

Penyelesaian Sistem Persamaan Non-Linier Dengan Metode Bisection & Metode Regula Falsi Menggunakan Bahasa Program Java

Endang Sunandar

Jurusan Sistem Komputer Universitas Raharja
Jalan Jendral Sudirman No 40 Modernland Cikokol Tangerang
endang.sunandar@raharja.info

ABSTRACT

The numerical method is a technique used to formulate mathematical problems so that they can be solved using ordinary arithmetic operations. In general, numerical methods are used to solve mathematical problems that cannot be solved by ordinary analytical methods. In the Numerical Method we know two types of system equations, namely the Linear Equation System and the Non-Linear Equation System. Each system of equations has several methods. In the System the Linear Equation between the methods is the Gauss Elimination method, the Gauss-Jordan Elimination method, the LU (Lower-Upper) Decomposition method. And for the Non-Linear Equation System between the methods is the Bisection method (Share-Two), Falsi Regula method, Newton Raphson method, Secant method, and Fix Iteration method. In this study, researchers are interested in comparing the two methods in the Non-Linear Equation System, namely the Bisection method and the Falsi Regula method. And this benchmarking process uses the Java programming language tool, this is to facilitate analysis of method completion algorithms, and monitoring in terms of execution time and output analysis. So that we can clearly know what differences occur between the two methods.

Keywords: *Bisecton Method, Falsi Regulatory Method*

ABSTRAK

Metode Numerik merupakan suatu teknik yang digunakan untuk memformulasikan persoalan matematika sehingga dapat diselesaikan dengan menggunakan operasi aritmatika biasa. Pada umumnya metode numerik digunakan untuk menyelesaikan persoalan matematika yang tidak dapat diselesaikan dengan metode analitik biasa. Dalam Metode Numerik kita mengenal dua buah jenis sistem persamaan yaitu Sistem Persamaan Linier dan Sistem Persamaan Non-Linier. Masing-masing sistem persamaan memiliki beberapa metode. Dalam Sistem Persamaan Linier diantara metodenya adalah metode Eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode Dekomposisi LU (Lower-Upper). Dan untuk Sistem Persamaan Non-Linier diantara metodenya adalah metode Bisection (Bagi-Dua), metode Regula Falsi, metode Newton Raphson, metode Secant, dan metode Fix Iteration. Dalam penelitian ini, peneliti tertarik untuk membandingkan 2 buah metode yang ada dalam Sistem Persamaan Non-Linier, yaitu metode Bisection (Bagi-Dua) dan metode Regula Falsi. Dan proses perbandingan ini menggunakan perangkat bahasa pemrograman Java, hal ini untuk memudahkan dalam hal analisis algoritma penyelesaian metode, dan monitoring dalam hal waktu eksekusi dan analisis hasil output. Sehingga kita dapat secara jelas mengetahui perbedaan apa saja yang terjadi diantara ke-dua buah metode tersebut.

Kata kunci: *Metode Bisecton, Metode Regula Falsi*

1. PENDAHULUAN

Seperti kita ketahui bersama bahwa Sistem Persamaan Non-Linier adalah suatu Sistem Persamaan yang digunakan untuk menghitung Akar Persamaan Non-Linier menggunakan satu variable X , $f(x)$, atau secara umum dituliskan dengan formula: $f(x)=0$.

Beberapa metode yang ada diantaranya adalah metode Bisection, metode Regula Falsi, metode Newton Raphson, metode Secant. Dan dalam penelitian ini peneliti menggunakan 2 buah metode yaitu Bisection dan Regula Falsi. Dengan pertimbangan bahwa ke-2 metode ini adalah metode dasar yang digunakan dalam penyelesaian Sistem Persamaan Non-Linier.

Hasil yang diharapkan dari penelitian ini adalah untuk mengetahui perbedaan penyelesaian Sistem Persamaan Non-Linier dengan metode Bisection dan metode Regula Falsi menggunakan bahasa Java.

Penelitian sebelumnya yang pernah ada adalah “Aplikasi Analisis Tingkat Akurasi Penyelesaian Persamaan Non-Linier Dengan Metode Biseksi dan Metode Newton Raphson”[1]. Dalam penelitian tersebut digambarkan tentang hasil analisis bahwasanya waktu eksekusi metode Biseksi lebih lambat dibandingkan dengan metode Newton Raphson, dikarenakan dalam metode Biseksi selalu mencari / menentukan titik tengah. Dalam penelitian “Perhitungan Luas Penulangan Metode Bisection Pada Kolom Biaxial Menggunakan Delphi Dengan Rumus Bresler” menyatakan bahwa metode Bisection cocok diterapkan pada penentuan panjang luas tulangan karena metode ini terjadi konvergen ke angka yang dicari, dan metode ini termasuk metode tertutup [2]. Sedangkan dalam penelitian yang berjudul ”Solusi Numerik Persamaan Non Linier dengan Metode Bisection dan Regula Falsi” menyampaikan bahwa untuk menemukan akar persamaan positif dan negative secara berturut-turut dengan menggunakan metode Bisection memerlukan 19 kali iterasi, sedangkan dengan menggunakan metode Regula Falsi memerlukan 8 dan 12 kali iterasi [3].

2. METODE / PERANCANGAN PENELITIAN

Pada penelitian ini, peneliti menggunakan 2 buah metode yaitu **Bisection** dan **Regula Falsi**. Masing-masing metode memiliki formulasi dalam penyelesaiannya. Sedangkan untuk algoritma penyelesaian dari ke-dua metode ini adalah sama. Dapat dikatakan bahwa ke-dua metoda ini memiliki algoritma penyelesaian yang sama, hanya formulasinya yang berbeda (formulasi nilai tengahnya). Dan untuk implementasi dari masing-masing formulasi tersebut peneliti menggunakan bahasa program JAVA.

2.1. Metode Bisection

Metode ini disebut juga dengan metode bagi-dua, karena untuk mencari komponen nilai tengahnya (X_{mid}) menggunakan suatu rumus yaitu, suatu nilai x ditambahkan dengan nilai x didepannya, kemudian dibagi 2. Expresi formulasinya adalah sebagai berikut:

$$X_{mid} = (X_n + X_{n+1}) / 2 \quad (1)$$

Sedangkan untuk algoritma penyelesaian dari metode Bisection adalah sebagai berikut :

1. Tentukan sistem persamaan non-liniernya.
2. Tentukan nilai awal untuk X_n dan X_{n+1} , dan nilai toleransi-error nya.
3. Hitung nilai X_{mid} sesuai rumus diatas yaitu $X_{mid} = (X_n + X_{n+1}) / 2$
4. Hitung nilai $f(X_n)$, $f(X_{n+1})$, dan $f(X_{mid})$ nya, dengan cara memasukan nilai-nilai variabel tersebut ke dalam sistem persamaan non-liniernya.

5. Tentukan nilai absolut dari $f(X_{mid})$ nya (mengkonversikan menjadi bilangan positif).
6. Cek perbandingan antara nilai absolut dari $f(X_{mid})$ dengan nilai toleransi-error, jika:
 - a. Nilai absolut dari $f(X_{mid}) >$ dari nilai toleransi-error maka proses berlanjut ke langkah nomor 7.
 - b. Nilai absolut dari $f(X_{mid}) <$ dari nilai toleransi-error maka proses berhenti, dan yang menjadi jawaban dari nilai yang dicari adalah nilai yang ada pada variabel X_{mid} (nilai akar persamaan yang dicari).
7. Cek tanda (positif atau negatif) yang ada di $f(X_n)$ dengan tanda yang ada di $f(X_{mid})$, jika:
 - a. Tanda Sama, maka “nilai X_{mid} “ ditempatkan pada ”variabel X_n langkah berikutnya”, dan “nilai X_{n+1} langkah berikutnya” berasal dari “nilai X_{n+1} langkah sebelumnya”.
 - b. Tanda Berbeda, maka “nilai X_{mid} “ ditempatkan pada ”variabel X_{n+1} langkah berikutnya”, dan “nilai X_n langkah berikutnya” berasal dari “nilai X_n langkah sebelumnya”.
 - c. Selanjutnya proses berulang lagi ke langkah di nomor 3.

2.2. Metode Regula Falsi

Menurut teori bahwasanya metode ini adalah perbaikan dari metode sebelumnya yaitu metode Bisection. Apa bentuk perbaikannya nanti bisa dilihat di Bab 3 Hasil dan Pembahasan. Hanya yang membedakan dengan metode Bisection adalah pada rumus untuk mencari komponen nilai tengahnya (X_{mid}) saja. Expresi formulasinya adalah sebagai berikut:

$$X_{mid} = X_n - f(X_n) * [(X_{n+1} - X_n) / (f(X_{n+1}) - f(X_n))] \quad (2)$$

Sedangkan untuk algoritma penyelesaian dari metode Regula Falsi adalah sama dengan algoritma penyelesaian metode Bisection, yaitu sebagai berikut:

1. Tentukan sistem persamaan non-liniernya.
2. Tentukan nilai awal untuk X_n dan X_{n+1} , dan nilai toleransi-error nya.
3. Hitung nilai X_{mid} sesuai rumus diatas yaitu $X_{mid} = (X_n + X_{n+1}) / 2$
4. Hitung nilai $f(X_n)$, $f(X_{n+1})$, dan $f(X_{mid})$ nya, dengan cara memasukan nilai-nilai variabel tersebut ke dalam sistem persamaan non-liniernya.
5. Tentukan nilai absolut dari $f(X_{mid})$ nya (mengkonversikan menjadi bilangan positif).
6. Cek perbandingan antara nilai absolut dari $f(X_{mid})$ dengan nilai toleransi-error, jika:
 - a. Nilai absolut dari $f(X_{mid}) >$ dari nilai toleransi-error maka proses berlanjut ke langkah nomor 7.
 - b. Nilai absolut dari $f(X_{mid}) <$ dari nilai toleransi-error maka proses berhenti, dan yang menjadi jawaban dari nilai yang dicari adalah nilai yang ada pada variabel X_{mid} (nilai akar persamaan yang dicari).
7. Cek tanda (positif atau negatif) yang ada di $f(X_n)$ dengan tanda yang ada di $f(X_{mid})$, jika:
 - a. Tanda Sama, maka “nilai X_{mid} “ ditempatkan pada ”variabel X_n langkah berikutnya”, dan “nilai X_{n+1} langkah berikutnya” berasal dari “nilai X_{n+1} langkah sebelumnya”.
 - b. Tanda Berbeda, maka “nilai X_{mid} “ ditempatkan pada ”variabel X_{n+1} langkah berikutnya”, dan “nilai X_n langkah berikutnya” berasal dari “nilai X_n langkah sebelumnya”.

c. Selanjutnya proses berulang lagi ke langkah di nomor 3.

3. HASIL DAN PEMBAHASAN

Untuk mengimplementasikan penelitian ini, peneliti menggunakan bahasa program Java baik untuk metode Bisection maupun metode Regula Falsi. Masing-masing programnya adalah sebagai berikut:

3.1. Metode Bisection

- **Program Java untuk Metode Bisection**

```
/* @author Ir.Endang Sunandar,M.Kom
*/
import java.text.DecimalFormat;
public class metodeBisection {
    public static void main(String args[]){
        double xn=2.5;double xnNext=2.6;double xMid=0.0;
        double toleransi=0.005;double absFxmMid=1500; int no=0;
        double fXn=0.0; double fxnNext=0.0;double fXmid=0.0;
        // proses looping metode bisection
        System.out.println("Penyelesaian SPL dengan Metode BISECTION");
        System.out.println("-----");
        System.out.println("Persamaan : X^3 - 7X + 1");
        System.out.println("Nilai Xn=2.5 dan nilai Xn+1=2.6, dengan Toleransi-Error=0.005");
        System.out.println("-----");
        System.out.println("| No| Xn | Xn+1 | f(Xn) |f(Xn+1) | X(mid) | f(Xmid) | |f(Xmid)||");
        System.out.print("-----");
        while (absFxmMid > toleransi){
            DecimalFormat digit1=new DecimalFormat("0.000");
            xMid=(xn+xnNext)/2;
            //mencari nilai fungsi Xn
            fXn=(xn*xn*xn)-(7*xn)+1;
            //mencari nilai fungsi Xn+1
            fxnNext=(xnNext*xnNext*xnNext) - (7 * xnNext) +1;
            //mencari nilai fungsi xMid
            fXmid=(xMid*xMid*xMid) - (7*xMid) + 1;
            no++;
            System.out.println("");
            System.out.print("| "+no+" | ");
            System.out.print(digit1.format(xn)+" | ");
            System.out.print(digit1.format(xnNext)+" | ");
            System.out.print(digit1.format(fXn)+" | ");
            System.out.print(digit1.format(fxnNext)+" | ");
            System.out.print(digit1.format(xMid)+" | ");
            System.out.print(digit1.format(fXmid)+" | ");
            if ((fXn<0 && fXmid<0) || (fXn>0 && fXmid>0)){
                xn=xMid;
            } else {
```

```

        xnNext=xMid;
    }
    if (fXmid<0){
        absFxmId=fXmid * -1;
    }else {
        absFxmId=fXmid;
    } System.out.print(digit1.format(absFxmId)+" | ");

    }System.out.println();
    System.out.print("-----");
} }

```

• **Output Program Java untuk Metode Bisection**

Penyelesaian SPL dengan Metode BISECTION

Persamaan : $X^3 - 7X + 1$
 Nilai $X_n=2.5$ dan nilai $X_{n+1}=2.6$, dengan Toleransi-Error=0.005

No	X_n	X_{n+1}	$f(X_n)$	$f(X_{n+1})$	$X(\text{mid})$	$f(X_{\text{mid}})$	$ f(X_{\text{mid}}) $
1	2,500	2,600	-0,875	0,376	2,550	-0,269	0,269
2	2,550	2,600	-0,269	0,376	2,575	0,049	0,049
3	2,550	2,575	-0,269	0,049	2,562	-0,111	0,111
4	2,562	2,575	-0,111	0,049	2,569	-0,031	0,031
5	2,569	2,575	-0,031	0,049	2,572	0,009	0,009
6	2,569	2,572	-0,031	0,009	2,570	-0,011	0,011
7	2,570	2,572	-0,011	0,009	2,571	-0,001	0,001

Terlihat bahwasanya dengan menggunakan metode Bisection, jumlah langkah penyelesaian untuk mencari akar persamaannya adalah sebanyak 7 langkah. Hal ini dikarenakan nilai absolut $f(X_{\text{mid}})$ yang ada pada langkah ke-7 bernilai 0,001, dimana nilai ini lebih kecil dari nilai toleransi-error yang sebesar 0.005. Dengan demikian nilai yang didapatkan adalah nilai yang berada pada variabel X_{mid} yaitu sebesar 2,571 sebagai akar persamaan dari sistem persamaan non-linier $f(x)= X^3 -7X +1$. Hal ini terlihat logis bahwasanya nilai 2,571 berada pada rentang nilai 2,5 dan 2,6.

3.2. Metode Regula Falsi

• **Program Java untuk Metode Regula Falsi**

```

/* @author Ir.Endang Sunandar,M.Kom
*/
import java.text.DecimalFormat;
public class metodeRegulaFalsi {
    public static void main(String args[]){
        double xn=2.5;double xnNext=2.6;double xMid=0.0;

```

```
double toleransi=0.005;double absFxmMid=1500; int no=0;
double fXn=0.0; double fxnNext=0.0;double fXmid=0.0;
double upper;double lower;
// proses looping metode bisection
System.out.println("Penyelesaian SPL dengan Metode REGULA FALSI");
System.out.println("-----");
System.out.println("Persamaan : X^3 - 7X + 1");
System.out.println("Nilai Xn=2.5 dan nilai Xn+1=2.6, dengan Toleransi-Error=0.005");
System.out.println("-----");
System.out.println("| No| Xn | Xn+1 | f(Xn) |f(Xn+1) | X(mid) | f(Xmid) | |f(Xmid)||");
System.out.print("-----");
while (absFxmMid > toleransi){
    DecimalFormat digit1=new DecimalFormat("0.000");
    //mencari nilai fungsi Xn
    fXn=(xn*xn*xn)-(7*xn)+1;
    //mencari nilai fungsi Xn+1
    fxnNext=(xnNext*xnNext*xnNext) - (7 * xnNext) +1;
    upper=(xnNext - xn);
    lower=(fxnNext - fXn);
    xMid=xn-(fXn*(upper/lower));
    //mencari nilai fungsi xMid
    fXmid=(xMid*xMid*xMid) - (7*xMid) + 1;
    no++;
    System.out.println("");
    System.out.print("| "+no+" | ");
    System.out.print(digit1.format(xn)+" | ");
    System.out.print(digit1.format(xnNext)+" | ");
    System.out.print(digit1.format(fXn)+" | ");
    System.out.print(digit1.format(fxnNext)+" | ");
    System.out.print(digit1.format(xMid)+" | ");
    System.out.print(digit1.format(fXmid)+" | ");
    if ((fXn<0 && fXmid<0) || (fXn>0 && fXmid>0)){
        xn=xMid;
    } else {
        xnNext=xMid;
    }
    if (fXmid<0){
        absFxmMid=fXmid * -1;
    }else {
        absFxmMid=fXmid;
    } System.out.print(digit1.format(absFxmMid)+" | ");
}System.out.println();
System.out.print("-----");
} }
```

• **Output Program Java untuk Metode Regula Falsi**

Penyelesaian SPL dengan Metode REGULA FALSI

Persamaan : $X^3 - 7X + 1$

Nilai $X_n=2.5$ dan nilai $X_{n+1}=2.6$, dengan Toleransi-Error=0.005

No	X_n	X_{n+1}	$f(X_n)$	$f(X_{n+1})$	$X(\text{mid})$	$f(X_{\text{mid}})$	$ f(X_{\text{mid}}) $
1	2,500	2,600	-0,875	0,376	2,570	-0,016	0,016
2	2,570	2,600	-0,016	0,376	2,571	-0,000	0,000

Terlihat bahwasanya dengan menggunakan metode Regula Falsi, jumlah langkah penyelesaian untuk mencari akar persamaannya adalah hanya 2 langkah. Hal ini dikarenakan nilai absolut $f(X_{\text{mid}})$ yang ada pada langkah ke-2 bernilai 0,000, dimana nilai ini lebih kecil dari nilai toleransi-error yang sebesar 0.005. Dengan demikian nilai yang didapatkan adalah nilai yang berada pada variabel X_{mid} yaitu sebesar 2,571 sebagai akar persamaan dari sistem persamaan non-linier $f(x)=X^3 -7X +1$. Hal ini terlihat logis bahwasanya nilai 2,571 berada pada rentang nilai 2,5 dan 2,6.

Dengan demikian kita dapat melihat bahwasanya perbandingan antara metode Bisection dengan metoda Regula Falsi adalah bahwa dengan menggunakan metode Regula Falsi, langkahnya lebih ringkas (2 langkah) dibandingkan dengan metode Bisection (7 langkah), untuk mendapatkan akar persamaan yang dicari. Hal ini secara logika dapat dilihat yaitu bahwa formulasi nilai tengah (X_{mid}) di metode Regula Falsi lebih kompleks dibandingkan dengan formulasi nilai tengah (X_{mid}) di metode Bisection, akan tetapi dengan kekompleksitasan formulasi itu memberikan waktu eksekusi yang cepat berupa jumlah langkah yang lebih ringkas.

KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan pada perbandingan 2 metode, yaitu metode Bisection dan metode Regula Falsi maka dapat diambil beberapa kesimpulan sebagai berikut :

Bahwa metode Bisection memiliki formulasi nilai tengah yang sangat sederhana, tetapi memiliki waktu eksekusi yang lebih lama dibandingkan dengan metode Regula Falsi.

Sedangkan metode Regulasi Falsi memiliki formulasi nilai tengah yang kompleks dan tidak sederhana, tetapi memiliki waktu eksekusi yang lebih cepat dibandingkan dengan metode Bisection.

Untuk pengembangan penelitian lebih lanjut maka dapat menggunakan beberapa metode sistem persamaan non-linier yang lainnya seperti metode Newton Raphson, metode Secant, dan metode Fix Iteration.

Dan penentuan sistem persamaan liniernya pun menggunakan metode interaktif berdasarkan inputan user di program aplikasi, sehingga program aplikasi bersifat dinamis.

DAFTAR PUSTAKA

[1] Tentua Meilany Nonsi, [2017],”Aplikasi Analisis Tingkat Akurasi Penyelesaian Sistem Persamaan Non-Linier dengan metode Bisection dan metode Newton Raphson”, Jalan PGRI Sonosewu no 117 Yogyakarta, Universitas PGRI Yogyakarta Press.

- [2] Kamaludin, [2017], "*Perhitungan Luas Penulangan Metode Bisection pada Kolom Biaxial Menggunakan Delphi dengan Rumus Bresler*", Jalan PHH Mustofa 23 Bandung, Program Studi Teknik Sipil ITENAS
- [3] Jatining Wigati, [2017], "*Solusi Numerik Persamaan Non Linier dengan Metode Bisection dan Regula Falsi*", Jalan Raya Mojosari 02 Kepanjen Malang, Jurnal G-Tech Fakultas Sains dan Teknologi Universitas Islam Raden Rahmat Malang.