

# METODE HOMOTOPI UNTUK MENYELESAIKAN SISTEM PERSAMAAN TAK LINEAR

## (HOMOTOPY METHOD FOR SOLVING NONLINEAR EQUATIONS SYSTEM)

Anie Lusiani, M.Si  
( Staf Pengajar UP. MKU Politeknik Negeri Bandung )

### ABSTRAK

Satu metode penyelesaian sistem persamaan tak linear secara numerik yang cukup banyak digunakan adalah Metode Newton. Dalam tulisan ini dibahas Metode Homotopi yang memiliki beberapa kelebihan dibandingkan dengan Metode Newton. Pendekatan Metode Homotopi yang dibahas di sini adalah pendekatan prediktor-korektor, yaitu membangun lintasan penyelesaian secara kontinu melalui vektor singgung di titik korektor terakhir.

**Kata Kunci:** Sistem persamaan tak linear, Metode Homotopi, lintasan penyelesaian, prediktor-korektor.

### ABSTRACT

One of the methods for solving nonlinear equations system numerically is Newton Method. In general, Newton Method cannot be guaranteed to converge to a solution of system, if it is not started close to that solution. Homotopy Method has more likely to converge to such a solution, if we can construct a zero path continuously. In this work, we discuss Homotopy Method by using predictor-corrector approach.

**Key Words:** Nonlinear equations system, Homotopy Method, zero path, predictor-corrector.

### PENDAHULUAN

Penelitian untuk menentukan penyelesaian sistem persamaan tak linear secara numerik telah dilakukan dengan menggunakan beberapa metode. Metode yang paling banyak digunakan adalah Metode Newton. Dalam Mathews<sup>[2]</sup>, metode ini sangat bergantung pada pemilihan nilai awal, karena nilai awal sangat mempengaruhi konvergensi penyelesaian sistem. Jika pemilihan nilai awal ini tidak tepat maka ada dua kemungkinan, pertama, penyelesaian diperoleh melalui iterasi yang sangat banyak, kedua, penyelesaian tidak diperoleh

karena matriks Jacobi dari sistem yang diberikan ternyata singular di tengah-tengah iterasi.

Dalam penelitian ini dipelajari suatu metode penyelesaian sistem persamaan tak linear yang dapat mengurangi ketergantungan terhadap pemilihan nilai awal dan singularitas matriks Jacobi. Metode ini disebut Metode Homotopi (*Homotopy Method*) atau Metode Kontinuasi (*Continuation Method*), karena digunakan kontinuitas kurva dalam prosesnya. Harapan

lain dari metode ini adalah dapat menentukan lebih dari satu penyelesaian atau akar dari sistem persamaan tak linear dengan satu kali perintah dalam program numerik.

Dalam Nocedal<sup>[3]</sup> dijelaskan bahwa di bagian akhir algoritma Metode Homotopi digunakan Metode Newton untuk menentukan titik korektor yang disebut sebagai **Newton Corrector Process**. Kemudian dalam simulasi, Metode Newton tetap digunakan sebagai pembanding dari sisi akurasi hasil akhir proses. Jadi, Metode Newton tetap menjadi fokus dalam penelitian ini.

## METODE HOMOTOPI DENGAN TITIK TETAP

Sebagai pengkhususan masalah dari sistem  $f(x) = 0$  dengan  $x, 0 \in \mathbb{R}^n$ , didefinisikan pemetaan Homotopi  $H(x, \lambda)$  dengan titik tetap atau **Fixed Point Homotopy**, yaitu:

$$H(x, \lambda) = \lambda f(x) + (1 - \lambda)(x - a), \quad (1)$$

dengan  $\lambda$  parameter skalar dan  $a \in \mathbb{R}^n$  merupakan vektor tetap atau titik tetap.

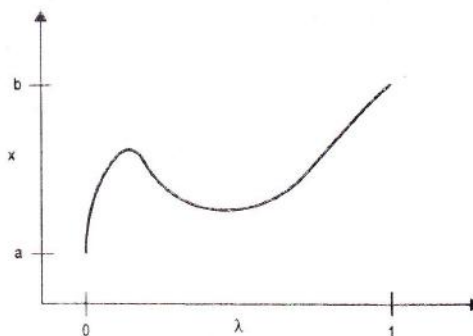
Perhatikan persamaan,

$$H(x, \lambda) = 0 \quad (2)$$

yang merupakan pemetaan dari  $\mathbb{R}^{n+1}$  ke  $\mathbb{R}^n$ , juga merupakan kurva di  $\mathbb{R}^{n+1}$  dengan parameter  $\lambda$ . Jika dipilih  $\lambda = 0$ , maka dari (1) dan (2) diperoleh  $H(x, 0) = x - a = 0$ , yang memberikan penyelesaian  $x = a$ . Jika  $\lambda = 1$ , maka diperoleh sistem semula, yaitu  $H(x, 1) = f(x) = 0$ . Jika penyelesaian sistem ini  $b$ , maka persamaan (2) merupakan kurva di  $\mathbb{R}^{n+1}$  yang melalui titik  $(a, 0)$  dan  $(b, 1)$ . Jadi, menjejak kurva ini di sistem koordinat  $(x, \lambda)$  pada interval  $\lambda \in [0, 1]$  sama dengan mencari penyelesaian sistem  $f(x) = 0$ .

Untuk lebih jelas, perhatikan Gambar 1 yang merupakan kurva  $H(x, \lambda) = 0$  untuk  $x \in \mathbb{R}$ .

Ketika  $\lambda = 0$ , persamaan  $H(x, \lambda) = 0$  memberikan nilai  $x = a$ . Jika nilai  $\lambda$  ditambah sedikit, misalnya sebesar  $\delta$ , maka  $H(x, \lambda) = 0$  memberikan nilai  $x$  yang bersesuaian dengan nilai  $\lambda = \delta$ . Demikian seterusnya, dilakukan penambahan pada nilai  $\lambda$  ini secara bertahap sampai pada nilai  $\lambda = 1$ . Ketika nilai  $\lambda = 1$ , maka nilai  $x$  yang memenuhi  $H(x, \lambda) = 0$ , yaitu  $b$  merupakan penyelesaian sistem  $f(x) = 0$ , karena  $H(x, 1) = f(x) = 0$ .



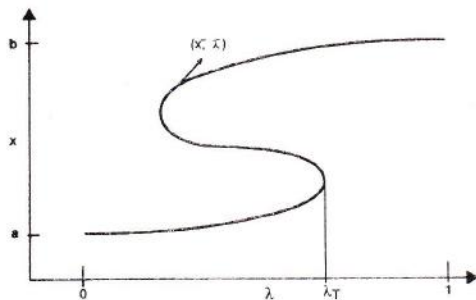
Gambar 1: Sebuah lintasan penyelesaian: trayektori titik-titik  $(x, \lambda)$  yang memenuhi  $H(x, \lambda) = 0$

Trayektori dari titik-titik pada kurva yang memenuhi persamaan (2) disebut **zero path** atau **lintasan penyelesaian**.

## IMPLEMENTASI METODE HOMOTOPI

Penjejakan lintasan penyelesaian dengan pendekatan di atas akan berhasil pada kurva di Gambar 1, tetapi tidak dapat dilakukan pada kurva di Gambar 2, karena kurva ini mengandung **turning point**,  $\lambda_r$ , sehingga penjejakan akan berhenti di titik ini. Setelah itu, ada proses pengurangan nilai  $\lambda$  selama diperlukan. Pendekatan lain diperlukan untuk menjejak lintasan pada Gambar 2. Untuk hal ini akan dimulai dengan menjadikan  $x$  dan  $\lambda$  sebagai fungsi dari

peubah bebas  $s$ , yang merepresentasikan panjang busur lintasan penyelesaian.



Gambar 2: Sebuah lintasan penyelesaian dengan *turning point*.

Jadi,  $H(x(s), \lambda(s)) = 0$  untuk setiap  $s \geq 0$ . Diperoleh turunan total  $H$  terhadap  $s$ , yaitu

$$\frac{\partial}{\partial x} H(x, \lambda) \dot{x} + \frac{\partial}{\partial \lambda} H(x, \lambda) \dot{\lambda} = 0 \quad (3)$$

dengan  $(\dot{x}, \dot{\lambda}) = (\frac{dx}{ds}, \frac{d\lambda}{ds})$  merupakan vektor singgung lintasan penyelesaian pada suatu nilai  $s$ , seperti ilustrasi pada Gambar 2. Dari persamaan (3) dapat dilihat bahwa vektor ini berada dalam kernel matriks yang berukuran  $n \times (n+1)$  berikutini.

$$\begin{bmatrix} \frac{\partial}{\partial x} H(x, \lambda) & \frac{\partial}{\partial \lambda} H(x, \lambda) \end{bmatrix} \quad (4)$$

Jika matriks ini memiliki *rank* penuh, maka kernelnya berdimensi satu. Artinya, vektor-vektor singgung sebagai penyelesaian sistem (3) hanya dibangun oleh satu vektor saja. Hal ini sesuai dengan karakteristik vektor singgung pada kurva di suatu titik. Oleh karena itu vektor singgung ini akan terdefinisi dengan baik jika matriks di atas mempunyai *rank* penuh.

Berkaitan dengan lintasan penyelesaian yang akan dibangun, maka arah vektor singgung ini harus diatur atau dipilih sedemikian sehingga lintasan penyelesaian diperoleh secara utuh pada selang yang diberikan. Pemilihan ini dimaksudkan agar vektor singgung yang diperoleh memiliki arah

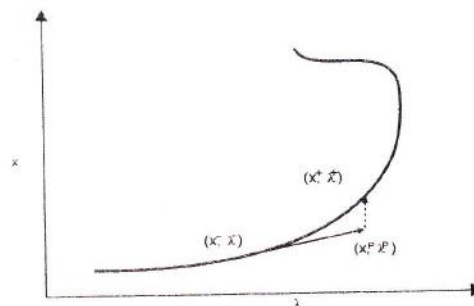
sedemikian sehingga selalu menapak maju lintasan penyelesaian.

Adapun kriteria pemilihan ini adalah memberikan tanda positif atau negatif pada vektor singgung hasil penghitungan, agar sudut antara vektor singgung ini pada suatu nilai  $s$  dengan vektor singgung pada nilai  $s$  sebelumnya terletak pada interval  $[0, \frac{\pi}{2}]$ . Adapun panjang vektor singgung ini adalah satu, karena panjang kurva lintasan dijadikan sebagai parameter. Setelah diperoleh vektor singgung di sembarang titik  $(x, \lambda)$ , kemudian dicari penyelesaian  $x$  yang bersesuaian dengan nilai  $\lambda = 1$ . Dengan kata lain, lintasan penyelesaian dibangun dengan pendekatan aljabar. Metode ini disebut juga pendekatan **Predictor-Corrector**. Uraianya sebagai berikut.

Pertama, ambil sembarang nilai awal,  $(a, 0)$ , kemudian hitung vektor singgungnya. Ambil satu langkah kecil, sebesar sigma, sepanjang arah vektor singgung ini untuk mendapatkan sebuah titik prediktor  $(x'', \lambda'')$ , yaitu

$$(x'', \lambda'') = (x, \lambda) + \sigma(\dot{x}, \dot{\lambda}) \quad (5)$$

Biasanya titik prediktor ini tidak terletak pada lintasan penyelesaian. Oleh karena itu dibutuhkan iterasi korektor untuk membawa titik itu kembali pada lintasannya. Hasil koreksi ini ditulis sebagai  $(x^+, \lambda^+)$ , dengan syarat  $H(x^+, \lambda^+) = 0$ . Proses ini dapat dilihat pada Gambar 3.



Gambar 3: Prosedur **prediktor-korektor**, menggunakan  $\lambda$  sebagai komponen tetap dalam proses koreksi

Selama proses koreksi dipilih sebuah komponen tetap dari titik prediktor yaitu sebuah komponen yang perubahannya paling cepat pada langkah terakhir. Karena titik prediktor cukup dekat dengan titik target  $(x^*, \lambda^*)$ , maka digunakan Proses Koreksi Newton (*Newton Corrector Process*). Hal ini tidak akan mengakibatkan kegagalan dalam memperoleh penyelesaian, disebabkan oleh cukup dekatnya nilai awal dari penyelesaian. Jika indeks dari komponen ini  $i$ , maka langkah berikutnya seperti pada penurunan Metode Newton.

Karena  $H$  fungsi dari  $\mathfrak{R}^{n+1}$  ke  $\mathfrak{R}^n$ , maka dapat ditulis sebagai

$$H(y) = \begin{pmatrix} H_1(y) \\ H_2(y) \\ \vdots \\ H_n(y) \end{pmatrix},$$

dengan  $y = (x, \lambda) = (x_1, x_2, \dots, x_n, \lambda)^T$ . Deret Taylor untuk  $H_i(y)$  di  $y = y_0$  adalah

$$H_i(y) = H_i(y_0) + (x_1 - x_{1_0}) \frac{\partial H_i}{\partial x_1}(y_0) + \dots + (x_n - x_{n_0}) \frac{\partial H_i}{\partial x_n}(y_0) + (\lambda - \lambda_0) \frac{\partial H_i}{\partial \lambda}(y_0) + \dots$$

untuk  $i = 1, 2, \dots, n$ .

Dengan mengabaikan orde kedua deret Taylor di atas dan memisalkan  $y$  sebagai akar dari  $H$ , maka diperoleh sistem

$$-H = \left( \frac{\partial H}{\partial x} \mid \frac{\partial H}{\partial \lambda} \right) \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix}.$$

Berkaitan dengan komponen tetap, maka sistem menjadi

$$\begin{pmatrix} \frac{\partial H}{\partial x} \mid \frac{\partial H}{\partial \lambda} \\ e_i \end{pmatrix} \begin{pmatrix} \delta x \\ \delta \lambda \end{pmatrix} = \begin{pmatrix} -H \\ 0 \end{pmatrix}, \quad (6)$$

dengan  $\partial H / \partial x, \partial H / \partial \lambda$ , dan  $H$  dihitung di titik prediktor terakhir. Pada baris terakhir di dalam sistem (6), ditentukan salah satu dari komponen  $(\delta x, \delta \lambda)$  yang bernilai nol. Vektor  $e_i$  berada di  $\mathfrak{R}^{n+1}$  dan semua komponennya nol kecuali komponen ke- $i$

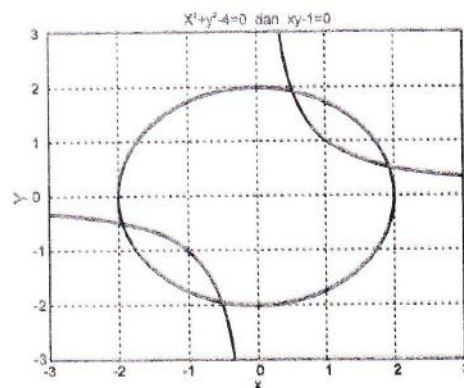
bernilai satu (bersesuaian dengan komponen tetap yang dimaksud di atas). Pada Gambar 3, komponen  $\lambda$  berperan sebagai komponen tetap pada iterasi yang bersangkutan. Pada iterasi selanjutnya, mungkin saja terpilih salah satu komponen  $x$  sebagai komponen tetap bila ditemukan *turning point* pada lintasan penyelesaian.

## SIMULASI

Ada sedikit perbedaan algoritma Metode Homotopi untuk menyelesaikan fungsi satu peubah dan untuk sistem persamaan, yaitu pada fungsi satu peubah tidak diperlukan faktorisasi QR pada saat penghitungan vektor singgung.

Contoh 1: Diberikan sistem persamaan taklinear dengan dua peubah bebas  $x$  dan  $y$ , yaitu

$$\begin{aligned} x^2 + y^2 - 4 &= 0 \\ xy - 1 &= 0 \end{aligned} \quad (7)$$



Gambar 4: Kurva Sistem (7) pada bidang  $xy$ .

Dari Gambar 4, diperoleh empat akar dengan pendekatan satu angka desimal, yaitu:  $(1,9;0,5)$ ,  $(0,5;1,9)$ ,  $(-1,9;-0,5)$ , dan  $(-0,5;-1,9)$ . Dengan menjalankan algoritma Metode Homotopi yang titik awalnya  $(1, 1, 0)$  dan sigma  $0,01$  diperoleh keluaran:

Tabel 1: Keluaran Metode Homotopi untuk sistem (7) dengan titik awal (1, 1, 0) dan sigma 0,01

| No | Iterasi | $x$    | $y$    | $\lambda$ |
|----|---------|--------|--------|-----------|
| 1  | 149     | 1.9262 | 0.5233 | 0.9833    |
| 2  | 150     | 1.9293 | 0.5203 | 0.9923    |
| 3  | 151     | 1.9323 | 0.5172 | 1.0013    |

Tabel 1 hanya memuat tiga iterasi terakhir di sekitar akar. Penentuan akar dapat dipilih dengan melihat nilai  $\lambda$  yang paling dekat dengan 1, atau dengan mengambil rata-rata dari dua akar yang memiliki nilai  $\lambda$  paling dekat dengan 1.

Sebagai pembandingan, digunakan Metode Newton dengan nilai awal untuk  $x$  dan  $y$  yang sama, yaitu (1, 1). Tetapi matriks Jacobi dari sistem (7) ini bernilai singular untuk nilai awal (1, 1), sehingga harus diubah, misalnya dipilih yang tidak terlalu jauh dari nilai awal semula, yaitu (2, 1). Hasilnya diperlihatkan pada tabel berikut.

Tabel 2: Keluaran Metode Newton untuk sistem (7) dengan nilai awal (2, 1).

| No | Iterasi | $x$    | $y$    |
|----|---------|--------|--------|
| 1  | 1       | 2.0000 | 1.0000 |
| 2  | 2       | 2.0000 | 0.5000 |
| 3  | 3       | 1.9333 | 0.5167 |
| 4  | 4       | 1.9319 | 0.5176 |
| 5  | 5       | 1.9319 | 0.5176 |

Dua keluaran di atas hanya memuat satu akar saja, sedangkan seperti telah dijelaskan di awal contoh 2, sistem memiliki empat buah akar. Dari hasil menjalankan algoritma kedua metode, diperoleh kesimpulan bahwa untuk Metode Newton akan menghasilkan tiga akar yang lain hanya jika nilai awalnya diubah dengan syarat tidak terlalu jauh dari perkiraan akar itu sendiri. Misalnya untuk akar yang kedua, dipilih nilai awal (1, 2), akar yang ketiga, dipilih nilai awal (-2, -1), dan akar keempat, dipilih nilai awal (-1, -2).

Hasilnya dapat dilihat pada tiga tabel berikut ini.

Tabel 3: Keluaran Metode Newton untuk sistem (7) dengan nilai awal (1, 2).

| No | Iterasi | $x$    | $y$    |
|----|---------|--------|--------|
| 1  | 1       | 1.0000 | 2.0000 |
| 2  | 2       | 0.5000 | 2.0000 |
| 3  | 3       | 0.5167 | 1.9333 |
| 4  | 4       | 0.5176 | 1.9319 |
| 5  | 5       | 0.5176 | 1.9319 |

Tabel 4: Keluaran Metode Newton untuk sistem (7) dengan nilai awal (-1, -2).

| No | Iterasi | $x$     | $y$     |
|----|---------|---------|---------|
| 1  | 1       | -1.0000 | -2.0000 |
| 2  | 2       | -0.5000 | -2.0000 |
| 3  | 3       | -0.5167 | -1.9333 |
| 4  | 4       | -0.5176 | -1.9319 |
| 5  | 5       | -0.5176 | -1.9319 |

Tabel 5: Keluaran Metode Newton untuk sistem (7) dengan nilai awal (-2, -1).

| No | Iterasi | $x$     | $y$     |
|----|---------|---------|---------|
| 1  | 1       | -2.0000 | -1.0000 |
| 2  | 2       | -2.0000 | -0.5000 |
| 3  | 3       | -1.9333 | -0.5167 |
| 4  | 4       | -1.9319 | -0.5176 |
| 5  | 5       | -1.9319 | -0.5176 |

Jumlah iterasi tiap keluaran Metode Newton sangat sedikit, karena pemilihan nilai awal yang cukup dekat dengan perkiraan akar.

Pada Metode Homotopi dengan nilai atau titik awal yang sama, cukup sulit untuk mendapatkan tiga akar yang lain, sekalipun jumlah iterasinya diperbesar. Oleh karena itu titik awalnya pun diubah menjadi (0, -2, 0). Dengan jumlah iterasi kurang dari 1000, hanya diperoleh dua akar, yaitu akar pertama dan kedua. Setelah jumlah iterasi diperbesar hingga 2600 iterasi, diperoleh dua akar yang terakhir. Pada tabel di bawah

ini terlihat akar yang keempat diperoleh pada iterasi yang ke 2560.

Tabel 6: Keluaran Metode Homotopi untuk sistem (7) dengan nilai awal (0, -2, 0).

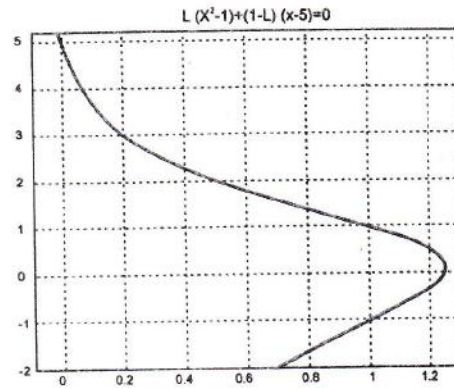
| No | iterasi | $x$     | $y$     | $\lambda$ |
|----|---------|---------|---------|-----------|
| 1  | 344     | 1.9309  | 0.5101  | 0.9940    |
| 2  | 345     | 1.9319  | 0.5179  | 1.0002    |
| 3  | 346     | 1.9329  | 0.5257  | 1.0064    |
| 4  | 600     | 0.5322  | 1.9288  | 1.0068    |
| 5  | 601     | 0.5233  | 1.9307  | 1.0026    |
| 6  | 602     | -0.5144 | 1.9325  | 0.9985    |
| 7  | 1111    | -1.9366 | -0.5115 | 0.9938    |
| 8  | 1112    | -1.9318 | -0.5176 | 1.0000    |
| 9  | 1113    | -1.9271 | -0.5237 | 1.0063    |
| 10 | 2560    | -0.5177 | -1.9318 | 1.0006    |
| 11 | 2561    | -0.5169 | -1.9333 | 0.9908    |

Jadi, pada Metode Homotopi, dengan pemilihan titik awal yang tepat dan jumlah iterasi yang memadai, semua akar akan diperoleh sekaligus dengan satu kali perintah. Tetapi jika pemilihan titik awal kurang atau tidak tepat, sekalipun jumlah iterasinya cukup besar, mungkin saja akar tidak akan diperoleh.

Pada fungsi satu peubah, hal ini dapat diilustrasikan dengan kurva lintasan penyelesaian yang tidak kontinu (tidak terbatas nilai  $x$ -nya) sepanjang interval  $\lambda$  di  $[0, 1]$ . Lihat contoh 3 berikut ini.

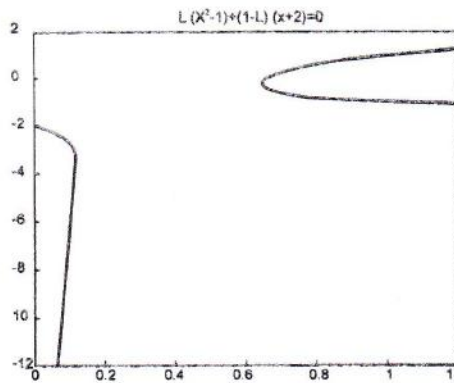
Contoh 3:

Diberikan fungsi  $f(x) = x^2 - 1$ . Fungsi ini memiliki dua akar real, yaitu 1 dan -1. Dengan memilih titik awal (5, 0), dua akar ini diperoleh sekaligus, seperti terlihat pada Gambar 4 berikut ini



Gambar 4: Lintasan penyelesaian fungsi  $f(x) = x^2 - 1$  dengan titik awal (5, 0)

Tetapi jika diambil titik awal (-2, 0), maka lintasan penyelesaiannya tidak terbatas, seperti terlihat pada gambar berikut ini.



Gambar 5: Lintasan penyelesaian fungsi  $f(x) = x^2 - 1$  dengan titik awal (-2, 0)

Berikut ini diberikan sistem persamaan taklinear dengan 3 peubah bebas  $x$ ,  $y$ , dan  $z$ .

Contoh 4: Dengan menggunakan algoritma homotopi, seperti pada contoh 3, akan diselesaikan sistem

$$\begin{aligned} 9x^2 + 36y^2 + 4z^2 - 36 &= 0 \\ x^2 - 2y^2 - 20z &= 0 \\ 16x - x^3 + 2y^2 - 16z^2 &= 0 \end{aligned} \quad (8)$$

Jika dipilih titik awal (0, 1, -1, 0) dengan sigma 0,01, maka diperoleh satu akar yaitu (-0.1147; -0.9978; -0.0988) dengan iterasi kurang dari 450.

Tabel 7: Keluaran Metode Homotopi untuk sistem (8) dengan titik awal (0, 1, -1, 0).

| No | Iterasi | x      | y      | z      | $\lambda$ |
|----|---------|--------|--------|--------|-----------|
| 1  | 432     | -      | -      | -      | 0.9978    |
|    |         | 0.1155 | 0.9978 | 0.0982 |           |
| 2  | 433     | -      | -      | -      | 0.9909    |
|    |         | 0.1151 | 0.9978 | 0.0998 |           |
| 3  | 434     | -      | -      | -      | 1.0008    |
|    |         | 0.1147 | 0.9978 | 0.0988 |           |

Jika dipilih titik awal (0, 0, 0, 0) dengan sigma yang sama, diperoleh akar yang sama tetapi jumlah iterasinya berkurang.

Tabel 8: Keluaran Metode Homotopi untuk sistem (8) dengan titik awal (0, 0, 0, 0).

| No | Iterasi | x      | y      | z      | $\lambda$ |
|----|---------|--------|--------|--------|-----------|
| 1  | 330     | -      | -      | -      | 0.9821    |
|    |         | 0.1145 | 0.9978 | 0.0998 |           |
| 2  | 331     | -      | -      | -      | 0.9921    |
|    |         | 0.1146 | 0.9978 | 0.0993 |           |
| 3  | 332     | -      | -      | -      | 1.0021    |
|    |         | 0.1148 | 0.9978 | 0.0988 |           |

Pemilihan titik awal nol, tidak dapat dilakukan untuk metode Newton, karena mengakibatkan matriks Jacobi dari sistem singular sehingga penyelesaian gagal diperoleh.

Adapun program dari algoritma metode Homotopi dengan pendekatan prediktor-korektor untuk sistem persamaan n peubah dapat dilihat pada lampiran.

## KESIMPULAN DAN SARAN

Metode Homotopi memiliki kelebihan, dibanding dengan metode Newton pertama, tidak gagal dalam memberikan penyelesaian sebuah sistem persamaan taklinear, walaupun matriks Jacobi dari sistem bersifat singular. Kedua, metode ini juga dapat memberikan semua penyelesaian secara serentak dengan satu kali perintah dengan

pemilihan titik awal yang tepat dan jumlah iterasi yang memadai. Ketiga, pemilihan titik awal yang tidak terlalu dekat atau cukup jauh dari akar, akan tetap memberikan penyelesaian, jika lintasan penyelesaiannya kontinu pada interval  $\lambda$  di  $[0, 1]$ .

Secara garis besar, Metode Homotopi ini masih sensitif terhadap pemilihan titik awal seperti halnya Metode Newton, walaupun dalam hal yang berbeda. Oleh karena itu, pada penelitian selanjutnya diharapkan dapat menemukan batasan yang tepat dalam pemilihan titik awal ini, sehingga selalu diperoleh penyelesaian dari setiap sistem yang diberikan.

## DAFTAR PUSTAKA

1. Hager, W.W., (1998), *Applied Numerical Linear Algebra*, Prentice Hall, USA
2. Mathews, J.H. and Kurtis D.Fink, (1999), *Numerical Method Using MATLAB*, Prentice Hall, USA
3. Nocedal, J and Stephen J.W., (1999), *Numerical Optimization*, Springer, New York
4. Schwartz, A., (1964), *Analytic Geometry and Calculus*, Holt, Rinehart and Winston, USA
5. <http://www.maths.lth.se/na/courses/FMN081/FMN081-06/lecture8.pdf>. 30 Agustus 2007  
<http://www.emis.de/journals/RCM/Articulos/808.pdf>. 30 Agustus 2007

**Lampiran**

**Program Utama Fungsi Homotopi untuk  
Sistem Persamaan n Peubah**

```

n0=input('Masukan:');%masukan
banyaknya peubah
for i=1:n0
    xv(i,1)=sym(input('nama
variable:', 's'));%masukan    nama-nama
    peubah
end
av=a0;%masukan titik tetap a0
f=f(x);%masukan    sistem    persamaan
taklinear n peubah
N=N0;%banyaknya iterasi
Sigma=sigma0;%faktor pengali vektor singgung
untuk mendapatkan titik prediktor
Eps=eps;%panjang range yang diambil di
sekitar akar
S=fsHomotopynvar(av,f,sigma,N,eps,n0,xv);
%fungsi Homotopi n peubah
Fungsi Homotopi untuk Sistem
Persamaan n Peubah
function
S=fsHomotopynvar(av,f,sigma,N,eps,n0,xv)
syms L;
p=length(xv);
H=L*f+(1-L)*(xv-av);
for i=1:p,
    for j=i:p+1,
        If j==p+1,
            J(i,j)=diff(H(i),L);
        else
            J(i,j)=diff(H(i),xv(j));
        end
    end
end
end
A=J;
[m,n]=size(A);
CP{1}=[av;0];
XV=[conj(xv'),L];
for i=1:N
    As=subs(A,XV,CP,{i});
    [Q,R]=faktorisasiQR(As);
    R1=R(:,1:n-1);
    inv(R1);
    w=R(:,n);

```

```

v12=(inv(R1)*w);
v=[v12;-1];
c=-1/norm(v);
tv{i}=c*v;
If i>=2
    Alpha=acos(tv{i}'*tv{i-1});
    If alpha<pi/2
        tv{i}=c*v;
    else
        tv{i}=-c*v;
    end
end
PP=CP{i}+sigma*tv{i};
k=sigma*tv{i};
maks=max(abs(k));
B=zeros(m+1,n);
for j=1:n
    If maks==abs(k(j))
        B(m+1,j)=1;
    end
end
Ap=subs(A,XV,PP);
B(1:m,:)=Ap;
h1=subs(H,XV,PP);
h=[-h1;0];
delta=B\h;
CP{i+1}=PP+delta;
CP{i+1};
i=i+1;end

```