin kecil $h$ (semakin besar $n$), hasil perhitungan semakin akurat.

## 3.3 Algoritma Boole Komposit

### Contoh 2: Boole

Hitung:

$$I_2 = \int_1^5 \left(e^{-x}\sin(x^2) + 3x^2\right)\,dx$$

dengan $n = 400$ menggunakan Boole Komposit.

### Boole's Rule - Code R

**Algorithm 2** Algoritma Boole's Rule Komposit

1: Input fungsi $f(x)$, batas $a, b$, jumlah sub-selang $n$ (kelipatan 4).
2: Hitung $h = \dfrac{b-a}{n}$.
3: Bentuk titik $x_i = a + ih$, untuk $i = 0, 1, \dots, n$.
4: Hitung $f(x_i)$ untuk semua $i$.
5: Inisialisasi $S \leftarrow 7f(x_0) + 7f(x_n)$.
6: **for** $i = 1$ sampai $n - 1$ **do**
7: $\quad k \leftarrow i \bmod 4$
8: $\quad$ **if** $k = 1$ atau $k = 3$ **then**
9: $\qquad S \leftarrow S + 32f(x_i)$
10: $\quad$ **else if** $k = 2$ **then**
11: $\qquad S \leftarrow S + 12f(x_i)$
12: $\quad$ **end if**
13: **end for**
14: Hitung aproksimasi integral: $I \leftarrow \dfrac{2h}{45}S$.
15: Kembalikan $I$.

```r
#################################################################
#        METODE BOOLE RULE
#        Numerical Integration Part 2
#################################################################

# Definisi fungsi (bisa diganti)
f <- function(x) exp(-x) * sin(x^2) + 3 * x^2

# Batas integrasi
a <- 1
b <- 5

# Jumlah sub-selang (harus kelipatan 4)
n <- 400

# Lebar langkah
h <- (b - a) / n

# Titik x dan nilai f(x)
x  <- numeric(n + 1)
fx <- numeric(n + 1)

for (i in 0:n) {
  x[i + 1]  <- a + i * h
  fx[i + 1] <- f(x[i + 1])
}

# Rumus Boole Rule Komposit
S <- 7 * fx[1] + 7 * fx[n + 1]

for (i in 2:n) {
```

```r
32   k <- (i - 1) %% 4
33   if (k == 1 || k == 3) {
34     S <- S + 32 * fx[i]
35   } else if (k == 2) {
36     S <- S + 12 * fx[i]
37   }
38 }
39 
40 I_boole <- (2 * h / 45) * S
41 
42 # Bandingkan dengan hasil bawaan R
43 I_builtin <- integrate(f, lower = a, upper = b)$value
44 
45 cat("=====================================================\n")
46 cat("                     METODE BOOLE\n")
47 cat("=====================================================\n")
48 cat("Jumlah sub-selang (n):", n, "\n")
49 cat("Hasil Boole Komposit      :", I_boole, "\n")
50 cat("Hasil integrate() (built-in):", I_builtin, "\n")
51 cat("Galat absolut              :", abs(I_builtin - I_boole), "\n")
52 cat("=====================================================\n")
```

### Penjelasan Kode

1. Baris 7: Mendefinisikan fungsi $f(x)$.
2. Baris 10–11: Menentukan batas integrasi.
3. Baris 14: Menentukan sub-selang (harus kelipatan 4).
4. Baris 17: Menghitung panjang langkah $h$.
5. Baris 20–26: Menghasilkan titik $x_i$ dan nilai $f(x_i)$.
6. Baris 29: Inisialisasi dengan $7f(x_0) + 7f(x_n)$.
7. Baris 31–38: Mengisi koefisien Boole:
   - posisi ke-1 dan ke-3 mod 4 $\to$ koefisien 32,
   - posisi ke-2 mod 4 $\to$ koefisien 12.
8. Baris 40: Menghitung nilai akhir $(2h/45)S$.
9. Baris 42–52: Membandingkan dengan hasil bawaan R.

**Catatan:** Boole memiliki orde galat $O(h^6)$ sehingga ini adalah salah satu Newton–Cotes paling akurat sebelum rentan osilasi.

## 3.4 Algoritma Romberg

**Algorithm 3** Algoritma Romberg Integration

Input fungsi $f(x)$, batas $a, b$, maksimum iterasi $K$.
Bentuk matriks $R$ berukuran $K \times K$.
**for** $k = 1$ sampai $K$ **do**
    $n \leftarrow 2^{k-1}$, $h \leftarrow \dfrac{b-a}{n}$.
    Hitung titik $x_i = a + ih$ dan $f(x_i)$ untuk $i = 0, \dots, n$.
    $R_{k,1} \leftarrow \dfrac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]$.
**end for**
**for** $j = 2$ sampai $K$ **do**
    **for** $k = j$ sampai $K$ **do**
        $R_{k,j} \leftarrow R_{k,j-1} + \dfrac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}$.
    **end for**
**end for**
Kembalikan $R$ sebagai tabel Romberg.

### Contoh 3: Romberg

Hitung:

$$I_3 = \int_0^{\pi/2} \left(e^{-x}\sin(x^2) + 3x^2\right)\, dx$$

dan amati bagaimana nilai pada tabel Romberg mendekati nilai eksak.

### Romberg Integration - Code R

```r
####################################################################
#        METODE ROMBERG INTEGRATION
#        Numerical Integration Part 2 - Komputasi Statistik
####################################################################

# Definisi fungsi (bisa diganti)
f <- function(x) exp(-x) * sin(x^2) + 3 * x^2

# Batas integrasi
a <- 0
b <- pi / 2

# Maksimum level iterasi
max_k <- 5

# Matriks kosong untuk tabel Romberg
R <- matrix(0, nrow = max_k, ncol = max_k)

```

```
# Langkah 1: Aturan Trapesium untuk setiap level
for (k in 1:max_k) {
  n <- 2^(k - 1)
  h <- (b - a) / n
  x  <- seq(a, b, by = h)
  fx <- f(x)
  R[k, 1] <- (h / 2) * (fx[1] + 2 * sum(fx[2:n]) + fx[n + 1])
}

# Langkah 2: Richardson Extrapolation
for (j in 2:max_k) {
  for (k in j:max_k) {
    R[k, j] <- R[k, j - 1] +
      (R[k, j - 1] - R[k - 1, j - 1]) / (4^(j - 1) - 1)
  }
}

colnames(R) <- paste0("R", 1:max_k)
rownames(R) <- paste0("Iter-", 1:max_k)

print(R)

# Bandingkan dengan hasil eksak bawaan R
I_builtin <- integrate(f, lower = a, upper = b)$value
I_romberg <- R[max_k, max_k]

cat("=====================================================\n")
cat("             METODE ROMBERG - HASIL\n")
cat("=====================================================\n")
cat("Level Iterasi (max_k):", max_k, "\n")
cat("Hasil Romberg Integration :", I_romberg, "\n")
cat("Hasil integrate() (built-in):", I_builtin, "\n")
cat("Galat absolut              :", abs(I_builtin - I_romberg), "
  \n")
cat("=====================================================\n")
```

**Penjelasan Kode**

1. Baris 7: Mendefinisikan fungsi yang akan diintegralkan.
2. Baris 10–11: Menentukan batas integrasi $a, b$.
3. Baris 14: Menentukan jumlah iterasi maksimum Romberg.
4. Baris 17: Membuat tabel kosong $R$.
5. Baris 20–26: Mengisi kolom pertama $R_{k,1}$ dengan aturan trapesium.
6. Baris 29–34: Melakukan Richardson Extrapolation untuk menghasilkan $R_{k,j}$.
7. Baris 36–37: Memberi nama kolom dan baris tabel Romberg.
8. Baris 42–43: Mengambil hasil akhir Romberg $R_{K,K}$.

**Catatan:** Romberg adalah metode orde tinggi (superkonvergen) yang menggabungkan trapesium + extrapolation sehingga galat turun sangat cepat.

# 4 Analisis

**Analisis Konvergensi**

Bandingkan kecepatan konvergensi:

- Simpson 3/8 vs Boole vs Romberg untuk fungsi yang sama. Anda bisa membandingkan jumlah n yang di perlukan untuk setiap metode mencapai konvergensi (Sama dengan hasil perhitungan Eksak). Untuk itu anda perlu mengubah-ubah jumlah n secara bertahap sesuai ketentuan setiap algoritma (Simpson 3/8 dengan kelipan 3, Boole dengan kelipatan 4, dan Romberg dengan mengganti nilai Max-k ).

# 5 Latihan

1. Gunakan ketiga metode (Simpson 3/8, Boole, Romberg) untuk menghitung:

$$\int_0^{2\pi} (\sin x + e^{-x})\, dx.$$

   Bandingkan hasil dan galat relatif masing-masing metode.

2. Untuk Simpson 3/8 dan Boole, buat grafik hubungan antara $n$ dan galat absolut (gunakan beberapa nilai $n$ yang berbeda).

# 6 Tugas Praktikum

1. Definisikan fungsi:
$$f(x) = \cos(x^2) + e^{-x}\ln(x+1).$$
   Hitung:
   - $I_1 = \int_0^3 f(x)\,dx$ (Simpson 3/8),
   - $I_2 = \int_0^4 f(x)\,dx$ (Boole),
   - $I_3 = \int_0^2 f(x)\,dx$ (Romberg dengan $max\_k = 6$).
2. Bandingkan hasil numerik dengan `integrate()`.
3. Buat tabel yang berisi nilai aproksimasi, nilai eksak, dan galat relatif untuk setiap metode.

---

Modul Praktikum Komputasi Statistik – Pertemuan 12
Program Studi Sains Data, Institut Teknologi Sumatera