UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Komputasi Statistika

#11 Meeting

Root Finding : Secant & Bisection

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001

Root Finding

Metode Secant

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Metode Secant adalah sebuah metode pencarian akar (root finding) iteratif yang menggunakan aproksimasi garis potong (secant line) untuk menghampiri akar sebuah fungsi. Metode ini sering digunakan sebagai alternatif dari Metode Newton-Raphson, terutama ketika turunan pertama fungsi ($f'(x)$) sulit atau tidak efisien untuk dihitung.

- Metode ini tidak memerlukan perhitungan turunan analitik ($f'(x)$), yang merupakan kelemahan utama Metode Newton-Raphson.

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

- Pada setiap iterasi baru (misal, untuk mencari x_3), metode ini menggunakan dua nilai x terakhir yang dihitung (yaitu x_2 dan x_1), dan tidak lagi menggunakan nilai x terlama (x_0).

Root Finding

Metode Secant

Contoh 1: Mencari Akar dari $f(x) = \cos(x) - x$

Rumus Iterasi: $x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$ **Rumus Error:** $\text{Error} = |x_{n+1} - x_n|$

- **Tebakan Awal:** $x_0 = 0.5$ dan $x_1 = 0.6$
- **Nilai Fungsi:**
  - $f(x_0) = f(0.5) = \cos(0.5) - 0.5 \approx 0.3775825619$
  - $f(x_1) = f(0.6) = \cos(0.6) - 0.6 \approx 0.2253356149$

Iterasi 1 (Mencari x_2):

1. **Numerator:** $f(x_1) \times (x_1 - x_0) = 0.2253356149 \times (0.6 - 0.5) = 0.0225335615$
2. **Denominator:** $f(x_1) - f(x_0) = 0.2253356149 - 0.3775825619 = -0.1522469470$
3. x_2: $0.6 - (0.0225335615 / -0.1522469470) \approx 0.6 - (-0.148006663) \approx \mathbf{0.748006663}$
4. **Error:** $|0.748006663 - 0.6| = \mathbf{0.148006663}$

Iterasi 2 (Mencari x_3):

- **Tebakan Baru:** $x_1 = 0.6$ dan $x_2 = 0.748006663$
- **Nilai Fungsi Baru:**
  - $f(x_1) \approx 0.2253356149$
  - $f(x_2) = f(0.748006663) \approx -0.0149605011$

1. **Numerator:** $f(x_2) \times (x_2 - x_1) = -0.0149605011 \times (0.748006663 - 0.6) = -0.002214249...$
2. **Denominator:** $f(x_2) - f(x_1) = -0.0149605011 - 0.2253356149 = -0.2402961160$
3. x_3: $0.748006663 - (-0.002214249.../-0.2402961160) \approx 0.748006663 - 0.009214687 \approx \mathbf{0.738791976}$
4. **Error:** $|0.738791976 - 0.748006663| = \mathbf{0.009214687}$

Root Finding

Metode Secant (Contoh 1)

```
f <- function(x) { return(cos(x) - x) }

x_prev <- 0.5; x_curr <- 0.6
fx_prev <- f(x_prev); fx_curr <- f(x_curr)
tolerance <- 1e-7; max_iter <- 20
iter <- 0; error <- 1

results <- data.frame(Iterasi=integer(), x_n_min_1=double(), x_n=double(),
                      fx_n=double(), x_n_plus_1=double(), Error=double())
```

```
while (error > tolerance && iter < max_iter) {
 iter <- iter + 1
  x_new <- x_curr - (fx_curr * (x_curr - x_prev)) / (fx_curr - fx_prev)
 error <- abs(x_new - x_curr)

  results[iter, ] <- c(iter, x_prev, x_curr, fx_curr, x_new, error)

 x_prev <- x_curr; fx_prev <- fx_curr
 x_curr <- x_new; fx_curr <- f(x_curr)
}
print(results, digits = 8)
```

Root Finding

Metode Secant (Contoh 1)

```
> print(results, digits = 8)
  Iterasi  x_n_min_1        x_n           fx_n x_n_plus_1         Error
1       1 0.50000000 0.60000000  2.2533561e-01 0.74800666 1.4800666e-01
2       2 0.60000000 0.74800666 -1.4960501e-02 0.73879197 9.2146879e-03
3       3 0.74800666 0.73879197  4.9061313e-04 0.73908456 2.9259035e-04
4       4 0.73879197 0.73908456  9.6216332e-07 0.73908513 5.7493954e-07
5       5 0.73908456 0.73908513 -6.2292838e-11 0.73908513 3.7220560e-11
```

Root Finding

Metode Secant (Contoh 2)

Mencari Akar dari $f(x) = x^2 - x - 2 = 0$

Skenario A (Target $x = 2$)

- Tebakan Awal: $x_0 = 1.5$ dan $x_1 = 1.6$
- Nilai Fungsi:
  - $f(x_0) = f(1.5) = (1.5)^2 - 1.5 - 2 = 2.25 - 1.5 - 2 = -\mathbf{1.25}$
  - $f(x_1) = f(1.6) = (1.6)^2 - 1.6 - 2 = 2.56 - 1.6 - 2 = -\mathbf{1.04}$

Iterasi 1 (Mencari x_2):

1. Numerator: $f(x_1) \times (x_1 - x_0) = -1.04 \times (1.6 - 1.5) = -0.104$
2. Denominator: $f(x_1) - f(x_0) = -1.04 - (-1.25) = 0.21$
3. x_2: $1.6 - (-0.104/0.21) \approx 1.6 - (-0.495238095) \approx \mathbf{2.09523810}$
4. Error: $|2.09523810 - 1.6| = \mathbf{0.49523810}$

Root Finding

Metode Secant (Contoh 2)

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Iterasi 2 (Mencari x_3):

- **Tebakan Baru:** $x_1 = 1.6$ dan $x_2 = 2.09523810$
- **Nilai Fungsi Baru:**
  - $f(x_1) = -1.04$
  - $f(x_2) = f(2.09523810) \approx 0.29478458$

1. **Numerator:** $f(x_2) \times (x_2 - x_1) = 0.29478458 \times (2.09523810 - 1.6) \approx 0.1460517...$
2. **Denominator:** $f(x_2) - f(x_1) = 0.29478458 - (-1.04) = 1.33478458$
3. x_3: $2.09523810 - (0.1460517.../1.33478458) \approx 2.09523810 - 0.10937237... \approx \mathbf{1.98586572}$
4. **Error:** $|1.98586572 - 2.09523810| = \mathbf{0.10937237}$

Root Finding

Metode Secant (Contoh 2)

Skenario B (Target $x = -1$)

- **Tebakan Awal:** $x_0 = -0.5$ dan $x_1 = -0.6$
- **Nilai Fungsi:**
  - $f(x_0) = f(-0.5) = (-0.5)^2 - (-0.5) - 2 = 0.25 + 0.5 - 2 = -\mathbf{1.25}$
  - $f(x_1) = f(-0.6) = (-0.6)^2 - (-0.6) - 2 = 0.36 + 0.6 - 2 = -\mathbf{1.04}$

Iterasi 1 (Mencari x_2):

1. **Numerator:** $f(x_1) \times (x_1 - x_0) = -1.04 \times (-0.6 - (-0.5)) = 0.104$
2. **Denominator:** $f(x_1) - f(x_0) = -1.04 - (-1.25) = 0.21$
3. x_2: $-0.6 - (0.104/0.21) \approx -0.6 - 0.495238095 \approx -\mathbf{1.09523810}$
4. **Error:** $|-1.09523810 - (-0.6)| = \mathbf{0.49523810}$

Root Finding

Metode Secant (Contoh 2)

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Iterasi 2 (Mencari x_3):

- **Tebakan Baru:** $x_1 = -0.6$ dan $x_2 = -1.09523810$
- **Nilai Fungsi Baru:**
  - $f(x_1) = -1.04$
  - $f(x_2) = f(-1.09523810) \approx 0.29478458$

1. **Numerator:** $f(x_2) \times (x_2 - x_1) = 0.29478458 \times (-1.09523810 - (-0.6)) \approx -0.1460517...$
2. **Denominator:** $f(x_2) - f(x_1) = 0.29478458 - (-1.04) = 1.33478458$
3. x_3: $-1.09523810 - (-0.1460517.../1.33478458) \approx -1.09523810 - (-0.10937237...) \approx -\mathbf{0.98586572}$
4. **Error:** $|-0.98586572 - (-1.09523810)| = \mathbf{0.10937237}$

Root Finding

Metode Secant (Contoh 2)

```
# 1. Definisikan f(x)
f <- function(x) {
  return(x^2 - x - 2)
}
# --- Fungsi Helper Secant ---
run_secant <- function(x0, x1, tolerance = 1e-7, max_iter =
20) {
   # Inisialisasi
  x_prev <- x0
  x_curr <- x1
  fx_prev <- f(x_prev)
  fx_curr <- f(x_curr)
   iter <- 0
  error <- 1

  # Siapkan tabel 6-kolom yang benar
  results <- data.frame(Iterasi = integer(),
                        xn_min_1 = double(), # x_n-1
                        xn = double(),       # x_n
                        fxn = double(),      # f(x_n)
                        xn_plus_1 = double(),# x_n+1 (hasil baru)
                        Error = double())
 # Proses Iterasi
 while (error > tolerance && iter < max_iter) {
 iter <- iter + 1
  # Rumus Secant
  x_new <- x_curr - (fx_curr * (x_curr - x_prev)) / (fx_curr -
fx_prev)
  error <- abs(x_new - x_curr) # Error = |x_n+1 - x_n|
  # Simpan hasil lengkap
 results[iter, ] <- c(iter, x_prev, x_curr, fx_curr, x_new, error)
  # Update tebakan
 x_prev <- x_curr
 fx_prev <- fx_curr
  x_curr <- x_new
 fx_curr <- f(x_curr)
 }
 print(results, digits = 8)
}
# Skenario A: Tebakan x0 = 1.5, x1 = 1.6
run_secant(x0 = 1.5, x1 = 1.6)
# Skenario B: Tebakan x0 = -0.5, x1 = -0.6
run_secant(x0 = -0.5, x1 = -0.6)
```

Root Finding

Metode Secant (Contoh 2)

Skenario A (Target $x = 2$)

```
> # Skenario A: Tebakan x0 = 1.5, x1 = 1.6
> run_secant(x0 = 1.5, x1 = 1.6)
  Iterasi  xn_min_1        xn            fxn xn_plus_1         Error
1       1 1.5000000 1.6000000 -1.0400000e+00 2.0952381 4.9523810e-01
2       2 1.6000000 2.0952381  2.9478458e-01 1.9858657 1.0937237e-01
3       3 2.0952381 1.9858657 -4.2203049e-02 1.9995631 1.3697380e-02
4       4 1.9858657 1.9995631 -1.3104967e-03 2.0000021 4.3896430e-04
5       5 1.9995631 2.0000021  6.2053504e-06 2.0000000 2.0687500e-06
6       6 2.0000021 2.0000000 -9.0382768e-10 2.0000000 3.0127567e-10
```

Skenario B (Target $x = -1$)

```
> # Skenario B: Tebakan x0 = -0.5, x1 = -0.6
> run_secant(x0 = -0.5, x1 = -0.6)
  Iterasi    xn_min_1          xn            fxn   xn_plus_1         Error
1       1 -0.50000000 -0.60000000 -1.0400000e+00 -1.09523810 4.9523810e-01
2       2 -0.60000000 -1.09523810  2.9478458e-01 -0.98586572 1.0937237e-01
3       3 -1.09523810 -0.98586572 -4.2203049e-02 -0.99956310 1.3697380e-02
4       4 -0.98586572 -0.99956310 -1.3104967e-03 -1.00000207 4.3896430e-04
5       5 -0.99956310 -1.00000207  6.2053504e-06 -1.00000000 2.0687500e-06
6       6 -1.00000207 -1.00000000 -9.0382768e-10 -1.00000000 3.0127567e-10
```

Root Finding

Metode Bisection

Metode Biseksi (atau Metode Bagi Dua) adalah salah satu metode pencarian akar numerik yang paling dasar dan robust (andal). Prinsip kerjanya didasarkan pada pembagian interval secara berulang (*interval halving*). Metode ini mengidentifikasi sebuah interval yang dijamin memiliki akar, lalu membagi dua interval tersebut secara terus-menerus, dan mempersempit pencarian ke sub-interval di mana akar tersebut berada.

- Metode ini adalah metode "tertutup" (*bracketing method*) yang memerlukan interval awal $[a, b]$.
- **Syarat Awal:** Interval $[a, b]$ harus dipastikan "mengurung" akar. Syarat ini terpenuhi jika $f(a)$ dan $f(b)$ memiliki tanda yang berlawanan (yaitu, $f(a) \times f(b) < 0$). Berdasarkan Teorema Nilai Antara (*Intermediate Value Theorem*), jika syarat ini terpenuhi, setidaknya satu akar dijamin ada di dalam interval $[a, b]$.

Root Finding

Metode Bisection

- **Proses Iterasi:**

  1. Titik tengah interval dihitung: $c = (a + b)/2$.
  2. Nilai fungsi di titik tengah, $f(c)$, dievaluasi.
  3. Jika $f(c)$ memiliki tanda yang berlawanan dengan $f(a)$ (yaitu $f(a) \times f(c) < 0$), maka akar berada di sub-interval $[a, c]$. Interval baru diset menjadi $[a, c]$ (dimana b baru $= c$).
  4. Jika tidak (yaitu $f(c)$ berlawanan tanda dengan $f(b)$), maka akar berada di sub-interval $[c, b]$. Interval baru diset menjadi $[c, b]$ (dimana a baru $= c$).

- Proses ini diulang, di mana lebar interval $[a, b]$ berkurang setengahnya pada setiap iterasi, hingga interval tersebut cukup kecil (sesuai batas toleransi error).

Root Finding

Metode Bisection (Contoh 1)

Contoh 1: Mencari Akar dari $f(x) = \cos(x) - x$

Rumus Iterasi: $c = (a + b)/2$ **Rumus Error:** $\text{Error} = (b - a)/2$

- **Kurungan Awal:**
  - $a = 0.5 \rightarrow f(0.5) = \cos(0.5) - 0.5 \approx 0.37758...$ (**Positif**)
  - $b = 1.0 \rightarrow f(1.0) = \cos(1.0) - 1.0 \approx -0.45970...$ (**Negatif**)
  - Syarat $f(a) \times f(b) < 0$ terpenuhi.

Iterasi 1:

1. c_1 **(Titik Tengah):** $(0.5 + 1.0)/2 = \mathbf{0.75}$
2. $f(c_1)$: $f(0.75) = \cos(0.75) - 0.75 \approx -0.01831...$ (**Negatif**)
3. **Error:** $(1.0 - 0.5)/2 = \mathbf{0.25}$
4. **Update Kurungan:** $f(c_1)$ (Negatif) menggantikan $f(b)$. Kurungan baru: $[0.5, 0.75]$

Root Finding

Metode Bisection (Contoh 1)

Iterasi 2:

1. c_2 **(Titik Tengah):** $(0.5 + 0.75)/2 = \mathbf{0.625}$
2. $f(c_2)$: $f(0.625) = \cos(0.625) - 0.625 \approx 0.18596...$ **(Positif)**
3. **Error:** $(0.75 - 0.5)/2 = \mathbf{0.125}$
4. **Update Kurungan:** $f(c_2)$ (Positif) menggantikan $f(a)$. Kurungan baru: $[0.625, 0.75]$

Iterasi 3:

1. c_3 **(Titik Tengah):** $(0.625 + 0.75)/2 = \mathbf{0.6875}$
2. $f(c_3)$: $f(0.6875) = \cos(0.6875) - 0.6875 \approx 0.08533...$ **(Positif)**
3. **Error:** $(0.75 - 0.625)/2 = \mathbf{0.0625}$
4. **Update Kurungan:** $f(c_3)$ (Positif) menggantikan $f(a)$. Kurungan baru: $[0.6875, 0.75]$

Root Finding

Metode Bisection (Contoh 1)

```
# 1. Definisikan f(x)
f <- function(x) {
  return(cos(x) - x)
}
# 2. Inisialisasi (batas a dan b)
a <- 0.5
b <- 1.0
fa <- f(a)
fb <- f(b)
# Cek syarat awal
if (fa * fb >= 0) {
  stop("Syarat Biseksi tidak terpenuhi. f(a) dan f(b) harus beda tanda.")
}
tolerance <- 1e-7
max_iter <- 30 # Biseksi butuh lebih banyak iterasi
iter <- 0
# Siapkan tabel untuk hasil
results <- data.frame(Iterasi = integer(),
                      a = double(),
                      b = double(),
                      c = double(),      # Titik tengah
                      fc = double(),     # f(c)
                      Error = double())
```

```
# 3. Proses Iterasi
while ( (b - a) / 2 > tolerance && iter < max_iter) {
 iter <- iter + 1
  c <- (a + b) / 2
 fc <- f(c)
 error <- (b - a) / 2
  results[iter, ] <- c(iter, a, b, c, fc, error)
  # Perbarui kurungan
 if (fa * fc < 0) { # Akar ada di [a, c]
 b <- c
 fb <- fc
 } else { # Akar ada di [c, b]
 a <- c
 fa <- fc
 }
}

# Tampilkan hasil

print(results, digits = 8)
```

Root Finding

Metode Bisection (Contoh 1)

```
> print(results, digits = 8)
   Iterasi          a          b          c            fc        Error
1        1 0.50000000 1.00000000 0.75000000 -1.8311131e-02 2.5000000e-01
2        2 0.50000000 0.75000000 0.62500000  1.8596312e-01 1.2500000e-01
3        3 0.62500000 0.75000000 0.68750000  8.5334946e-02 6.2500000e-02
4        4 0.68750000 0.75000000 0.71875000  3.3879372e-02 3.1250000e-02
5        5 0.71875000 0.75000000 0.73437500  7.8747255e-03 1.5625000e-02
6        6 0.73437500 0.75000000 0.74218750 -5.1957117e-03 7.8125000e-03
7        7 0.73437500 0.74218750 0.73828125  1.3451498e-03 3.9062500e-03
8        8 0.73828125 0.74218750 0.74023438 -1.9238728e-03 1.9531250e-03
9        9 0.73828125 0.74023438 0.73925781 -2.8900915e-04 9.7656250e-04
10      10 0.73828125 0.73925781 0.73876953  5.2815843e-04 4.8828125e-04
11      11 0.73876953 0.73925781 0.73901367  1.1959667e-04 2.4414062e-04
12      12 0.73901367 0.73925781 0.73913574 -8.4700731e-05 1.2207031e-04
13      13 0.73901367 0.73913574 0.73907471  1.7449347e-05 6.1035156e-05
14      14 0.73907471 0.73913574 0.73910522 -3.3625348e-05 3.0517578e-05
15      15 0.73907471 0.73910522 0.73908997 -8.0879147e-06 1.5258789e-05
16      16 0.73907471 0.73908997 0.73908234  4.6807375e-06 7.6293945e-06
17      17 0.73908234 0.73908997 0.73908615 -1.7035833e-06 3.8146973e-06
18      18 0.73908234 0.73908615 0.73908424  1.4885784e-06 1.9073486e-06
19      19 0.73908424 0.73908615 0.73908520 -1.0750208e-07 9.5367432e-07
20      20 0.73908424 0.73908520 0.73908472  6.9053827e-07 4.7683716e-07
21      21 0.73908472 0.73908520 0.73908496  2.9151812e-07 2.3841858e-07
22      22 0.73908496 0.73908520 0.73908508  9.2008025e-08 1.1920929e-07
```

Root Finding

Metode Bisection (Contoh 2)

Mencari Akar dari $f(x) = x^2 - x - 2 = 0$

Skenario A (Target $x = 2$)

- **Kurungan Awal:**
  - $a = 1.5 \rightarrow f(1.5) = (1.5)^2 - 1.5 - 2 = -1.25$ (**Negatif**)
  - $b = 3 \rightarrow f(3) = (3)^2 - 3 - 2 = 4$ (**Positif**)
  - Syarat $f(a) \times f(b) < 0$ terpenuhi.

Iterasi 1:

1. c_1 **(Titik Tengah):** $(1.5 + 3)/2 = \mathbf{2.25}$
2. $f(c_1)$**:** $f(2.25) = (2.25)^2 - 2.25 - 2 = 0.8125$ (**Positif**)
3. **Error:** $(3 - 1.5)/2 = \mathbf{0.75}$
4. **Update Kurungan:** $f(c_1)$ (Positif) menggantikan $f(b)$. Kurungan baru: $[1.5, 2.25]$

Root Finding

Metode Bisection (Contoh 2)

Iterasi 2:

1. c_2 **(Titik Tengah):** $(1.5 + 2.25)/2 = \mathbf{1.875}$
2. $f(c_2)$: $f(1.875) = (1.875)^2 - 1.875 - 2 = -0.359375$ (**Negatif**)
3. **Error:** $(2.25 - 1.5)/2 = \mathbf{0.375}$
4. **Update Kurungan:** $f(c_2)$ (Negatif) menggantikan $f(a)$. Kurungan baru: $[1.875, 2.25]$

Iterasi 3:

1. c_3 **(Titik Tengah):** $(1.875 + 2.25)/2 = \mathbf{2.0625}$
2. $f(c_3)$: $f(2.0625) = (2.0625)^2 - 2.0625 - 2 = 0.19140625$ (**Positif**)
3. **Error:** $(2.25 - 1.875)/2 = \mathbf{0.1875}$
4. **Update Kurungan:** $f(c_3)$ (Positif) menggantikan $f(b)$. Kurungan baru: $[1.875, 2.0625]$

Root Finding

Metode Bisection (Contoh 2)

Skenario B (Target $x = -1$)

- **Kurungan Awal:**
  - $a = -2 \rightarrow f(-2) = (-2)^2 - (-2) - 2 = 4$ (**Positif**)
  - $b = -0.5 \rightarrow f(-0.5) = (-0.5)^2 - (-0.5) - 2 = -1.25$ (**Negatif**)
  - Syarat $f(a) \times f(b) < 0$ terpenuhi.

Iterasi 1:

1. c_1 **(Titik Tengah):** $(-2 + -0.5)/2 = \mathbf{-1.25}$
2. $f(c_1)$**:** $f(-1.25) = (-1.25)^2 - (-1.25) - 2 = 0.8125$ (**Positif**)
3. **Error:** $(-0.5 - (-2))/2 = \mathbf{0.75}$
4. **Update Kurungan:** $f(c_1)$ (Positif) menggantikan $f(a)$. Kurungan baru: $[-1.25, -0.5]$

Root Finding

Metode Bisection (Contoh 2)

Iterasi 2:

1. c_2 **(Titik Tengah):** $(-1.25 + -0.5)/2 = \mathbf{-0.875}$
2. $f(c_2)$: $f(-0.875) = (-0.875)^2 - (-0.875) - 2 = -0.359375$ (**Negatif**)
3. **Error:** $(-0.5 - (-1.25))/2 = \mathbf{0.375}$
4. **Update Kurungan:** $f(c_2)$ (Negatif) menggantikan $f(b)$. Kurungan baru: $[-1.25, -0.875]$

Iterasi 3:

1. c_3 **(Titik Tengah):** $(-1.25 + -0.875)/2 = \mathbf{-1.0625}$
2. $f(c_3)$: $f(-1.0625) = (-1.0625)^2 - (-1.0625) - 2 = 0.19140625$ (**Positif**)
3. **Error:** $(-0.875 - (-1.25))/2 = \mathbf{0.1875}$
4. **Update Kurungan:** $f(c_3)$ (Positif) menggantikan $f(a)$. Kurungan baru: $[-1.0625, -0.875]$

Root Finding

Metode Bisection (Contoh 2)

```
# 1. Definisikan f(x)
f <- function(x) {
  return(x^2 - x - 2)
}

# Inisialisasi umum
tolerance <- 1e-7
max_iter <- 30

cat("\nKASUS 2: f(x) = x^2 - x - 2\n")
cat("---------------------------------------------\n")

# --- SKENARIO A (Target 2) ---
cat("Skenario A: Target 2 (Kurungan [1.5, 3])\n")
a <- 1.5
b <- 3
fa <- f(a)
fb <- f(b)
iter <- 0
results_A <- data.frame(Iterasi = integer(), a = double(), b = double(),
                        c = double(), fc = double(), Error = double())

if (fa * fb >= 0) {
 cat("Syarat Biseksi Skenario A tidak terpenuhi.\n")
} else {
 while ( (b - a) / 2 > tolerance && iter < max_iter) {
 iter <- iter + 1
 c_A <- (a + b) / 2
 fc_A <- f(c_A)
 error <- (b - a) / 2
 results_A[iter, ] <- c(iter, a, b, c_A, fc_A, error)

 if (fa * fc_A < 0) { b <- c_A; fb <- fc_A }
 else { a <- c_A; fa <- fc_A }
 }
 print(results_A, digits = 8)
 cat("\nAkarnya (A) adalah (estimasi terbaik):", c_A, "\n")
}
```

Root Finding

Metode Bisection (Contoh 2)


```
# --- SKENARIO B (Target -1) ---
cat("Skenario B: Target -1 (Kurungan [-2, -0.5])\n")
a <- -2
b <- -0.5
fa <- f(a)
fb <- f(b)
iter <- 0
results_B <- data.frame(Iterasi = integer(), a = double(), b = double(),
                        c = double(), fc = double(), Error = double())
```

```
if (fa * fb >= 0) {
  cat("Syarat Biseksi Skenario B tidak terpenuhi.\n")
} else {
  while ( (b - a) / 2 > tolerance && iter < max_iter) {
    iter <- iter + 1
    c_B <- (a + b) / 2
    fc_B <- f(c_B)
    error <- (b - a) / 2
    results_B[iter, ] <- c(iter, a, b, c_B, fc_B, error)

    if (fa * fc_B < 0) { b <- c_B; fb <- fc_B }
    else { a <- c_B; fa <- fc_B }
  }
  print(results_B, digits = 8)
}
```

Root Finding

Metode Bisection (Contoh 2)

```
+   print(results_A, digits = 8)
+   cat("\nAkarnya (A) adalah (estimasi terbaik):", c_A, "\n")
+ }
   Iterasi         a         b         c             fc         Error
1        1 1.5000000 3.0000000 2.2500000  8.1250000e-01 7.5000000e-01
2        2 1.5000000 2.2500000 1.8750000 -3.5937500e-01 3.7500000e-01
3        3 1.8750000 2.2500000 2.0625000  1.9140625e-01 1.8750000e-01
4        4 1.8750000 2.0625000 1.9687500 -9.2773438e-02 9.3750000e-02
5        5 1.9687500 2.0625000 2.0156250  4.7119141e-02 4.6875000e-02
6        6 1.9687500 2.0156250 1.9921875 -2.3376465e-02 2.3437500e-02
7        7 1.9921875 2.0156250 2.0039062  1.1734009e-02 1.1718750e-02
8        8 1.9921875 2.0039062 1.9980469 -5.8555603e-03 5.8593750e-03
9        9 1.9980469 2.0039062 2.0009766  2.9306412e-03 2.9296875e-03
10      10 1.9980469 2.0009766 1.9995117 -1.4646053e-03 1.4648438e-03
11      11 1.9995117 2.0009766 2.0002441  7.3248148e-04 7.3242188e-04
12      12 1.9995117 2.0002441 1.9998779 -3.6619604e-04 3.6621094e-04
13      13 1.9998779 2.0002441 2.0000610  1.8310919e-04 1.8310547e-04
14      14 1.9998779 2.0000610 1.9999695 -9.1551803e-05 9.1552734e-05
15      15 1.9999695 2.0000610 2.0000153  4.5776600e-05 4.5776367e-05
16      16 1.9999695 2.0000153 1.9999924 -2.2888125e-05 2.2888184e-05
17      17 1.9999924 2.0000153 2.0000038  1.1444106e-05 1.1444092e-05
18      18 1.9999924 2.0000038 1.9999981 -5.7220423e-06 5.7220459e-06
19      19 1.9999981 2.0000038 2.0000010  2.8610239e-06 2.8610229e-06
20      20 1.9999981 2.0000010 1.9999995 -1.4305112e-06 1.4305115e-06
21      21 1.9999995 2.0000010 2.0000002  7.1525579e-07 7.1525574e-07
22      22 1.9999995 2.0000002 1.9999999 -3.5762785e-07 3.5762787e-07
23      23 1.9999999 2.0000002 2.0000001  1.7881394e-07 1.7881393e-07

Akarnya (A) adalah (estimasi terbaik): 2
```

Root Finding

Metode Bisection (Contoh 2)

```
+   print(results_B, digits = 8)
+ }
   Iterasi          a           b           c            fc         Error
1        1 -2.0000000 -0.50000000 -1.25000000  8.1250000e-01 7.5000000e-01
2        2 -1.2500000 -0.50000000 -0.87500000 -3.5937500e-01 3.7500000e-01
3        3 -1.2500000 -0.87500000 -1.06250000  1.9140625e-01 1.8750000e-01
4        4 -1.0625000 -0.87500000 -0.96875000 -9.2773438e-02 9.3750000e-02
5        5 -1.0625000 -0.96875000 -1.01562500  4.7119141e-02 4.6875000e-02
6        6 -1.0156250 -0.96875000 -0.99218750 -2.3376465e-02 2.3437500e-02
7        7 -1.0156250 -0.99218750 -1.00390625  1.1734009e-02 1.1718750e-02
8        8 -1.0039062 -0.99218750 -0.99804688 -5.8555603e-03 5.8593750e-03
9        9 -1.0039062 -0.99804688 -1.00097656  2.9306412e-03 2.9296875e-03
10      10 -1.0009766 -0.99804688 -0.99951172 -1.4646053e-03 1.4648438e-03
11      11 -1.0009766 -0.99951172 -1.00024414  7.3248148e-04 7.3242188e-04
12      12 -1.0002441 -0.99951172 -0.99987793 -3.6619604e-04 3.6621094e-04
13      13 -1.0002441 -0.99987793 -1.00006104  1.8310919e-04 1.8310547e-04
14      14 -1.0000610 -0.99987793 -0.99996948 -9.1551803e-05 9.1552734e-05
15      15 -1.0000610 -0.99996948 -1.00001526  4.5776600e-05 4.5776367e-05
16      16 -1.0000153 -0.99996948 -0.99999237 -2.2888125e-05 2.2888184e-05
17      17 -1.0000153 -0.99999237 -1.00000381  1.1444106e-05 1.1444092e-05
18      18 -1.0000038 -0.99999237 -0.99999809 -5.7220423e-06 5.7220459e-06
19      19 -1.0000038 -0.99999809 -1.00000095  2.8610239e-06 2.8610229e-06
20      20 -1.0000010 -0.99999809 -0.99999952 -1.4305112e-06 1.4305115e-06
21      21 -1.0000010 -0.99999952 -1.00000024  7.1525579e-07 7.1525574e-07
22      22 -1.0000002 -0.99999952 -0.99999988 -3.5762785e-07 3.5762787e-07
23      23 -1.0000002 -0.99999988 -1.00000006  1.7881394e-07 1.7881393e-07
```

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si

NIP. 199005202024061001

ferdian.bangkit@untirta.ac.id