

LAPORAN TUGAS
METODE NUMERIK



TI-2F

Farrel Augusta Dinata

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Soal

Membuat sebuah program yang bisa digunakan untuk mencari akar penyelesaian suatu persamaan non linier dengan metode bisection (bagi dua)

Jawab: Berikut adalah tampilan program yang telah saya buat:

```
karl@LAPTOP-V9Q55RPI:/mnt/d/02-Sekolah/01-KULIAH/03-Semester 3/03-Matematika 3/04-Minggu 4/tugas$ ./bisection-method.sh
GUIDANCE! THIS IS IMPORTANT
=====
1. You need to write manually the equation
2. If there is multiplication, put asterisk (*) symbol
3. If there is addition, put plus (+) symbol
4. If there is subtraction, put minus (-) symbol
5. If there is division, put slash (/) symbol
6. IF there is exponentiation, put power (^) symbol

Example:
Math equation: x^2+2x-20
Input: x^2 + 2*x - 20
```

Program ditulis dengan bahasa pemrograman shell. Shell ini bisa mengeksekusi berbagai Unix command sekaligus program yang ditulis di dalam sebuah file (biasa disebut juga shell script). Karena sebenarnya habitat utama dari shell script berada di sistem operasi berbasis Unix maka untuk mengeksekusinya bisa di sistem operasi Unix langsung ataupun menggunakan WSL di Windows.

Alasan saya membuat program berbasis shell ini karena shell cukup jarang saya temui untuk aplikasinya di kehidupan sehari-hari. Padahal saya rasa kemampuan dari shell ini juga mirip dengan bahasa pemrograman populer lainnya seperti Java, JavaScript, dan Python. Tantangan dalam membuat program shell ini adalah beberapa syntax yang cukup berbeda dari bahasa pemrograman pada umumnya sehingga proses pembuatan program memakan waktu lebih lama.

Program yang saya buat memungkinkan untuk menemukan solusi dari akar penyelesaian dari sistem persamaan linear ataupun non linear. Prosesnya dilakukan dengan metode bagi dua atau bisection yang bakal menentukan titik hasilnya berada di area kiri atau kanan. Berikut adalah dua contoh kasus untuk mendapatkan titik akar penyelesaian untuk sistem persamaan linear dan sistem persamaan non linear.

Contoh: $2x-6$

```
karl@LAPTOP-V9Q55RPI:/mnt/d/02-Sekolah/01-KULIAH/03-Semester 3/03-Matematika 3/04-Minggu 4/tugas$ ./bisection-method.sh
GUIDANCE! THIS IS IMPORTANT
=====
1. You need to write manually the equation
2. If there is multiplication, put asterisk (*) symbol
3. If there is addition, put plus (+) symbol
4. If there is subtraction, put minus (-) symbol
5. If there is division, put slash (/) symbol
6. IF there is exponentiation, put power (^) symbol

Example:
Math equation: x^2+2x-20
Input: x^2 + 2*x - 20

Enter your equation: 2*x - 6
Add range:
- Left x (a): 1.5
- Right x (b): 3.5
Max iteration: 10
```

Data yang dimasukkan adalah persamaan yang ingin dipecahkan masalahnya, jangkauan area (x paling kiri dan x paling kanan), dan maksimum iterasi. Setelah itu program akan memproses data-data tersebut dan akan melakukan iterasi hingga mendapatkan nilai yang sesuai.

```
Iteration 1:
a = 1.5
b = 3.5
x = 2.5
f(a) = -3
f(b) = 1
f(x) = -1

Iteration 2:
a = 2.5
b = 3.5
x = 3
f(a) = -1
f(b) = 1
f(x) = 0

=====
Root found for equation 2*x - 6 using the bisection method at x=3
=====
Final result: 3
Iterations: 2
```

Hasil iterasi sebagai gambar di atas. Didapatkan sebuah proses dengan dua kali iterasi. Dari awal ditentukan bahwa maksimum iterasi adalah 10. Namun, karena nilai $f(x) = 0$, maka iterasi akan langsung dihentikan.

Contoh: $x^2 + 2x - 20$

```
karl@LAPTOP-V9Q5SRPI:/mnt/d/02-Sekolah/01-KULIAH/03-Semester 3/03-Matematika 3/04-Minggu 4/tugas$ ./bisection-method.sh
GUIDANCE! THIS IS IMPORTANT
=====
1. You need to write manually the equation
2. If there is multiplication, put asterisk (*) symbol
3. If there is addition, put plus (+) symbol
4. If there is subtraction, put minus (-) symbol
5. If there is division, put slash (/) symbol
6. IF there is exponentiation, put power (^) symbol

Example:
Math equation: x^2+2x-20
Input: x^2 + 2*x - 20

Enter your equation: x^2 + 2*x - 20
Add range:
- Left x (a): -6
- Right x (b): -2
Max iteration: 6
```

Data-data tersebut akan diproses dan dihasilkan iterasi sebagai berikut:

```

Iteration 1:
a = -6
b = -2
x = -4
f(a) = 4
f(b) = -20
f(x) = -12

Iteration 2:
a = -6
b = -4
x = -5
f(a) = 4
f(b) = -12
f(x) = -5

Iteration 3:
a = -6
b = -5
x = -5.5
f(a) = 4
f(b) = -5
f(x) = -0.75

Iteration 4:
a = -6
b = -5.5
x = -5.75
f(a) = 4
f(b) = -0.75
f(x) = 1.5625

```

```

Iteration 5:
a = -5.75
b = -5.5
x = -5.625
f(a) = 1.5625
f(b) = -0.75
f(x) = 0.390625

Iteration 6:
a = -5.625
b = -5.5
x = -5.5625
f(a) = 0.390625
f(b) = -0.75
f(x) = -0.18359375

=====
Root found for equation x^2 + 2*x - 20 using the bisection method at x=-5.5625 and f(x)=-0.18359375
=====
Final result: -5.5625
Iterations: 6

```

Setelah melakukan 6 kali iterasi didapatkan nilai x berada di -5.5625 dengan nilai $f(x) = -0.18359375$. Hasil ini sudah mendekati nol. Jika ingin mendapatkan nilai yang lebih akurat, maka perlu melakukan iterasi tambahan.

Pada sebagian proses aritmatika pada shell menghasilkan nilai integer. Jadi agar nilai berupa float atau desimal maka saya menggunakan awk dan bc agar bisa handle nilai yang bentuknya tidak berupa bilangan bulat. Ini sangat berbeda dari bahasa pemrograman populer

yang mana mereka sudah bisa handle operasi aritmatika dengan input dan output nilai float.

Hasil keseluruhan kode saya upload ke GitHub pada repository berikut:

<https://github.com/FarrelAD/Math-Programming/tree/main/numerical-method/bisection-method>