

Diberikan persamaan berikut:

$$x^2 = 3 + \ln(x)$$

1. Bagaimana metode bagi-dua mencari hampiran akar-akar persamaan tak linier tersebut?

Sebelum menggunakan metode bagi-dua, persamaan akan diubah menjadi bentuk $f(x) = 0$

$$f(x) = 3 + \ln(x) - x^2 = 0$$

Kemudian metode bagi-dua memiliki langkah-langkah sebagai berikut:

- (a) Cari selang (a, b) sehingga $f(a) \cdot f(b) < 0$

Langkah ini dilakukan untuk mencari selang $(a, b) \subset D_f$ sedemikian sehingga $f(x)$ pada selang tersebut melewati titik 0. Terdapat 2 kasus dimana hal ini terjadi

- i. $f(a)$ negatif dan $f(b)$ positif

$$\left. \begin{array}{l} f(a) < 0 \\ f(b) > 0 \end{array} \right\} \Rightarrow f(a) \cdot f(b) < 0$$

- ii. $f(a)$ positif dan $f(b)$ negatif

$$\left. \begin{array}{l} f(a) > 0 \\ f(b) < 0 \end{array} \right\} \Rightarrow f(a) \cdot f(b) < 0$$

Dari kedua kasus, diperoleh $f(a) \cdot f(b) < 0$

- (b) Pilih titik tengah $t \in (a, b)$

Setelah memiliki selang (a, b) , pilih salah satu titik $t \in (a, b)$. Untuk menyederhanakan algoritma, pilih

$$t = \frac{1}{2}(a + b)$$

Kemudian jika

- i. $f(t) = 0$

$$\Rightarrow \text{Solusinya } x = t$$

- ii. $f(t) \cdot f(a) < 0$

$$\Rightarrow \text{Ubah selang } (a, b) \text{ menjadi } (a, t) \text{ dan ulangi dari langkah (a)}$$

$$\text{iii. } f(t) \cdot f(b) < 0$$

\Rightarrow Ubah selang (a, b) menjadi (t, b) dan ulangi dari langkah (a)

Lakukan kedua langkah ini sampai ditemukan solusinya atau sampai tingkat galat yang diinginkan

Perhitungan akar dengan metode ini dilakukan menggunakan bahasa pemrograman "Rust" yaitu sebagai berikut:

```
use std::io;

// INPUT KE VARIABEL FLOATING POINT F64
fn read() -> f64 {
    let mut buffer = String::new();
    io::stdin().read_line(&mut buffer).expect("Failed");
    return buffer.trim().parse().expect("Failed");
}

// FUNGSI F(X)
fn f(x : f64) -> f64 {
    return 3.0 + x.ln() - (x * x);
}

fn main() {

    // INPUT BATAS BAWAH
    println!();
    println!("Batas bawah =");
    let mut a = read();

    // INPUT BATAS ATAS
    println!();
    println!("Batas atas =");
    let mut b = read();

    // VALIDASI INTERVAL
    while f(a) * f(b) >= 0.0{
        println!();
        println!("Batas bawah =");
        a = read();

        println!();
        println!("Batas atas =");
        b = read();
    }

    // INPUT GALAT
    println!();
    println!("Galat =");
    let galat = read();

    // LOOPING ALGORITMA
    let mut c = (a + b) / 2.0;
    while (f(c) != 0.0) & ((a - b).abs() > galat){

        // OUTPUT SEMENTARA
        println!();
```

```

println!();
println!("a = {a}");
println!("b = {b}");
println!("c = {c}");
println!();
println!("f(c) = {}",f(c));
println!("|a - b| = {}",(a - b).abs());
println!();
println!("f(a) * f(c) = {}",f(a)*f(c));
println!("f(b) * f(c) = {}",f(b)*f(c));

// PERBARUAN INTERVAL
if f(a) * f(c) < 0.0 {
    b = c;
}
if f(b) * f(c) < 0.0 {
    a = c;
}

// PERBARUAN NILAI C
c = (a + b) / 2.0;
}

// OUTPUT AKHIR
println!();
println!();
println!("a = {a}");
println!("b = {b}");
println!("c = {c}");
println!();
println!("f(c) = {}",f(c));
println!("|a - b| = {}",(a - b).abs());
println!();
println!("f(a) * f(c) = {}",f(a)*f(c));
println!("f(b) * f(c) = {}",f(b)*f(c));
println!();
println!("x = {c}");
}

```

Untuk $f(x) = 3 + \ln(x) - x^2$, dipilih selang $(0.01, 1)$ sebagai interval awal dengan tingkat galat sebesar 0.00001. Hasil yang didapatkan adalah sebagai berikut:

```

Batas bawah =
0.01

Batas atas =
1

Galat =
0.00001

a = 0.01
b = 1
c = 0.505

f(c) = 2.061778150293223
|a - b| = 0.99

```

```

f(a) * f(c) = -3.3097109947873844
f(b) * f(c) = 4.123556300586446

a = 0.01
b = 0.505
c = 0.2575

f(c) = 1.5769581911216537
|a - b| = 0.495

f(a) * f(c) = -2.5314439687573005
f(b) * f(c) = 3.25133794238055

.

.

.

a = 0.04991058349609375
b = 0.04992568969726563
c = 0.049918136596679694

f(c) = 0.0001372962301794808
|a - b| = 0.000015106201171878608

f(a) * f(c) = -0.0000000018220511677599045
f(b) * f(c) = 0.0000000395194017977414

a = 0.04991058349609375
b = 0.049918136596679694
c = 0.04991436004638672

f(c) = 0.00006201551748219458
|a - b| = 0.0000075531005859427736

f(a) * f(c) = -0.0000000008230047241643406
f(b) * f(c) = 0.000000008514496762935003

x = 0.04991436004638672

```

Diperoleh akar dari $f(x) = 3 + \ln(x) - x^2$ dengan galat sebesar 0.00001 adalah 0.04991153 sehingga dapat disimpulkan solusi dari $x^2 = 3 + \ln(x)$ dengan galat sebesar 0.00001 adalah $x = 0.04991436004638672$

2. Bagaimana metode posisi palsu mencari hampiran akar-akar persamaan tak linier tersebut?

Sebelum melakukan metode posisi palsu, ubah persamaan menjadi bentuk $f(x) = 0$

$$f(x) = 3 + \ln(x) - x^2$$

Kemudian, lakukan langkah-langkah sebagai berikut:

- (a) Tentukan interval (a, b) sehingga $f(a) \cdot f(b) < 0$

Setelah dijelaskan sebelumnya, syarat $f(a) \cdot f(b) < 0$ menjamin bahwa akar dari $f(x)$ berada di dalam interval (a, b)

- (b) Pilih titik $c \in (a, b)$

Pilih salah satu titik c di interval (a, b) . Dalam metode ini, nilai c ditentukan sebagai berikut:

$$c = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

Kemudian periksa apabila

- i. $f(c) = 0$

\Rightarrow Solusinya $x = c$

- ii. $f(a) \cdot f(c) < 0$

\Rightarrow Ubah interval (a, b) menjadi (a, c) dan ulangi langkah

- iii. $f(b) \cdot f(c) < 0$

\Rightarrow Ubah interval (a, b) menjadi (c, b) dan ulangi langkah

Ulangi langkah-lakukan tersebut sampai solusinya ditemukan atau sampai nilai galat dicapai.

Perhitungan akar dengan metode ini dilakukan menggunakan bahasa pemrograman "Rust" yaitu sebagai berikut:

```
use std::io;

// INPUT KE VARIABEL F64
fn read() -> f64 {
    let mut buffer = String::new();
    io::stdin().read_line(&mut buffer).expect("Failed");
    return buffer.trim().parse().expect("Failed");
}

// FUNGSI F(X)
fn f(x : f64) -> f64 {
    return 3.0 + x.ln() - (x * x);
}

fn main() {

    // INPUT BATAS BAWAH
    println!();
    println!("Batas bawah =");
    let mut a = read();

    // INPUT BATAS ATAS
    println!();
    println!("Batas atas =");
    let mut b = read();
```

```

// VALIDASI INTERVAL
while f(a) * f(b) >= 0.0{
    println!();
    println!("Batas bawah =");
    a = read();

    println!();
    println!("Batas atas =");
    b = read();
}

// INPUT GALAT
println!();
println!("Galat =");
let galat = read();

// LOOPING ALGORITMA
let mut c = (a * f(b) - b * f(a))/(f(b) - f(a));
while (f(c) != 0.0) & ((a - b).abs() > galat){

    // OUTPUT SEMENTARA
    println!();
    println!();
    println!("a = {a}");
    println!("b = {b}");
    println!("c = {c}");
    println!();
    println!("f(c) = {}",f(c));
    println!("|a - b| = {}",(a - b).abs());
    println!();
    println!("f(a) * f(c) = {}",f(a)*f(c));
    println!("f(b) * f(c) = {}",f(b)*f(c));

    // PERBARUAN INTERVAL
    if f(a) * f(c) < 0.0 {
        b = c;
    }
    if f(b) * f(c) < 0.0 {
        a = c;
    }

    // PERBARUAN NILAI C
    c = (a * f(b) - b * f(a))/(f(b) - f(a));
}

// OUTPUT AKHIR
println!();
println!();
println!("a = {a}");
println!("b = {b}");
println!("c = {c}");
println!();
println!("f(c) = {}",f(c));
println!("|a - b| = {}",(a - b).abs());
println!();
println!("f(a) * f(c) = {}",f(a)*f(c));
println!("f(b) * f(c) = {}",f(b)*f(c));

```

```

        println!();
        println!("x = {c}");
    }

```

Untuk $f(x) = 3 + \ln(x) - x^2$, dipilih selang $(0.01, 1)$ sebagai interval awal dengan tingkat galat sebesar 0.00001. Hasil yang didapatkan adalah sebagai berikut:

```

Batas bawah =
0.01

Batas atas =
1

Galat =
0.00001

a = 0.01
b = 1
c = 0.45080399030971813

f(c) = 2.0000531170838625
|a - b| = 0.99

f(a) * f(c) = -3.2106256392472727
f(b) * f(c) = 4.000106234167725

a = 0.01
b = 0.45080399030971813
c = 0.20626797488753582

f(c) = 1.3788744159408937
|a - b| = 0.4408039903097181

f(a) * f(c) = -2.2134659901316587
f(b) * f(c) = 2.7578220736697747

.

.

.

a = 0.01
b = 0.04991124917784136
c = 0.04991124917784135

f(c) = -0.000000000000000025673907444456745
|a - b| = 0.039911249177841356

f(a) * f(c) = 0.00000000000000004121355817840411
f(b) * f(c) = -0.0000000000000000000000000000004798875753616133

a = 0.04991124917784135
b = 0.04991124917784136

```

```

c = 0.04991124917784136

f(c) = 0.00000000000000018691645453650096
|a - b| = 0.0000000000000006938893903907228

f(a) * f(c) = -0.0000000000000000000000000000004798875753616133
f(b) * f(c) = 0.0000000000000000000000000000003493776097649583

x = 0.04991124917784136

```

Diperoleh akar dari $f(x) = 3 + \ln(x) - x^2$ dengan galat sebesar 0.00001 adalah 0.04991153 sehingga dapat disimpulkan solusi dari $x^2 = 3 + \ln(x)$ dengan galat sebesar 0.00001 adalah $x = 0.04991124917784136$

3. Dapatkah kalian memperoleh nilai eksak untuk akar-akar riil dari masalah tersebut?

Pada nomor sebelumnya, digunakan galat sebesar 0.00001. Untuk memperoleh nilai eksaknya, nilai galat harus 0. Namun ketika algoritma dijalankan dengan galat sebesar 0, program tidak berhenti untuk waktu yang lama. Apabila program berhenti dan menghasilkan solusi pun tidak menjamin bahwa hasil tersebut adalah nilai eksak karena algoritma diatas menggunakan *floating point f32* yaitu representasi bilangan riil dengan bilangan bulat biner 32 bit yang tidak akurat sepenuhnya.

4. Apa yang dapat kalian simpulkan mengenai kedua metode yang digunakan (metode bagi-dua dan metode posisi palsu)?

Dari kedua metode tersebut, akar persamaan non-linier dapat ditentukan dengan menentukan suatu interval dimana akar persamaan tersebut termuat kemudian memperkecil interval tersebut sampai interval berukuran cukup kecil atau sampai nilai eksaknya ditemukan. Kedua metode ini memiliki perbedaan dari cara interval diperkecil dengan metode bagi-dua membagi interval menjadi 2 interval sama besar yang kemudian ditentukan interval mana yang masih memuat akarnya, dan metode posisi palsu yang membagi interval menjadi 2 sesuai nilai fungsi pada batas interval yang kemudian ditentukan interval mana yang masih memuat akarnya.

Sumber:

1. [Bisection method - Wikipedia](#)
2. [Regula falsi - Wikipedia](#)