IMPLEMENTASI ALGORITMA BROYDEN UNTUK MENYELESAIKAN SISTEM PERSAMAAN FUZZY PENUH TAK LINEAR

TESIS

Oleh

WAHYU MEGARANI

PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022

ABSTRACT

IMPLEMENTATION OF BROYDEN'S ALGORITHM TO SOLVE FULLY FUZZY NONLINEAR SYSTEM

By

Wahyu Megarani

In this thesis, an algorithm of numerical method to solve fully fuzzy nonlinear system using Broyden's Method was constructed in a pseudocode and be implemented by Matlab computer programming. To show that the proposed algorithm work, two numerical examples were provided. Based on experimental result, the algorithm give the solution with a small error in a relatively short time.

Keywords: Algorithm, Fully Fuzzy Nonlinear System, Broyden's Method.

ABSTRAK

IMPLEMENTASI ALGORITMA BROYDEN UNTUK MENYELESAIKAN SISTEM PERSAMAAN FUZZY PENUH TAK LINEAR

Oleh

Wahyu Megarani

Pada tesis ini, sebuah algoritma metode numerik untuk menyelesaikan sistem persamaan fuzzy penuh tak linear menggunakan Metode Broyden telah dikonstruksi dalam *pseudocode* dan diimplementasikan dengan pemrograman Matlab. Untuk menunjukkan bahwa algoritma yang diusulkan bekerja, dua contoh numerik telah disajikan. Berdasarkan hasil percobaan, algoritma tersebut memberikan solusi dengan kesalahan yang kecil dalam waktu yang relatif singkat.

Kata kunci: Algoritma, Sistem Persamaan Fuzzy Penuh Tak Linear, Metode Broyden.

IMPLEMENTASI ALGORITMA BROYDEN UNTUK MENYELESAIKAN SISTEM PERSAMAAN FUZZY PENUH TAK LINEAR

Oleh

Wahyu Megarani

TESIS

Sebagai salah satu syarat untuk mencapai gelar
MAGISTER MATEMATIKA

Pada
Program Studi Magister Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam

PROGRAM STUDI MAGISTER MATEMATIKA
FAKULTAS MATEMATIKADAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022

Judul Tesis	: **IMPLEMENTASI ALGORITMA BROYDEN UNTUK MENYELESAIKAN SISTEM PERSAMAAN FUZZY PENUH TAK LINEAR**
Nama Mahasiswa	: **Wahyu Megarani**
Nomor Pokok Mahasiswa	: **2027031010**
Program Studi	: **Magister Matematika**
Jurusan	: **Matematika**
Fakultas	: **Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Prof. Dr. La Zakaria, S.Si., M.Sc.
NIP 19690213 199402 1 001

Dr. Ahmad Faisol, S.Si., M.Sc.
NIP 19800206 200312 1 003

2. Ketua Program Studi Magister Matematika

Dr. Asmiati, S.Si., M.Si.
NIP 19760411 200012 2 001

MENGESAHKAN

1. Tim Penguji

Ketua	: **Prof. Dr. La Zakaria, S.Si., M.Sc.**	....
Sekretaris	: **Dr. Ahmad Faisol, S.Si., M.Sc.**	....
Penguji Bukan Pembimbing	: a. **Dr. Aang Nuryaman, S.Si., M.Si.**	....
	b. **Prof. Dra. Wamiliana, M.A., Ph.D.**	....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP. 19740705 200003 1 001

3. Direktur Program Pasca Sarjana

Prof. Dr. Ahmad Saudi Samosir, S.T., M.T.
NIP. 19710415 199803 1 005

Tanggal Lulus Ujian : **18 Agustus 2022**

PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama	: **Wahyu Megarani**
Nomor Pokok Mahasiswa	: **2027031010**
Program Studi	: **Magister Matematika**
Jurusan	: **Matematika**

Dengan ini menyatakan bahwa tesis saya yang berjudul **"IMPLEMENTASI ALGORITMA BROYDEN UNTUK MENYELESAIKAN SISTEM PERSAMAAN FUZZY PENUH TAK LINEAR"** adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa tesis ini merupakan hasil salinan atau telah dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai ketentuan akademik yang berlaku.

Bandar Lampung, 18 Agustus 2022
Penulis

METERAI TEMPEL
10000
7A0DAJX990105843

Wahyu Megarani
NPM. 2027031010

RIWAYAT HIDUP

Penulis bernama lengkap Wahyu Megarani, anak ketiga dari tiga bersaudara dengan ayah bernama Wakidi dan ibu bernama Yeyet Supriyati. Penulis dilahirkan di Bandar Lampung pada tanggal 2 Mei 1998. Penulis menyelesaikan pendidikan di TK Tri Dharma pada tahun 2004, SDN 1 Bumi Waras pada tahun 2010, SMPN 6 Bandar Lampung pada tahun 2013, dan SMAN 8 Bandar Lampung pada tahun 2016.

Pada tahun 2016 penulis diterima sebagai mahasiswa Universitas Lampung melalui jalur SBMPTN pada Program Studi S1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam. Selama menjadi mahasiswa, pada tahun 2017-2018 penulis mengikuti organisasi tingkat fakultas yaitu ROIS FMIPA sebagai anggota Biro Kesekretariatan dan Rumah Tangga. Pada bulan Januari 2019 penulis melaksanakan Kuliah Kerja Nyata (KKN) selama 40 hari di Desa Sinar Harapan, Kecamatan Sungkai Barat Kabupaten Lampung Utara. Pada bulan Juli 2019 penulis melaksanakan Kerja Praktik (KP) selama 40 hari di Balai Riset dan Standardisasi (Baristand) Industri Bandar Lampung. Penulis menyelesaikan pendidikan S1 Matematika pada bulan September tahun 2020 dan pada tahun yang sama penulis melanjutkan pendidikan S2 di Universitas Lampung pada Program Studi Magister Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam.

MOTTO

Rasulullah Shallallahu Alayhi Wasallam bersabda:

"Ya Allah, barangsiapa yang diberi tanggung jawab untuk menangani urusan umatku, lalu ia mempersulit mereka, maka persulitlah hidupnya. Dan barangsiapa yang diberi tanggung jawab untuk mengurusi umatku, lalu ia memudahkan urusan mereka, maka mudahkanlah hidupnya."

(HR. Muslim)

PERSEMBAHAN

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya dalam menjalani kehidupan, dan shalawat serta salam selalu tercurah kepada Nabi Muhammad Shallallahu Alayhi Wasallam.

Ku persembahkan tesis ini untuk:

Keluarga

Ibu, Bapak, kedua kakak beserta keluarganya yang senantiasa mendoakan dengan ikhlas di setiap waktunya.

Dosen Pembimbing dan Pembahas

Dosen pembimbing dan pembahas yang telah memberikan pelajaran, arahan, dan masukan dalam penulisan tesis.

Sahabat dan Teman

Sahabat serta teman-teman seangkatan dan seperjuangan yang selalu memberikan motivasi.

Universitas Lampung

SANWACANA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan kasih sayang serta karunia-Nya kepada penulis sehingga tesis yang berjudul "Implementasi Algoritma Broyden untuk Menyelesaikan Sistem Persamaan Fuzzy Penuh Tak Linear" ini dapat diselesaikan. Shalawat serta salam semoga selalu tercurahkan kepada Nabi Muhammad SAW dan kita nantikan syafaatnya di yaumil akhir nanti.

Terselesaikannya tesis ini tidak lepas dari bantuan dan kerja sama dari berbagai pihak. Oleh karena itu, dalam kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Ibu dan Ayah yang senantiasa memberikan motivasi dan doa kepada penulis.
2. Kedua kakak yaitu Tuti Wahyuni dan Wahyudi beserta keluarga yang turut serta mendoakan dan memberikan motivasi.
3. Bapak Prof. Dr. La Zakaria, S.Si., M.Sc., selaku dosen pembimbing pertama yang selalu membimbing dan memberi arahan selama penulisan tesis.
4. Bapak Dr. Ahmad Faisol, S.Si., M.Sc., selaku dosen pembimbing kedua yang telah mengoreksi format penulisan, serta memberikan kritik dan saran selama penulisan tesis.

5. Bapak Dr. Aang Nuryaman, S.Si., M.Si., selaku dosen penguji utama yang telah mengoreksi kekurangan, serta memberi kritik dan saran selama penulisan tesis.
6. Ibu Prof. Dra. Wamiliana, M.A., Ph.D., selaku dosen penguji kedua yang telah memberikan arahan, masukan, serta kritik dan saran selama penulisan tesis.
7. Ibu Dr. Asmiati, S.Si., M.Si., selaku Ketua Program Studi Magister Matematika FMIPA Universitas Lampung.
8. Bapak Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T., selaku Dekan FMIPA Universitas Lampung.
9. Bapak Prof. Dr. Ahmad Saudi Samosir, S.T., M.T., selaku Direktur Pasca Sarjana.
10. Seluruh dosen serta karyawan di Jurusan Matematika FMIPA Universitas Lampung.
11. Muhammad Alif Abimanyu yang senantiasa memberi motivasi, dukungan, dan menemani sejak awal perkuliahan sampai terselesaikannya tesis ini.
12. Para sahabat, Isma, Rara, Evril, Fiqoh, dan Erisa yang selalu memberikan motivasi kepada penulis.
13. Teman-teman Magister Matematika angkatan 2020.

Bandar Lampung, 18 Agustus 2022
Penulis

Wahyu Megarani

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Sistem persamaan tak linear merupakan kumpulan dari beberapa persamaan tak linear yang saling berhubungan untuk mencari solusi yaitu berupa nilai variabel yang memenuhi semua persamaan tak linear tersebut. Secara umum, sistem persamaan tak linear terdiri atas m persamaan dengan n variabel dan dapat dibentuk secara langsung dari masalah-masalah nyata sehingga dibutuhkan proses penyelesaian. Penyelesaian sistem persamaan tak linear dapat dilakukan dengan dua metode, yaitu metode langsung yang biasa disebut metode analitik dan metode tidak langsung yang biasa disebut metode numerik.

Seiring perkembangan ilmu matematika, konstanta dalam sistem persamaan tak linear yang biasanya berupa bilangan riil kini dapat berupa bilangan fuzzy, dan dapat pula diselesaikan menggunakan metode yang sama. Fuzzy dapat diartikan sebagai kabur atau samar-samar, dan digunakan dalam masalah yang mengandung unsur ketidakpastian. Dari sistem persamaan fuzzy tak linear pun dapat dikembangkan kembali menjadi sistem persamaan fuzzy penuh tak linear yang menerapkan operasi perhitungan bilangan fuzzy yang berbeda dari operasi bilangan riil.

Beberapa peneliti telah melakukan penelitian mengenai teori himpunan fuzzy, di antaranya Ziqan et al., (2021) yang melakukan sebuah studi tentang sistem fuzzy penuh linear dengan fungsi keanggotaan bilangan fuzzy trapesium dan heksagonal. Pada tahun 2019 Deswita & Mashadi memperkenalkan konsep baru aritmatika pada bilangan fuzzy dengan fungsi keanggotaan segitiga sehingga didapat bentuk perkalian bilangan fuzzy dalam beberapa kasus. Selanjutnya dilakukan beberapa penelitian guna mendapatkan solusi dari persamaan maupun sistem persamaan fuzzy baik linear maupun tak linear, seperti Ramli et al., (2010) memperkenalkan Algoritma Broyden untuk menyelesaikan persamaan fuzzy tak linear. Senthilkumar & Rajendran (2011) memperkenalkan sebuah metode untuk menyelesaikan sistem fuzzy penuh linear dengan koefisien matriks yang simetris. Selanjutnya Malkawi et al., (2015) mengusulkan sebuah metode komputasi untuk menghasilkan solusi positif dari sembarang sistem fuzzy penuh linear. Pada tahun 2018, Inearat & Qatanani melakukan penelitian tentang tiga skema iterasi numerik yaitu Jacobi, Gauss Seidel, dan SOR (*Successive Over-Relaxation*) untuk menyelesaikan sistem fuzzy linear. Kemudian Marzuki et al., (2018) melakukan penelitian tentang metode Dekomposisi Nilai Singular (SDV) untuk menyelesaikan sistem fuzzy penuh linear. Safitri & Mashadi (2019) mendiskusikan aljabar fuzzy alternatif untuk menyelesaikan sistem fuzzy penuh ganda linear dengan metode Dekomposisi ST. Kemudian Jafari et al., (2020) mengusulkan sebuah metode pendekatan untuk menentukan perkiraan solusi dari sistem fuzzy penuh tak linear dan Megarani et al., (2022) mengusulkan Algoritma Jacobi untuk menyelesaikan sistem persamaan fuzzy penuh linear ganda.

Jafarian & Jafari melakukan penelitian pada tahun 2019 tentang sebuah metode untuk menyelesaikan sistem persamaan matriks fuzzy penuh tak linear $\tilde{A}\tilde{X} + \tilde{C}\tilde{X}^2 + \cdots + \tilde{E}\tilde{X}^n = \tilde{B}$ dengan cara menambah variabel baru dan membuat kendala kesamaan, lalu mengeliminasi variabel baru tersebut guna mendapatkan solusi berupa bilangan fuzzy tak negatif. Berdasarkan hasil pada jurnal tersebut dan penelitian yang dilakukan oleh Ramli mengenai Metode Broyden yang digunakan untuk menyelesaikan sebuah persamaan fuzzy tak linear, penulis tertarik untuk melakukan penelitian guna menyelesaikan sistem persamaan fuzzy penuh tak linear yang dibahas dengan menggunakan metode numerik, yaitu Metode Broyden agar solusinya dapat diperoleh dalam waktu yang relatif lebih singkat dan diimplementasikan dengan pemrograman Matlab.

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengkonstruksi algoritma Metode Broyden untuk menyelesaikan sistem persamaan fuzzy penuh tak linear dan dapat diimplementasikan menggunakan Matlab.

1.3 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut :

1. Memberikan informasi tentang penggunaan Metode Broyden dalam menentukan solusi sistem persamaan fuzzy penuh tak linear.
2. Memberikan wawasan dalam membuat algoritma Metode Broyden agar dapat diimplementasikan menggunakan Matlab.

3. Dapat dijadikan referensi pada penelitian berikutnya mengenai penyelesaian sistem persamaan fuzzy penuh tak linear.

II. TINJAUAN PUSTAKA

Pada bab ini diberikan beberapa definisi dan penjelasan mengenai sistem persamaan tak linear, teori himpunan fuzzy, dan metode numerik untuk mendukung penulis dalam menyelesaikan permasalahan yang akan dikaji.

2.1 Persamaan Tak linear

Menurut Dugopolski (2006), persamaan tak linear adalah suatu persamaan yang grafiknya bukan berupa garis lurus. Salah satu contoh bentuk persamaan tak linear adalah sebagai berikut:

$$x^2 + 3x + 2 = 0$$

di mana persamaan tak linear tersebut dapat diselesaikan dengan menentukan akar-akar persamaannya, yaitu nilai x yang menyebabkan fungsi $f(x)$ sama dengan nol.

Sementara yang dimaksud dengan sistem persamaan tak linear adalah kumpulan dari beberapa persamaan tak linear yang saling berkaitan satu sama lain, dengan bentuk umum sebagai berikut:

$$\begin{gathered} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_i(x_1, x_2, \dots, x_n) = 0 \end{gathered} \tag{1}$$

untuk setiap fungsi f_i dengan $i = 1, 2, \ldots, n$ dapat digambarkan dengan sebuah vektor $x = (x_1, x_2, \ldots, x_n)^t$ untuk dimensi ke-n pada ruang $\mathbb{R}^n$ pada garis real $\mathbb{R}$ (Burden & Faires, 2005).

2.2 Teori Himpunan Fuzzy

Menurut Deswita & Mashadi (2019), bilangan fuzzy adalah sebuah atribut dalam himpunan fuzzy di mana $\tilde{a}: \mathbb{R} \to [0,1]$ yang memenuhi:

1. $\tilde{a}$ semikontinu atas
2. $\tilde{a}(x) = 0$ di luar interval $[0,1]$
3. Ada bilangan real a, b pada $[c, d]$ di mana
   (i) $\tilde{a}(x)$ monoton naik pada $[c, a]$,
   (ii) $\tilde{a}(x)$ monoton turun pada $[b, d]$,
   (iii) $\tilde{a}(x) = 1$, untuk $a \leq x \leq b$.

Untuk menjelaskan poin 1 di atas maka diberikan satu definisi mengenai fungsi semikontinu sebagai berikut.

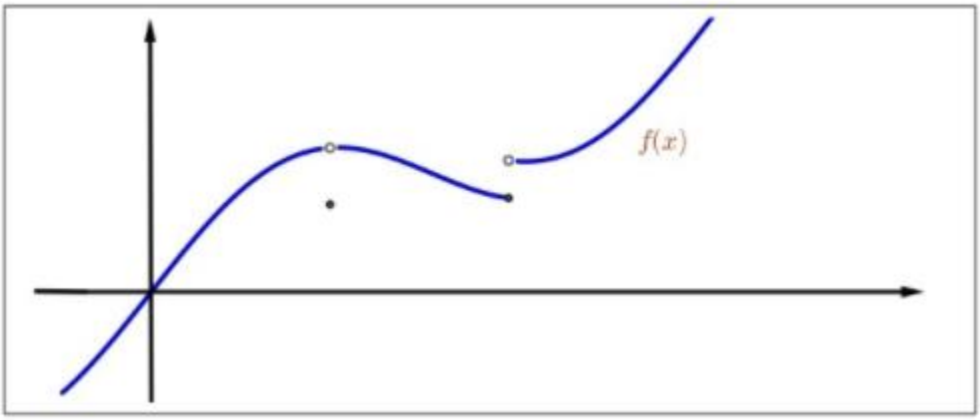
Definisi 2.2.1 Misalkan $f: D \to \mathbb{R}$ dan $\bar{x} \in D$. Dikatakan bahwa f semikontinu bawah pada $\bar{x}$ jika untuk setiap $\varepsilon > 0$, ada $\delta > 0$ sedemikian sehingga

$$f(\bar{x}) - \varepsilon < f(x) \text{ untuk semua } x \in B(\bar{x}; \delta) \cap D. \tag{2}$$

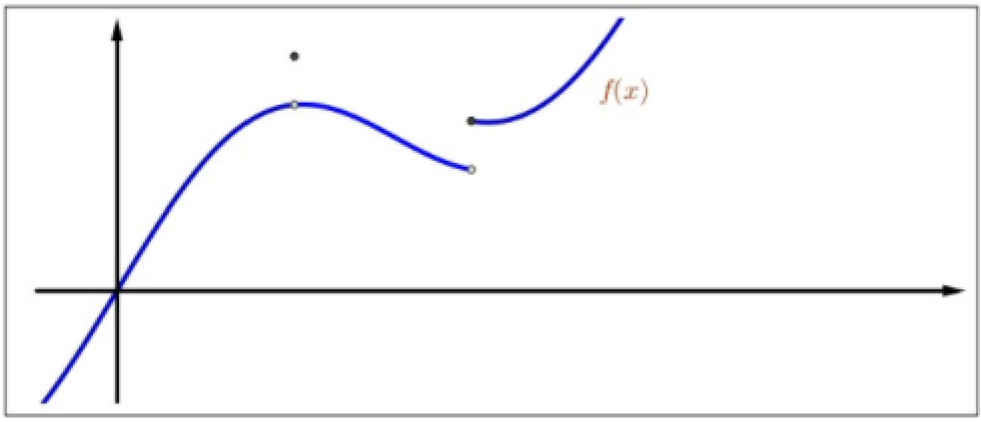
Dengan hal serupa, dikatakan bahwa f semikontinu atas pada $\bar{x}$ jika untuk setiap $\varepsilon > 0$, ada $\delta > 0$ sedemikian sehingga

$$f(x) < f(\bar{x}) + \varepsilon \text{ untuk semua } x \in B(\bar{x}; \delta) \cap D. \tag{3}$$

Jelas bahwa f kontinu pada $\bar{x}$ jika dan hanya jika f semikontinu bawah dan semikontinu atas pada titik tersebut (Liliana et al., 1973).



Gambar 1. Grafik Fungsi Semikontinu Bawah



Gambar 2. Grafik Fungsi Semikontinu Atas

Untuk mengetahui beberapa teori tentang himpunan fuzzy, diberikan beberapa definisi dan penjelasan sebagai berikut.

Definisi 2.2.2 Himpunan bagian fuzzy $\tilde{A}$ pada $\mathbb{R}$ ditentukan oleh fungsi keanggotaan $\mu_{\tilde{A}}: \mathbb{R} \to [0,1]$, yang menempatkan suatu bilangan real $\mu_{\tilde{A}}$ dalam interval $[0,1]$, untuk setiap $x \in \mathbb{R}$, di mana nilai $\mu_{\tilde{A}}$ pada x menunjukkan derajat keanggotaan dari x pada $\tilde{A}$ (Senthilkumar & Rajendran, 2011).

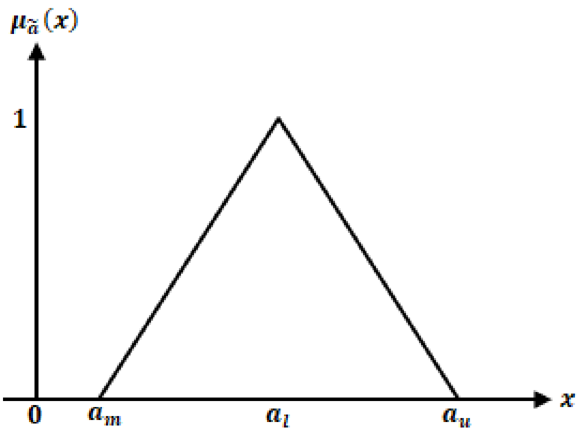
Contoh 2.2.2.1

Misal terdapat himpunan nilai ujian dari 7 orang murid, yaitu 1, 5, 7, 8, 7, 8, dan 9. Maka derajat keanggotaan nilai dari ketujuh murid tersebut dengan kedekatan pada nilai sempurna, yaitu 10 adalah $\mu_{\tilde{A}}(1) = 0{,}1$; $\mu_{\tilde{A}}(5) = 0{,}5$; $\mu_{\tilde{A}}(7) = 0{,}7$; $\mu_{\tilde{A}}(8) = 0{,}8$; $\mu_{\tilde{A}}(7) = 0{,}7$; $\mu_{\tilde{A}}(8) = 0{,}8$; $\mu_{\tilde{A}}(9) = 0{,}9$.

Definisi 2.2.3 Sembarang bilangan fuzzy dalam bentuk $\tilde{a} = (m - \alpha, m, m + \beta) = (a_m, a_l, a_u)$ dengan fungsi keanggotaan sebagai berikut

$$\mu_{\tilde{a}}(x) = \begin{cases} 1 - \dfrac{m - x}{\alpha}, & m - \alpha \leq x \leq m, \ \ \alpha > 0 \\ 1 - \dfrac{x - m}{\beta}, & m \leq x \leq m + \beta, \ \ \beta > 0 \\ 0, & selainnya \end{cases} \tag{4}$$

disebut bilangan fuzzy segitiga dan grafiknya dapat ditampilkan sebagai berikut



Gambar 3. Bilangan Fuzzy Segitiga $\tilde{a} = (a_m, a_l, a_u)$.

Dua bilangan fuzzy $\tilde{a} = (a_m, a_l, a_u)$ dan $\tilde{b} = (b_m, b_l, b_u)$ dikatakan sama, jika dan hanya jika $a_m = b_m$, $a_l = b_l$, dan $a_u = b_u$ (Jafarian & Jafari, 2019).

Matriks $\widetilde{\boldsymbol{A}} = (\tilde{a}_{ij})$ disebut matriks fuzzy jika setiap anggota $\widetilde{\boldsymbol{A}}$ adalah bilangan fuzzy. Matriks fuzzy $\widetilde{\boldsymbol{A}}$ akan bernilai positif (negatif) dan dilambangkan dengan $\widetilde{\boldsymbol{A}} > 0\ (\widetilde{\boldsymbol{A}} < 0)$, jika setiap anggota $\widetilde{\boldsymbol{A}}$ positif (negatif). $\widetilde{\boldsymbol{A}}$ akan menjadi non-positif (non-negatif) dan dilambangkan dengan $\widetilde{\boldsymbol{A}} \leq 0\ (\widetilde{\boldsymbol{A}} \geq 0)$ jika setiap anggota $\widetilde{\boldsymbol{A}}$ adalah non-positif (non-negatif). Matriks fuzzy persegi $\widetilde{\boldsymbol{A}} = (\tilde{a}_{ij})$ akan menjadi matriks fuzzy segitiga atas jika $\tilde{a}_{ij} = \tilde{0} = (0, 0, 0), \forall i > j$, dan matriks fuzzy persegi $\widetilde{\boldsymbol{A}} = (\tilde{a}_{ij})$ akan menjadi matriks fuzzy segitiga bawah jika $\tilde{a}_{ij} = \tilde{0} = (0, 0, 0), \forall i < j$ (Senthilkumar & Rajendran, 2011).

Selanjutnya diberikan satu definisi untuk memahami operasi perhitungan pada bilangan fuzzy segitiga, di mana operasi bilangan fuzzy segitiga berbeda dengan operasi bilangan real.

Definisi 2.2.4 Untuk dua bilangan fuzzy segitiga $\tilde{a} = (a_m, a_l, a_u)$ dan $\tilde{b} = (b_m, b_l, b_u)$, maka operasi perhitungannya adalah sebagai berikut:

a. $\tilde{a} \oplus \tilde{b} = (a_m + b_m,\ a_l + b_l,\ a_u + b_u)$

b. $-\tilde{a} = (-a_u, -a_l, -a_m)$

c. $\tilde{a} \ominus \tilde{b} = (a_m - b_u,\ a_l - b_l,\ a_u - b_m)$

d. Perkalian pada bilangan fuzzy ini dilambangkan oleh $\hat{*}$ dan didasarkan pada prinsip tambahan namun sedikit berbeda dari perkalian fuzzy klasik

$$\tilde{a} \,\hat{*}\, \tilde{b} = (c_m, c_l, c_u)$$

dengan $c_l = a_l \cdot b_l$

$$c_m = \min(a_m \cdot b_m,\ a_m \cdot b_u,\ a_u \cdot b_m,\ a_u \cdot b_u)$$

$$c_u = \max(a_m \cdot b_m,\ a_m \cdot b_u,\ a_u \cdot b_m,\ a_u \cdot b_u)$$

Jika $\widetilde{a}$ adalah sembarang bilangan fuzzy dan $\widetilde{b}$ bersifat non-negatif, maka:

$$\widetilde{a} \hat{*} \widetilde{b} = \begin{cases} (a_m \cdot b_m, a_l \cdot b_l, a_u \cdot b_u), & a_m \geq 0, \\ (a_m \cdot b_u, a_l \cdot b_l, a_u \cdot b_u), & a_m < 0,\ \ a_u \geq 0, \\ (a_m \cdot b_m, a_l \cdot b_l, a_u \cdot b_m), & a_m < 0,\ \ a_u < 0 \end{cases}$$

(Jafarian & Jafari, 2019).

2.3 Sistem Persamaan Fuzzy Penuh Tak linear

Menurut Jafarian & Jafari (2019), sistem persamaan fuzzy penuh tak linear memiliki bentuk umum sebagai berikut:

$$\begin{cases} (\tilde{a}_{11} \hat{*} \tilde{x}_1) \oplus (\tilde{a}_{12} \hat{*} \tilde{x}_2) \oplus \ldots \oplus (\tilde{a}_{1n} \hat{*} \tilde{x}_n) \oplus (\tilde{c}_{11} \hat{*} \tilde{x}_1^2) \oplus (\tilde{c}_{12} \hat{*} \tilde{x}_2^2) \oplus \ldots \\ \oplus (\tilde{c}_{1n} \hat{*} \tilde{x}_n^2) \oplus \ldots \oplus (\tilde{e}_{11} \hat{*} \tilde{x}_1^n) \oplus (\tilde{e}_{12} \hat{*} \tilde{x}_2^n) \oplus \ldots \oplus (\tilde{e}_{1n} \hat{*} \tilde{x}_n^n) = \tilde{b}_1 \\ \vdots \\ (\tilde{a}_{n1} \hat{*} \tilde{x}_1) \oplus (\tilde{a}_{n2} \hat{*} \tilde{x}_2) \oplus \ldots \oplus (\tilde{a}_{nn} \hat{*} \tilde{x}_n) \oplus (\tilde{c}_{n1} \hat{*} \tilde{x}_1^2) \oplus (\tilde{c}_{n2} \hat{*} \tilde{x}_2^2) \oplus \ldots \\ \oplus (\tilde{c}_{nn} \hat{*} \tilde{x}_n^2) \oplus \ldots \oplus (\tilde{e}_{n1} \hat{*} \tilde{x}_1^n) \oplus (\tilde{e}_{n2} \hat{*} \tilde{x}_2^n) \oplus \ldots \oplus (\tilde{e}_{nn} \hat{*} \tilde{x}_n^n) = \tilde{b}_n \end{cases} \quad (5)$$

di mana $\tilde{a}_{ij}, \tilde{c}_{ij}$, dan $\tilde{e}_{ij}$ untuk $1 \leq i, j \leq n$ adalah sembarang bilangan fuzzy segitiga, sedangkan $\tilde{b}_i$ pada ruas kanan dan elemen tidak diketahui $\tilde{x}_j$ adalah bilangan fuzzy tak negatif.

2.4 Algoritma

Algoritma merupakan metode umum yang digunakan untuk menyelesaikan kasus-kasus tertentu. Dalam menuliskan algoritma, dapat digunakan bahasa natural atau menggunakan notasi matematika, sehingga masih belum dapat dijalankan pada komputer. Dalam kehidupan sehari-hari, sudah dilakukan penyusunan algoritma untuk menyelesaikan permasalahan atau tantangan yang dihadapi. Sebagai contoh, pada saat diminta untuk menghitung luas segitiga. Sebelum membuat algoritmanya, diperlukan untuk mendefinisikan masukan (*input*) dan luaran (*output*) terlebih dahulu di mana *input* berupa nilai alas dan tinggi segitiga, dan *output* berupa luas segitiga yang sudah dihitung.

Susunan algoritmanya sebagai berikut:

1. Masukkan nilai alas (a) dan nilai tinggi segituga (t)
2. Maka untuk menghitung luas digunakan nilai alas dan tinggi yang sudah ditentukan
3. Rumus untuk menghitung luas segitiga yaitu $L = \frac{1}{2} \times a \times t$
4. Nilai *L* (luas) akan dicetak sebagai *output* (keluaran) ke perangkat

Algoritma akan lebih baik jika ditulis secara sistematis menggunakan beberapa skema, di antaranya kalimat deskriptif, *pseudocode,* dan *flowchart* (Windarto et al., 2016).

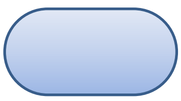
Kalimat deskriptif merupakan salah satu skema penulisan algoritma yang dilakukan dengan cara menuliskan rangkaian instruksi yang harus dikerjakan, dengan menggunakan bahasa yang jelas. Bahasa yang digunakan dalam kalimat deskriptif adalah bahasa yang mudah dimengerti oleh manusia seperti Bahasa Inggris, namun dapat dimodifikasi dengan bahasa sehari-hari termasuk Bahasa Indonesia. Hal ini dikarenakan tidak ada aturan baku dalam penulisan algoritma dengan kalimat deskriptif, maka setiap orang dapat membuat algoritma dengan cara penulisan dan notasinya sendiri sehingga algoritma tersebut dapat dengan mudah dimengerti.

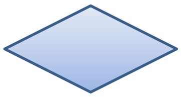
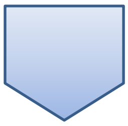
Pseudocode merupakan skema penulisan algoritma yang bahasanya menyerupai bahasa pemrograman tingkat tinggi, namun secara universal masih mudah dipahami karena lebih ringkas dibandingkan dengan algoritma. Dalam *pseudocode* terdapat deskripsi dari algoritma pemrograman komputer dengan struktur yang lebih sederhana dan hanya ditujukan agar dapat dibaca manusia, sehingga tidak dapat dipahami oleh komputer. *Pseudocode* perlu diterjemahkan ke dalam sintaks bahasa pemrograman terlebih dahulu agar bisa dipahami oleh komputer.

Skema penulisan algoritma yang terakhir yaitu *flowchart*. *Flowchart* merupakan cara penulisan algoritma menggunakan notasi grafis, berupa bagan yang memperlihatkan urutan pengerjaan dari suatu program dan hubungan antarproses serta pernyataannya (Retta et al., 2020).

Tabel berikut menampilkan simbol-simbol yang digunakan dalam menyusun *flowchart* (Windarto et al., 2016).

Tabel 1. Simbol-Simbol Dalam *Flowchart*

	***Terminator*** Sebagai simbol *start* atau *end* untuk memulai atau mengakhiri flowchart.
	***Input/Output*** Digunakan untuk menuliskan proses menerima data atau mengeluarkan data.
	**Proses** Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi matematika.

	***Conditional/Decision*** Digunakan untuk menyatakan proses yang membutuhkan keputusan.
	***Preparation*** Digunakan untuk memberikan nilai awal.
	***Arrow*** Sebagai penunjuk arah dan alur proses.
	***Connector (On-page)*** Digunakan untuk menyatikan beberapa *arrow*.
	***Connector (Off-page)*** Digunakan untuk menggabungkan *flowchart* yang harus digambarkan pada halaman yang berbeda. Biasanya pada simbol ini diberi nommor sebagai penanda, misalnya angka 1.
	***Display*** Digunakan untuk menampilkan data ke monitor.

2.5 Metode Numerik

Metode numerik adalah teknik untuk menyelesaikan permasalahan-permasalahan yang diformulasikan secara matematis dengan cara operasi hitungan (*arithmetic*). Berbagai permasalahan dalam bidang ilmu pengetahuan dan teknologi dapat digambarkan dalam bentuk persamaan matematika. Apabila persamaan tersebut mempunyai bentuk sederhana, penyelesaiannya dapat dilakukan secara analitik. Tetapi pada umumnya bentuk persamaan sulit diselesaikan secara analitik, sehingga penyelesaiannya dilakukan secara numerik. Hasil dari penyelesaian numerik merupakan nilai perkiraan atau pendekatan dari penyelesaian analitik atau eksak. Karena merupakan nilai pendekatan, maka terdapat kesalahan terhadap nilai eksak. Nilai kesalahan tersebut harus cukup kecil terhadap tingkat kesalahan yang ditetapkan. Dalam metode numerik terdapat beberapa bentuk proses hitungan atau algoritma untuk menyelesaikan suatu tipe persamaan matematis. Hitungan numerik dapat dilakukan dengan menggunakan salah satu dari bentuk proses hitungan yang paling efisien yang memerlukan waktu hitungan paling cepat (Triatmodjo, 2002).

2.5.1 Galat

Menurut Triatmodjo (2002), penyelesaian secara numerik dari sebuah persamaan atau sistem persamaan matematika dapat menghasilkan nilai hampiran terhadap nilai eksak (nilai sesungguhnya) dari penyelesaian analitik. Penyelesaian numerik senantiasa menghasilkan galat (nilai kesalahan) yang dapat dikalkulasi berdasarkan nilai eksak. Di dalam metode numerik, sering dilakukan proses hampiran secara iteratif. Dalam proses hampiran tersebut, dugaan saat ini dibuat berdasarkan dugaan sebelumnya. Dalam hal ini, galat adalah selisih antara nilai dugaan sebelumnya dengan nilai dugaan sekarang. Ada beberapa jenis galat dalam proses numerik. Salah satunya adalah galat relatif yang diperoleh dari formula berikut:

$$\varepsilon_a = \frac{p_*^{n+1} - p_*^n}{p_*^{n+1}} \times 100\% \tag{6}$$

dengan p_*^n : nilai perkiraan pada iterasi ke n

p_*^{n+1} : nilai perkiraan pada iterasi ke $n+1$

2.5.2 Metode Newton

Menurut Yang et al., (2005), Metode Newton dapat digunakan dengan baik untuk menyelesaikan persamaan tak linear, jika dan hanya jika turunan pertama dari

$f(x_i)$ ada dan kontinu di sepanjang solusinya. Ide utama dari Metode Newton adalah untuk menghampiri kurva $f'(x_i)$ dengan garis singgungnya pada sumbu x sehingga gradien kurva $f'(x_i)$ yaitu

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}} \tag{7}$$

atau

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad f'(x_i) \neq 0 \tag{8}$$

Karena Metode Newton hanya dapat digunakan untuk menyelesaikan persamaan tak linear, maka untuk menyelesaikan sistem persamaan tak linear dapat menggunakan metode lain, yaitu Newton-Raphson. Jika pada Metode Newton diperlukan nilai $f'(x_i)$ untuk setiap iterasinya, maka pada Metode Newton-Raphson diperlukan matriks Jacobian $J(x_i)$ untuk iterasinya sebagai pengganti dari $f'(x_i)$ di mana

$$J(x_i) = \begin{bmatrix} \frac{\partial f_1(x_i)}{\partial x_1} & \frac{\partial f_1(x_i)}{\partial x_2} & \cdots & \frac{\partial f_1(x_i)}{\partial x_n} \\ \frac{\partial f_2(x_i)}{\partial x_1} & \frac{\partial f_2(x_i)}{\partial x_2} & \cdots & \frac{\partial f_2(x_i)}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n(x_i)}{\partial x_1} & \frac{\partial f_n(x_i)}{\partial x_2} & & \frac{\partial f_n(x_i)}{\partial x_n} \end{bmatrix} \tag{9}$$

dengan syarat $J(x_i)$ adalah matriks tak singular. Sehingga diperoleh rumus Metode Newton-Raphson yaitu:

$$x_{i+1} = x_i - J(x_i)^{-1} F(x_i), \qquad i = 0,1,2,\ldots \tag{10}$$

(Burden & Faires, 2011).

2.5.3 Metode Broyden

Menurut Ramli et al., (2010), Metode Broyden adalah salah satu metode numerik yang digunakan untuk menyelesaikan sistem persamaan tak linear yang dirancang untuk mengembangkan Metode Newton. Metode ini merupakan generalisasi dari Metode Secant untuk persamaan dimensi banyak. Dengan menentukan nilai awal (x_0) terlebih dahulu, kemudian hampiran selanjutnya (x_1) dihitung dengan cara yang sama seperti pada Metode Newton.

Untuk menentukan x_2, Metode Secant menggunakan pendekatan beda hingga

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \tag{11}$$

sebagai pengganti untuk $f'(x_i)$ pada Metode Newton dengan mempertimbangkan $x_1 - x_0$ sebagai sebuah vektor, dan hasil pembagian yang tidak terdefinisi. Pada Metode Broyden, matriks Jacobian $J(x_1)$ diubah menjadi matriks A_1 sebagai berikut:

$$A_1(x_1 - x_0) = F(x_1) - F(x_0). \tag{12}$$

Setiap vektor tak nol pada $\mathbb{R}^n$ dapat ditulis sebagai jumlah dari perkalian $x_1 - x_0$. Pada komplemen ortogonal dari $x_1 - x_0$ dispesifikasi terlebih dahulu untuk mendefinisikan matriks A_1 sebagai berikut:

$$A_1 z = J(x_0)z, \qquad \text{di mana } (x_1 - x_0)^T z = 0. \tag{13}$$

Jadi, setiap vektor ortogonal untuk $x_1 - x_0$ tidak dipengaruhi oleh bentuk pembaruan $J(x_0)$ yang digunakan untuk menghitung x_1. A_1 didefinisikan oleh (12) dan (13) sebagai berikut:

$$A_1 = J(x_0) + \frac{[F(x_1) - F(x_0) - J(x_0)(x_1 - x_0)](x_1 - x_0)^T}{(x_1 - x_0)^T(x_1 - x_0)}. \qquad (14)$$

Untuk menentukan x_2, matriks $J(x_1)$ diganti dengan A_1 sebagai berikut:

$$x_2 = x_1 - A_1^{-1}F(x_1). \qquad (15)$$

Setelah x_1 ditentukan, maka hampiran baru x_{i+1} dapat ditentukan dengan menjalankan formula berikut

$$A_i = A_{i-1} + \frac{y_i - A_{i-1}s_i}{s_i^T s_i} s_i^T, \qquad (16.\text{a})$$

$$x_{i+1} = x_i - A_i^{-1}F(x_i), \qquad (16.\text{b})$$

di mana $y_i = F(x_i) - F(x_{i-1})$ dan $s_i = x_i - x_{i-1}$. Oleh karena itu, jumlah evaluasi fungsional skalar dikurangi dari $n^2 + n$ menjadi n perhitungan yang masih diperlukan untuk menyelesaikan sistem persamaan linear $n \times n$ terkait:

$$A_i s_{i+1} = -F(x_i). \qquad (17)$$

Pembaruan Broyden pada $(16.\text{a})$ membuat perubahan sekecil mungkin untuk matriks Jacobian (sebagaimana diukur dengan norm Euclidean $\|A_{i-1} - A_i\|_2$) yang konsisten dengan $y_i = A_i s_i$, seperti yang ditampilkan pada Lemma berikut.

Lemma 2.5.3.1 (Dennis dan Schnabel)

Di antara semua matriks A yang memenuhi $As_i = y_i$, matriks A_i yang didefinisikan oleh $(16.\text{a})$ meminimalkan selisih $\|A - A_{i-1}\|$.

Bukti

Misalkan A sembarang matriks yang memenuhi $As_i = y_i$. Dengan sifat-sifat norm Euclidean dan fakta bahwa $\|ss^T/s^Ts\| = 1$ untuk sembarang vektor s, maka diperoleh

$$\|A - A_{i-1}\| = \left\|\frac{y_i - A_{i-1}s_i}{s_i^T s_i} s_i^T\right\|$$

$$= \left\|\frac{(A - A_{i-1})s_i}{s_i^T s_i} s_i^T\right\| \leq \|A - A_{i-1}\| \left\|\frac{s_i s_i^T}{s_i^T s_i}\right\| = \|A - A_{i-1}\|.$$

Oleh karena itu, didapat

$$A_i \in \arg \min_{A: y_i = As_i} \|A - A_{i-1}\|$$

dan hasil terbukti.

Teorema 2.5.3.2 (Sherman-Morrison)

Misalkan $A \in \mathbb{R}^{n\times n}$ adalah matriks persegi yang bersifat *invertible* dan $u, v \in \mathbb{R}^n$ adalah vektor kolom. Maka $A + uv^T$ bersifat *invertible* jika dan hanya jika $1 + v^T A^{-1} u \neq 0$. Dalam kasus ini,

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u} \tag{18}$$

Di mana uv^T adalah hasil kali luar dari vektor u dan v (Bartlett, 1991).

Bukti

($\Leftarrow$) Untuk membuktikan dengan arah mundur bahwa $1 + v^T A^{-1} u \neq 0 \Rightarrow A + uv^T$ bersifat *invertible* dengan invers seperti yang ditunjukkan adalah benar, maka diverifikasi sifat-sifat inversnya. Sebuah matriks Y (dalam kasus ini ruas kanan pada formula Sherman-Morrison) adalah invers dari matriks X (dalam kasus ini

$A + uv^T$) jika dan hanya jika $XY = YX = I$. Verifikasi terlebih dulu bahwa ruas kanan (Y) memenuhi $XY = I$.

$$XY = (A + uv^T)\left(A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}\right)$$

$$= AA^{-1} + uv^TA^{-1} - \frac{AA^{-1}uv^TA^{-1} + uv^TA^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}$$

$$= I + uv^TA^{-1} - \frac{uv^TA^{-1} + uv^TA^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}$$

$$= I + uv^TA^{-1} - \frac{u(1+v^TA^{-1}u)v^TA^{-1}}{1+v^TA^{-1}u}$$

$$= I + uv^TA^{-1} - uv^TA^{-1}$$

$$= I$$

Untuk melengkapi pembuktian pada arah ini, perlu ditunjukkan bahwa $YX = I$ dengan cara yang sama seperti di atas:

$$YX = \left(A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}\right)(A + uv^T) = I$$

Faktanya, langkah terakhir tersebut dapat dihindari karena untuk matriks persegi X dan Y, $XY = I$ ekuivalen dengan $YX = I$.

($\Rightarrow$) Sebaliknya, jika $1 + v^TA^{-1}u = 0$ maka dengan menggunakan lemma determinan matriks, yaitu $\det(A + uv^T) = (1 + v^TA^{-1}u)\det(A) = 0$, maka $(A + uv^T)$ tidak bersifat *invertible* (kontradiksi).

Definisi 2.5.3.3 (Hasil Kali Luar)

Diberikan dua vektor berukuran $m \times 1$ dan $n \times 1$ berturut-turut

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

Hasil kali luarnya dinyatakan dengan $u \otimes v$ dan didefinisikan sebagai matriks A berukuran $m \times n$ yang diperoleh dari perkalian pada setiap elemen di u dengan setiap elemen di v sebagai berikut.

$$u \otimes v = A = \begin{bmatrix} u_1v_1 & u_1v_2 & \cdots & u_1v_n \\ u_2v_1 & u_2v_2 & \cdots & u_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_mv_1 & u_mv_2 & \cdots & u_mu_n \end{bmatrix}$$

Dengan menyatakan perkalian titik dengan $\cdot$, jika diberikan vektor w berukuran $n \times 1$ maka $(u \otimes v)w = (v \cdot w)u$. Jika diberikan vektor x berukuran $1 \times m$, maka $x(u \otimes v) = (x \cdot u)v^T$. Hasil kali luar $u \otimes v$ ekuivalen dengan perkalian matriks uv^T, asalkan u merupakan vektor kolom berukuran $m \times 1$ dan v merupakan vektor kolom berukuran $n \times 1$ (dalam v^T berupa vektor baris). Untuk vektor kompleks, sering digunakan untuk mendapatkan transpose konjugat pada v, yang disajikan dengan v^T (Lipschutz & Lipson, 2009):

$$u \otimes v = u(v^T)$$

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

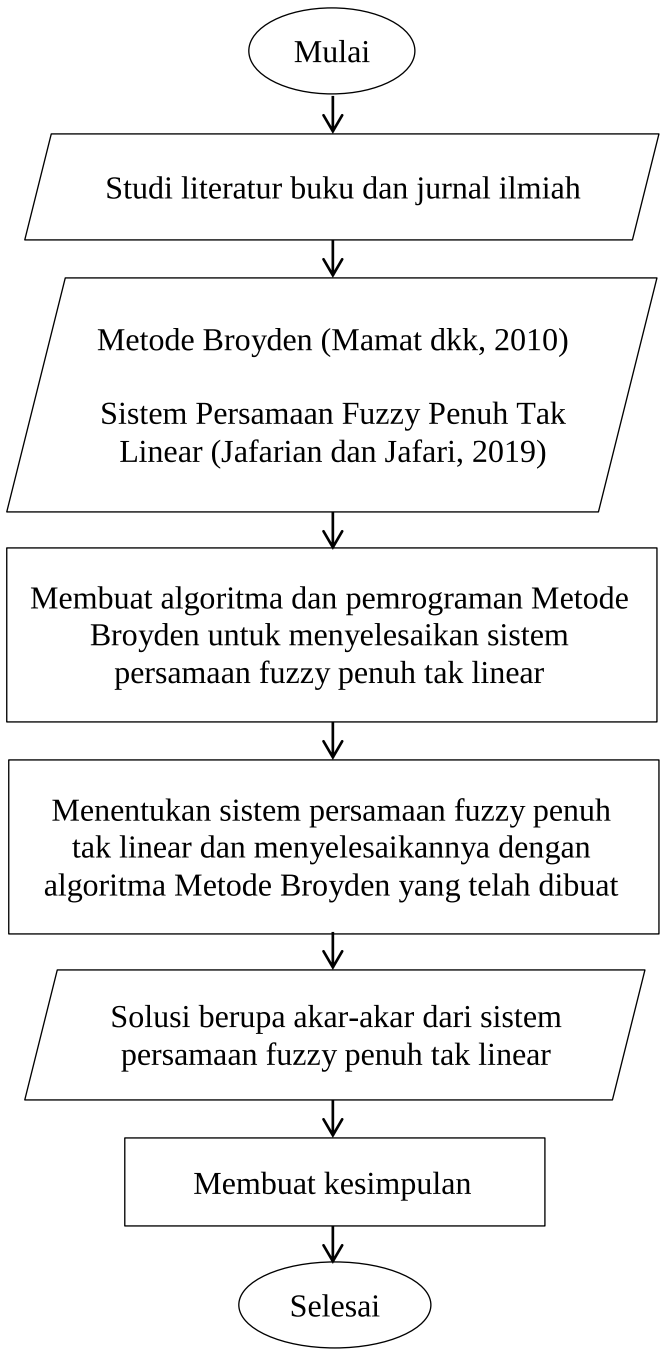
Penelitian ini dilaksanakan pada semester genap tahun akademik 2021/2022, bertempat di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

3.2 Metode Penelitian

Penelitian ini termasuk jenis penelitian studi literatur dengan mencari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Menentukan sistem persamaan fuzzy penuh tak linear.
2. Mengkonstruksi algoritma Metode Broyden untuk menyelesaikan sistem persamaan fuzzy penuh tak linear dan mengimplementasikannya menggunakan pemrograman Matlab.
3. Mendapatkan solusi penyelesaian dari sistem persamaan fuzzy penuh tak linear menggunakan algoritma Metode Broyden yang telah dibuat.
4. Menentukan kesimpulan.

Berikut disajikan diagram alir penelitian yang dilakukan oleh penulis:



Gambar 4. Diagram Alir Penelitian Implementasi Algoritma Broyden untuk Menyelesaikan Sistem Persamaan Fuzzy Penuh Tak Linear.

IV. HASIL DAN PEMBAHASAN

Pada bab ini akan diberikan uraian mengenai Metode Broyden dan beberapa penulisan algoritmanya, serta diberikan dua contoh peenggunaan Metode Broyden untuk menyelesaikan sistem persamaan fuzzy penuh tak linear yang diimplementasikan menggunakan Matlab.

4.1 Metode Broyden pada Sistem Persamaan Tak Linear

Metode Broyden merupakan perkembangan dari Metode Newton yang dapat digunakan untuk menyelesaikan sistem persamaan tak linear. Berdasarkan pada bab sebelumnya, Metode Broyden memiliki formula seperti pada persamaan $(16.\mathrm{a})$ dan $(16.\mathrm{b})$. Untuk mempermudah perhitungan pada setiap iterasi, maka invers matriks A_i pada persamaan $(16.\mathrm{a})$ dapat dicari menggunakan **Teorema 2.5.3.2**, dengan persamaan (18) sebagai berikut.

Misalkan $A = A_{i-1}$

$$u = \frac{y_i - A_{i-1}s_i}{s_i^T s_i}$$

$$v^T = s_i^T$$

Maka $\left(A_{i-1} + \frac{y_i - A_{i-1}s_i}{s_i^T s_i} s_i^T\right)^{-1} = A_{i-1}^{-1} - \frac{A_{i-1}^{-1}\left(\frac{y_i - A_{i-1}s_i}{s_i^T s_i}\right)s_i^T A_{i-1}^{-1}}{1 + s_i^T A_{i-1}^{-1}\left(\frac{y_i - A_{i-1}s_i}{s_i^T s_i}\right)}$

$$= A_{i-1}^{-1} - \frac{\frac{\left(A_{i-1}^{-1}y_i - s_i\right)s_i^T A_{i-1}^{-1}}{s_i^T s_i}}{\frac{s_i^T s_i}{s_i^T s_i} + \frac{s_i^T A_{i-1}^{-1}y_i - s_i^T s_i}{s_i^T s_i}}$$

$$= A_{i-1}^{-1} - \frac{\left(A_{i-1}^{-1}y_i - s_i\right)s_i^T A_{i-1}^{-1}}{s_i^T A_{i-1}^{-1}y_i}$$

$$= A_{i-1}^{-1} + \frac{\left(s_i - A_{i-1}^{-1}y_i\right)s_i^T A_{i-1}^{-1}}{s_i^T A_{i-1}^{-1}y_i}$$

Sehingga diperoleh rumus untuk menghitung invers matriks A_i yaitu

$$A_i^{-1} = A_{i-1}^{-1} + \frac{(\boldsymbol{s}_i - A_{i-1}^{-1}\boldsymbol{y}_i)\boldsymbol{s}_i^T A_{i-1}^{-1}}{\boldsymbol{s}_i^T A_{i-1}^{-1}\boldsymbol{y}_i} \qquad (19)$$

yang akan digunakan pada iterasi kedua dan seterusnya pada Metode Broyden untuk mencari nilai hampiran dari solusi sistem persamaan tak linear.

4.2. Algoritma Broyden untuk Menyelesaikan Sistem Persamaan Fuzzy Penuh Tak Linear

Pada penelitian ini, Metode Broyden digunakan untuk menyelesaikan sistem persamaan fuzzy penuh tak linear. Langkah-langkah dalam menyelesaikan sistem persamaan fuzzy penuh tak linear menggunakan Metode Broyden dapat dikonstruksi ke dalam sebuah algoritma yang kemudian dapat disebut sebagai Algoritma Broyden. Namun sebelum menjalankan Algoritma Broyden, sistem persamaan fuzzy penuh tak linear diubah terlebih dahulu menjadi sistem persamaan tegas tak linear dengan menggunakan operasi perhitungan bilangan fuzzy segitiga pada **Definisi 2.2.4**. Setelah diubah maka sistem persamaan tak linear dapat dicari solusinya menggunakan algoritma berikut.

Masukan: sistem persamaan tak linear yang akan diselesaikan; nilai hampiran awal $x_0 = (x_1, \dots, x_n)^T$; dan toleransi (tol).

Keluaran: solusi hampiran $x = (x_1, \dots, x_n)^T$

Perhitungan inti:

Langkah 1. Hitung matriks Jacobian $J(x)$.

Langkah 2. Tentukan $A_0 = J(x_0)$ (substitusi nilai awal ke dalam matriks Jacobian).

Tentukan $F(x_0)$ (substitusi nilai awal ke dalam sistem persamaan).

Langkah 3. Verifikasi bahwa $\det(A_0) \neq 0$. Jika ya, maka proses perhitungan berlanjut ke langkah berikutnya. Jika tidak, maka masukkan nilai hampiran awal lain.

Langkah 4. Tentukan A_0^{-1}.

Langkah 5. Hitung nilai hampiran untuk iterasi pertama (x_1) menggunakan formula Newton: $x_1 = x_0 - A_0^{-1}F(x_0)$.

Langkah 6. Perbarui nilai hampiran $x = x_1$

Hitung galat: $err = \|x_1\|$

Tentukan iterasi $k = 1$

Langkah 7. Ketika galat $>$ toleransi, lakukan Langkah 8 – 12.

Langkah 8. Hitung: $y_k = F(x_k) - F(x_{k-1})$

$$s_k = x_k - x_{k-1} = -A_{k-1}^{-1}F(x_{k-1})$$

$$p_k = s_k^T A_{k-1}^{-1} y_k$$

Langkah 9. Perbarui invers matriks Jacobian dengan formula $A_k^{-1} = A_{k-1}^{-1} + \frac{1}{p}(s_k - A_{k-1}^{-1}y_k)s_k^T A_{k-1}^{-1}$

Langkah 10. Hitung nilai hampiran selanjutnya dengan formula $x_{k+1} = x_k - A_k^{-1}F(x_k)$

Langkah 11. Perbarui galat: $err = \|x_{k+1} - x_k\|$

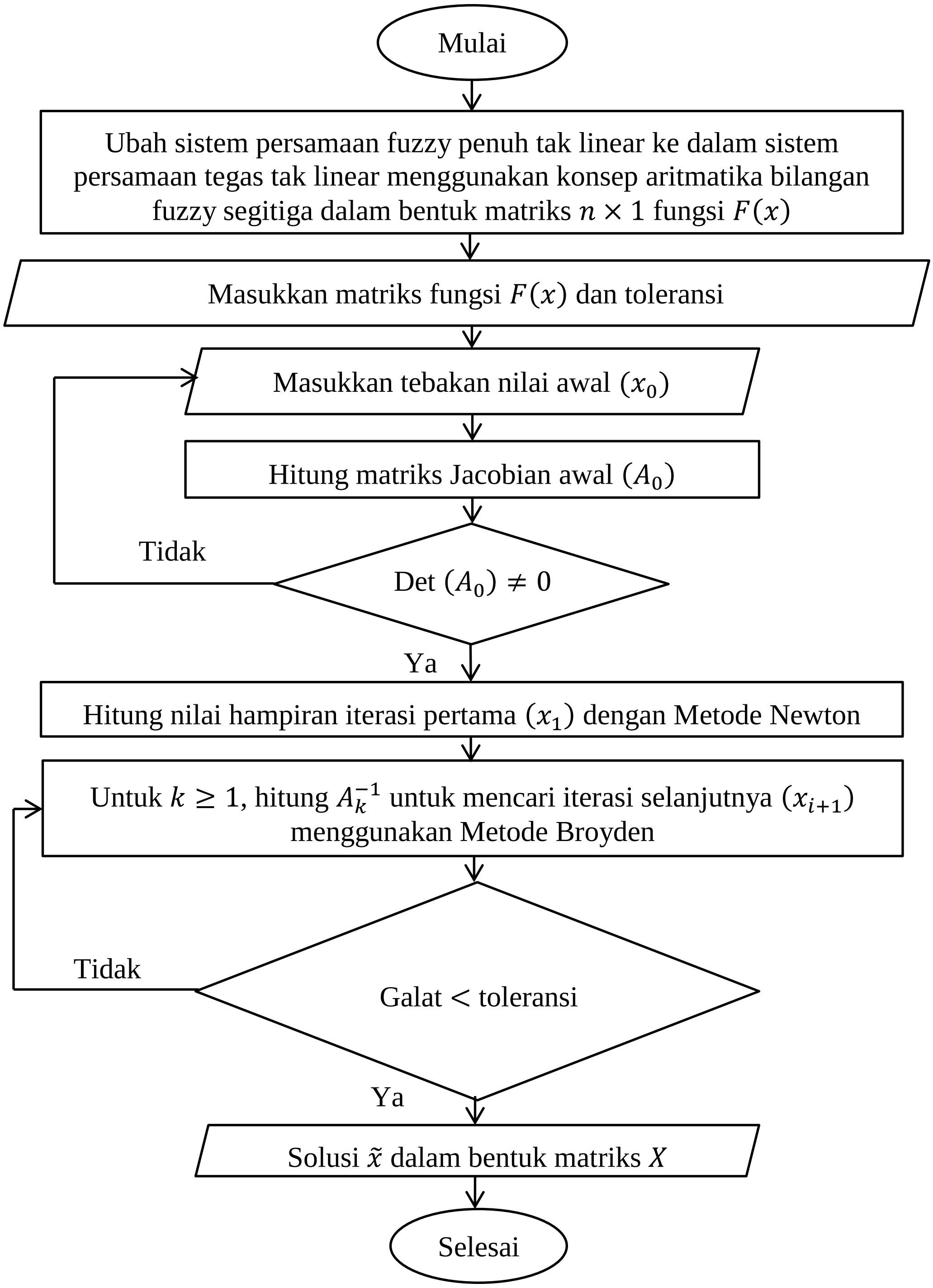
Langkah 12. Perbarui iterasi: $k = k + 1$

Langkah 13. Berhenti (jika kondisi sudah terpenuhi).

Algoritma Broyden yang telah dikonstruksi dapat ditulis dalam beberapa penulisan algoritma, seperti diagram alir dan *pseudocode* yang disajikan dalam subbab berikut.

4.2.1. Diagram Alir Algoritma Broyden untuk Menyelesaikan Sistem Persamaan Fuzzy Penuh Tak Linear

Gambar 5 berikut menggambarkan Algoritma Broyden untuk menyelesaikan sistem persamaan fuzzy penuh tak linear dalam bentuk diagram alir.



Gambar 5. Diagram Alir Algoritma Broyden

4.2.2. *Pseudocode* Algoritma Broyden

Berikut ini diberikan *pseudocode* menyelesaikan sistem persamaan fuzzy penuh tak linear yang telah diubah ke dalam sistem persamaan tegas tak linear menggunakan Algoritma Broyden.

{Diberikan sebuah sistem persamaan tak linear. Ditentukan suatu tebakan nilai awal x_0, dan toleransi. Solusi untuk sistem persamaan tak linear dicari menggunakan Metode Broyden.}

Functionalgorithm_Broyden (f, x: array [1..6] of real; A0, A: array [1..6, 1..6] of real; x_1, s, y, F, F_1: array [1..2] of real; st, ut: array [1..1, 1..6] of real; X: array [1..k, 1..6] of real; eps, p: real; k: integer)

Read

(f, x, eps, k)

begin

jac ← jacobian (f, [x1 x2 x3 y1 y2 y3], x);

A0 ← subs (jac, [x1; x2; x3; y1; y2; y3], x);

if det(A0) ≠ 0 do

begin

A ← double (inv(A0));

F ← double (subs (f, [x1; x2; x3; y1; y2; y3], x));

s ← -A * F;

x_1 ← x + s;

X ← [x'; x_1'];

x ← x_1;

```
        while (err > eps) do
        begin
        F_1 ← F;
        F ← double (subs (f, [x1; x2; x3; y1; y2; y3], x));
        y ← double (F - F_1);
        st ← transpose (s);
        p ← -st * (-A * y);
        ut ← st * A;
        A ← A + (1/p) * (s –A * y) * ut;
        s ← -A * F;
        x_new ← x + s;
        X ← [X; x_new'];
        err ← norm (x_new – x);
        x ← x_new;
        k ← k + 1;
        endwhile
write (X)
else
write (‘matriks A0 singular. Masukkan ulang nilai tebakan awal.’)
endif
endfunction
{Kondisi akhir pengulangan : err < eps, maka solusinya berupa matriks X.}
```

4.3. Implementasi Algoritma Broyden untuk Menyelesaikan Sistem Persamaan Fuzzy Penuh Tak Linear

Berikut ini akan disajikan dua contoh penyelesaian sistem persamaan fuzzy penuh tak linear pada Jafarian & Jafari (2019) secara numerik menggunakan Algoritma Broyden.

Contoh 4.3.1

Tinjau sistem persamaan fuzzy penuh tak linear berikut

$$(2,3,5) \circledast \tilde{x} \oplus (2,4,5) \circledast \tilde{y} \oplus (1,2,3) \circledast \tilde{x}^2 \oplus (3,5,6) \circledast \tilde{y}^2 = (19,140,467)$$
$$(1,2,3) \circledast \tilde{x} \oplus (3,4,6) \circledast \tilde{y} \oplus (3,4,5) \circledast \tilde{x}^2 \oplus (1,3,4) \circledast \tilde{y}^2 = (14,136,436)$$

dengan $\tilde{x}$ dan $\tilde{y}$ adalah dua bilangan fuzzy segitiga. Diberikan batas toleransi sebesar 10^{-2} dan nilai awal berupa dua bilangan fuzzy segitiga $\tilde{x} = (1,2,3)$ dan $\tilde{y} = (1,2,3)$. Maka solusi sistem persamaan fuzzy penuh tak linear tersebut dapat dicari dengan Metode Broyden sebagai berikut.

Penyelesaian

Asumsikan bahwa $\tilde{x} = (x_1, x_2, x_3)$ dan $\tilde{y} = (y_1, y_2, y_3)$. Maka sistem persamaan fuzzy penuh tak linear yang diberikan dapat ditulis sebagai berikut.

$$\begin{cases} (2,3,5) \circledast (x_1, x_2, x_3) \oplus (2,4,5) \circledast (y_1, y_2, y_3) \oplus (1,2,3) \circledast (x_1^2, x_2^2, x_3^2) \\ \oplus\, (3,5,6) \circledast (y_1^2, y_2^2, y_3^2) = (19,140,467) \\ (1,2,3) \circledast (x_1, x_2, x_3) \oplus (3,4,6) \circledast (y_1, y_2, y_3) \oplus (3,4,5) \circledast (x_1^2, x_2^2, x_3^2) \\ \oplus\, (1,3,4) \circledast (y_1^2, y_2^2, y_3^2) = (14,136,436) \end{cases}$$

Dengan menerapkan aritmatika bilangan fuzzy segitiga, sistem persamaan fuzzy penuh tak linear di atas diubah ke dalam bentuk sistem persamaan tegas tak linear sehingga sistem persamaan dapat ditulis sebagai berikut.

$$\begin{cases} 2x_1 + 2y_1 + x_1^2 + 3y_1^2 = 19 \\ 3x_2 + 4y_2 + 2x_2^2 + 5y_2^2 = 140 \\ 5x_3 + 5y_3 + 3x_3^2 + 6y_3^2 = 467 \\ x_1 + 3y_1 + 3x_1^2 + y_1^2 = 14 \\ 2x_2 + 4y_2 + 4x_2^2 + 3y_2^2 = 136 \\ 3x_3 + 6y_3 + 5x_3^2 + 4y_3^2 = 436 \end{cases}$$

Berdasarkan algoritma yang telah dikonstruksi, dapat ditentukan beberapa unsur masukan yaitu

$$F(x) = \begin{bmatrix} 2x_1 + 2y_1 + x_1^2 + 3y_1^2 - 19 \\ 3x_2 + 4y_2 + 2x_2^2 + 5y_2^2 - 140 \\ 5x_3 + 5y_3 + 3x_3^2 + 6y_3^2 - 467 \\ x_1 + 3y_1 + 3x_1^2 + y_1^2 - 14 \\ 2x_2 + 4y_2 + 4x_2^2 + 3y_2^2 - 136 \\ 3x_3 + 6y_3 + 5x_3^2 + 4y_3^2 - 436 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

$\text{tol} = 10^{-2}$

$N = 100$

Langkah 1. $J(x) = \begin{bmatrix} 2x_1 + 2 & 0 & 0 & 6y_1 + 2 & 0 & 0 \\ 0 & 4x_2 + 3 & 0 & 0 & 10y_2 + 4 & 0 \\ 0 & 0 & 6x_3 + 5 & 0 & 0 & 12y_3 + 5 \\ 6x_1 + 1 & 0 & 0 & 2y_1 + 3 & 0 & 0 \\ 0 & 8x_2 + 2 & 0 & 0 & 6y_2 + 4 & 0 \\ 0 & 0 & 10x_3 + 3 & 0 & 0 & 8y_3 + 6 \end{bmatrix}$

Langkah 2. $A_0 = \begin{bmatrix} 4 & 0 & 0 & 8 & 0 & 0 \\ 0 & 11 & 0 & 0 & 24 & 0 \\ 0 & 0 & 23 & 0 & 0 & 41 \\ 7 & 0 & 0 & 5 & 0 & 0 \\ 0 & 18 & 0 & 0 & 16 & 0 \\ 0 & 0 & 33 & 0 & 0 & 30 \end{bmatrix}$ dan $F(x_0) = \begin{bmatrix} -11 \\ -98 \\ -356 \\ -6 \\ -96 \\ -328 \end{bmatrix}$

Langkah 3. Dengan eliminasi Gauss diperoleh $\det(A_0) = -6110208$.

Langkah 4. $$A_0^{-1} = \begin{bmatrix} -0.1389 & 0 & 0 & 0.2222 & 0 & 0 \\ 0 & -0.0625 & 0 & 0 & 0.0938 & 0 \\ 0 & 0 & -0.0452 & 0 & 0 & 0.0618 \\ 0.1944 & 0 & 0 & -0.1111 & 0 & 0 \\ 0 & 0.0703 & 0 & 0 & -0.0430 & 0 \\ 0 & 0 & 0.0498 & 0 & 0 & -0.0347 \end{bmatrix}$$

Langkah 5. Menghitung iterasi pertama: $x_1 = x_0 - A_0^{-1}F(x_0) = \begin{bmatrix} 0.8056 \\ 4.8750 \\ 7.1750 \\ 2.4722 \\ 4.7656 \\ 9.3409 \end{bmatrix}$

Langkah 6. Hitung galat: $err = \|x_1\| = 13.8553$

Ketika $k = 1$ (iterasi kedua):

Langkah 7. $$y_1 = F(x_1) - F(x_0) = \begin{bmatrix} 17.5401 \\ 152.7747 \\ 649.5311 \\ 8.2809 \\ 152.0085 \\ 575.9783 \end{bmatrix}$$

$$s_1 = x_1 - x_0 = \begin{bmatrix} -0.1944 \\ 2.8750 \\ 4.1750 \\ 1.4722 \\ 2.7656 \\ 6.3409 \end{bmatrix}$$

$$p_1 = s_1^T A_0^{-1} y_1 = 133.2478$$

Langkah 8. Perbarui invers matriks Jacobian:

$$A_1^{-1} = A_0^{-1} + \frac{1}{p_1}(s_1 - A_0^{-1}y_1)s_1^T A_0^{-1}$$

$$= \begin{bmatrix} -0.1379 & 0 & 0.0004 & 0.2216 & 0.0005 & 0.0001 \\ -0.0043 & -0.0627 & -0.0017 & 0.0028 & 0.0917 & -0.0005 \\ -0.0048 & -0.0002 & -0.0472 & 0.0032 & -0.0023 & 0.0613 \\ 0.1921 & -0.0001 & -0.0010 & -0.1095 & -0.0012 & -0.0003 \\ -0.0034 & 0.0702 & -0.0014 & 0.0022 & -0.0446 & -0.0004 \\ -0.0141 & -0.0007 & 0.0441 & 0.0093 & -0.0068 & -0.0364 \end{bmatrix}$$

Langkah 9. $x_2 = x_1 - A_1^{-1}F(x_1) = \begin{bmatrix} 1.0338 \\ 3.8361 \\ 6.0077 \\ 1.8933 \\ 3.9442 \\ 5.9253 \end{bmatrix}$

Langkah 10. Perbarui galat: $err = \|x_2 - x_1\| = 3.8948$

Karena $3.8948 > 10^{-2}$, maka proses perhitungan dilanjutkan dengan mengulang langkah 7 – 10 untuk menghitung iterasi selanjutnya, yaitu ketika $k = 2$ (iterasi ketiga):

Langkah 7. $y_2 = F(x_2) - F(x_1) = \begin{bmatrix} -7.8633 \\ -60.2736 \\ -381.9304 \\ -2.7762 \\ -63.0269 \\ -309.5016 \end{bmatrix}$

$$s_2 = x_2 - x_1 = \begin{bmatrix} 0.2283 \\ -1.0389 \\ -1.1672 \\ -0.5789 \\ -0.8214 \\ -3.4155 \end{bmatrix}$$

$$p_2 = s_2^T A_1^{-1} y_2 = 20.2065$$

Langkah 8. $A_2^{-1} = A_1^{-1} + \frac{1}{p_2}(s_2 - A_1^{-1}y_2)s_2^T A_1^{-1}$

$$= \begin{bmatrix} -0.1378 & 0 & 0.0005 & 0.2215 & 0.0005 & 0 \\ -0.0047 & -0.0626 & -0.0022 & 0.0032 & 0.0915 & -0.0002 \\ -0.0031 & -0.0004 & -0.0453 & 0.0016 & -0.0016 & 0.0601 \\ 0.1917 & -0.0001 & -0.0014 & -0.1092 & -0.0013 & -0.0001 \\ -0.0031 & 0.0701 & -0.0010 & 0.0020 & -0.0445 & -0.0006 \\ -0.0205 & -0.0001 & 0.0368 & 0.0151 & -0.0093 & -0.0322 \end{bmatrix}$$

Langkah 9. $x_3 = x_2 - A_2^{-1}F(x_2) = \begin{bmatrix} 1.0124 \\ 3.9176 \\ 5.6875 \\ 1.9591 \\ 3.8860 \\ 7.1185 \end{bmatrix}$

Langkah 10. Perbarui galat: $err = \|x_3 - x_2\| = 1.2414$.

Karena $1.2414 > 10^{-2}$, maka proses perhitungan dilanjutkan dengan cara yang sama untuk menghitung iterasi berikutnya. Proses perhitungan berhenti ketika nilai galat $< 10^{-2}$, yaitu ketika $k = 7$ (iterasi kedelapan) dengan nilai hampiran yang diperoleh yaitu $x_8 = \begin{bmatrix} 1 \\ 4.0006 \\ 5.9999 \\ 2 \\ 3.9998 \\ 6.9999 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 4 \\ 6 \\ 2 \\ 4 \\ 7 \end{bmatrix}$ atau jika dikembalikan ke dalam bentuk bilangan fuzzy segitiga dapat ditulis sebagai $\tilde{x} = (1,4,6)$ dan $\tilde{y} = (2,4,7)$, dengan galat sebesar 0.0018.

Algoritma Broyden untuk menyelesaikan sistem persamaan fuzzy penuh tak linear pada Contoh 4.3.1 tersebut dapat diimplementasikan menggunakan Matlab, dengan menurunkan batas toleransinya menjadi 10^{-8}. Solusi sistem persamaan tersebut diperoleh pada iterasi ke-15 dengan galat sebesar 2.87×10^{-9} dalam waktu *running* 2.78 detik dan ditampilkan dalam Tabel 2 berikut.

Tabel 2. Solusi Penyelesaian Sistem Persamaan Contoh 4.3.1

Iterasi ke-	x_1	x_2	x_3	y_1	y_2	y_3
0	1.0000	2.0000	3.0000	1.0000	2.0000	3.0000
1	0.8056	4.8750	7.1750	2.4722	4.7656	9.3409
2	1.0338	3.8361	6.0077	1.8933	3.9442	5.9253
3	1.0124	3.9176	5.6875	1.9591	3.8860	7.1185
⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	1.0000	4.0000	6.0000	2.0000	4.0000	7.0000

Solusi pada sistem persamaan fuzzy penuh tak linear yang diperoleh pada Contoh 4.3.1 tersebut dapat direpresentasikan dengan fungsi keanggotaan yang disajikan dalam Tabel 3 dan ditampilkan pada Gambar 6 a dan 6 b..

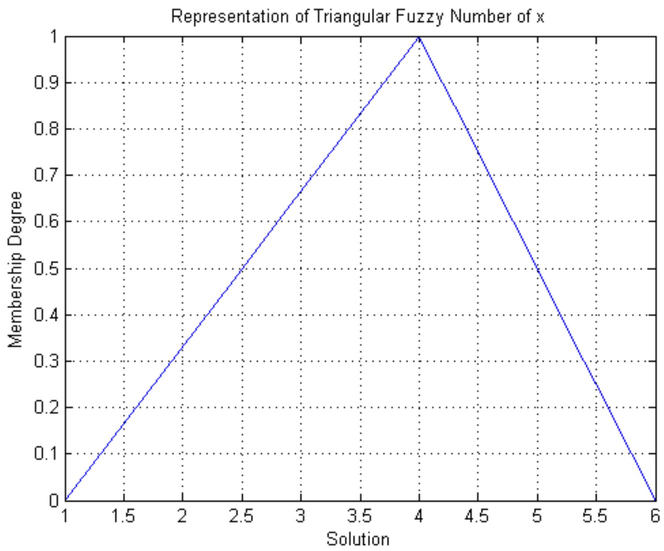
Tabel 3. Fungsi Keanggotaan Solusi dari Contoh 4.3.1.

$\boldsymbol{\mu}$	$\boldsymbol{x(\mu)}$		$\boldsymbol{y(\mu)}$	
	$\underline{\boldsymbol{x}}(\boldsymbol{\mu})$	$\overline{\boldsymbol{x}}(\boldsymbol{\mu})$	$\underline{\boldsymbol{y}}(\boldsymbol{\mu})$	$\overline{\boldsymbol{y}}(\boldsymbol{\mu})$
0	1.0000	6.0000	2.0000	7.0000
0.1	1.3000	5.8000	2.2000	6.7000
0.2	1.6000	5.6000	2.4000	6.4000
0.3	1.9000	5.4000	2.6000	6.1000
0.4	2.2000	5.2000	2.8000	5.8000
0.5	2.5000	5.0000	3.0000	5.5000
0.6	2.8000	4.8000	3.2000	5.2000
0.7	3.1000	4.6000	3.4000	4.9000
0.8	3.4000	4.4000	3,6000	4.6000
0.9	3.7000	4.2000	3.8000	4.3000
1	4.0000	4.0000	4.0000	4.0000

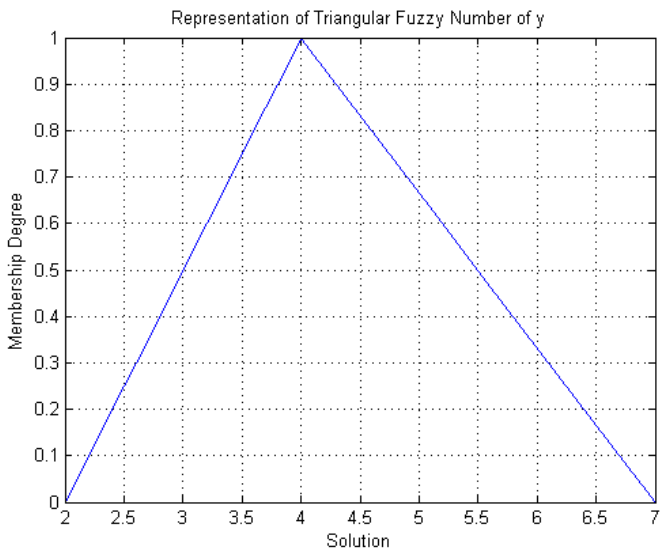
Tabel 3 menunjukkan fungsi keanggotaan dari solusi yang didapat dengan mensubstitusi derajat keanggotaan (μ) ke dalam bentuk parametrik dari sistem persamaan fuzzy penuh tak linear seperti berikut.

$$\begin{cases}(2+\mu)\underline{x}(\mu)+(2+2\mu)\underline{y}(\mu)+(1+\mu)\underline{x}^2(\mu)+(3+2\mu)\underline{y}^2(\mu)=(19+121\mu)\\(5-2\mu)\overline{x}(\mu)+(5-\mu)\overline{y}(\mu)+(3-\mu)\overline{x}^2(\mu)+(3-\mu)\overline{y}^2(\mu)=(467-327\mu)\end{cases}$$

$$\begin{cases}(1+\mu)\underline{x}(\mu)+(3+\mu)\underline{y}(\mu)+(3+\mu)\underline{x}^2(\mu)+(1+2\mu)\underline{y}^2(\mu)=(14+122\mu)\\(3-\mu)\overline{x}(\mu)+(6-2\mu)\overline{y}(\mu)+(5-\mu)\overline{x}^2(\mu)+(4-\mu)\overline{y}^2(\mu)=(436-300\mu)\end{cases}$$



Gambar 6 a. Representasi Bilangan Fuzzy Segitiga $\tilde{x}$ pada Contoh 3.4.1.



Gambar 6 b. Representasi Bilangan Fuzzy Segitiga $\tilde{y}$ pada Contoh 3.4.1.

Gambar 6 a menunjukkan bahwa solusinya berupa bilangan fuzzy segitiga di mana solusi $\tilde{x}$ memiliki nilai minimum 1, nilai rata-rata 4, dan nilai maksimum 6. Sedangkan Gambar 6 b menunjukkan bahwa solusinya berupa bilangan fuzzy segitiga di mana solusi $\tilde{y}$ memiliki nilai minimum 2, nilai rata-rata 4, dan nilai maksimum 7.

Contoh 4.3.2

Tinjau sistem persamaan fuzzy penuh tak linear berikut

$$(3,4,5) \hat{*} \tilde{x} \oplus (1,2,4) \hat{*} \tilde{y} \oplus (-5,-4,-2) \hat{*} \tilde{x}^2 \oplus (2,4,5) \hat{*} \tilde{y}^2$$
$$\oplus (2,3,5) \hat{*} \tilde{x}^3 \oplus (-3,-2,-1) \hat{*} \tilde{y}^3 = (-299,132,1180)$$
$$(-3,-2,-1) \hat{*} \tilde{x} \oplus (-4,-2,-1) \hat{*} \tilde{y} \oplus (1,2,3) \hat{*} \tilde{x}^2 \oplus (-4,-3,-1) \hat{*} \tilde{y}^2$$
$$\oplus (-5,-3,-2) \hat{*} \tilde{x}^3 \oplus (3,4,5) \hat{*} \tilde{y}^3 = (-1145,-93,365)$$

dengan $\tilde{x}$ dan $\tilde{y}$ adalah dua bilangan fuzzy segitiga. Diberikan batas toleransi sebesar 10^{-2} dan nilai awal berupa dua bilangan fuzzy segitiga $\tilde{x} = (2.5, 3.5, 4.5)$ dan $\tilde{y} = (1,2,3)$. Maka solusi sistem persamaan fuzzy penuh tak linear tersebut dapat dicari dengan Metode Broyden sebagai berikut.

Penyelesaian

Asumsikan bahwa $\tilde{x} = (x_1, x_2, x_3)$ dan $\tilde{y} = (y_1, y_2, y_3)$. Maka sistem persamaan fuzzy penuh tak linear yang diberikan dapat ditulis sebagai berikut.

$$\begin{cases} (3,4,5) \hat{*} (x_1,x_2,x_3) \oplus (1,2,4) \hat{*} (y_1,y_2,y_3) \oplus (-5,-4,-2) \hat{*} (x_1^2,x_2^2,x_3^2) \\ \oplus (2,4,5) \hat{*} (y_1^2,y_2^2,y_3^2) \oplus (2,3,5) \hat{*} (x_1^3,x_2^3,x_3^3) \oplus (-3,-2,-1) \\ \hat{*} (y_1^3,y_2^3,y_3^3) = (-299,132,1180) \\ (-3,-2,-1) \hat{*} (x_1,x_2,x_3) \oplus (-4,-2,-1) \hat{*} (y_1,y_2,y_3) \oplus (1,2,3) \\ \hat{*} (x_1^2,x_2^2,x_3^2) \oplus (-4,-3,-1) \hat{*} (y_1^2,y_2^2,y_3^2) \oplus (-5,-3,-2) \hat{*} (x_1^3,x_2^3,x_3^3) \\ \oplus (3,4,5) \hat{*} (y_1^3,y_2^3,y_3^3) = (-1145,-93,365) \end{cases}$$

Dengan mengaplikasikan aritmatika bilangan fuzzy segitiga, maka sistem persamaan fuzzy penuh tak linear di atas dapat ditulis sebagai berikut.

$$\begin{cases} 3x_1 + y_1 - 5x_3^2 + 2y_1^2 + 2x_1^3 - 3y_3^3 = -299 \\ 4x_2 + 2y_2 - 4x_2^2 + 4y_2^2 + 3x_2^3 - 2y_2^3 = 132 \\ 5x_3 + 4y_3 - 2x_1^2 + 5y_3^2 + 5x_3^3 - y_1^3 = 1180 \\ -3x_3 - 4y_3 + x_1^2 - 4y_3^2 - 5x_3^3 + 3y_1^3 = -1145 \\ -2x_2 - 2y_2 + 2x_2^2 - 3y_2^2 - 3x_2^3 + 4y_2^3 = -93 \\ -x_1 - y_1 + 3x_3^2 - y_1^2 - 2x_1^3 + 5y_3^3 = 365 \end{cases}$$

Dengan mengikuti langkah-langkah yang sama pada contoh sebelumnya, maka solusi sistem persamaan fuzzy penuh tak linear pada Contoh 4.3.2 diperoleh pada iterasi kesembilan dengan solusi penyelesaian berupa dua bilangan fuzzy segitiga yaitu $\tilde{x} = (3,4,6)$ dan $\tilde{y} = (2,3,4)$ dengan galat sebesar 0.0028. Jika diimplementasi menggunakan Matlab dan batas toleransi diturunkan menjadi 10^{-8}, maka solusi sistem persamaan fuzzy penuh tak linear pada Contoh 4.3.2 diperoleh pada iterasi ke-17 dengan galat sebesar 2.92×10^{-10} dalam waktu *running* 3.17 detik dan ditampilkan dalam Tabel 4 berikut.

Tabel 4. Solusi Penyelesaian Sistem Persamaan Contoh 4.3.2.

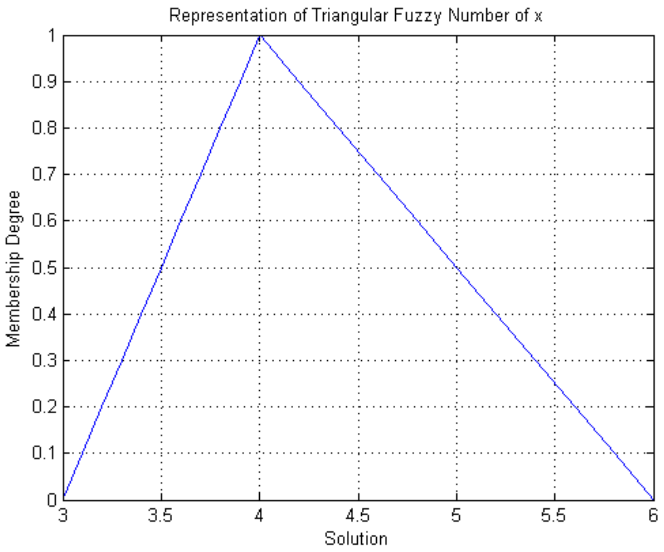
Iterasi ke-	x_1	x_2	x_3	y_1	y_2	y_3
0	2.5000	3.5000	4.5000	1.0000	2.0000	3.0000
1	3.4093	4.0043	6.5267	3.2024	3.5197	4.4214
2	3.3463	4.0530	6.1405	0.1668	3.0238	4.1233
3	2.8387	4.0162	5.8997	1.3096	2.8973	3.8994
⋮	⋮	⋮	⋮	⋮	⋮	⋮
17	3.0000	4.0000	6.0000	2.0000	3.0000	4.0000

Solusi pada sistem persamaan fuzzy penuh tak linear yang diperoleh pada Contoh 4.3.2 tersebut dapat direpresentasikan dengan fungsi keanggotaan yang disajikan dalam Tabel 5 dan ditampilkan pada Gambar 7 a dan 7 b.

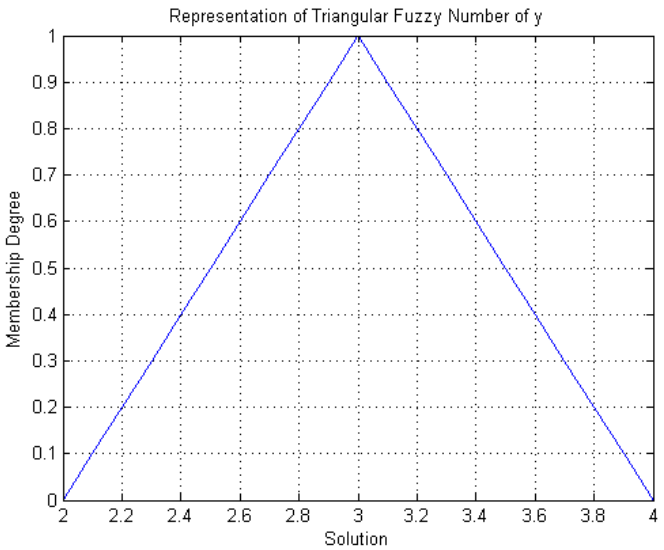
Tabel 5. Fungsi Keanggotaan Solusi dari Contoh 4.3.2.

$\boldsymbol{\mu}$	$\boldsymbol{x(\mu)}$		$\boldsymbol{y(\mu)}$	
	$\underline{\boldsymbol{x}}(\boldsymbol{\mu})$	$\overline{\boldsymbol{x}}(\boldsymbol{\mu})$	$\underline{\boldsymbol{y}}(\boldsymbol{\mu})$	$\overline{\boldsymbol{y}}(\boldsymbol{\mu})$
0	3.0000	6.0000	2.0000	4.0000
0.1	3.1000	5.8000	2.1000	3.9000
0.2	3.2000	5.6000	2.2000	3.8000
0.3	3.3000	5.4000	2.3000	3.7000
0.4	3.4000	5.2000	2.4000	3.6000
0.5	3.5000	5.0000	2.5000	3.5000

μ	$x(\mu)$		$y(\mu)$	
	$\underline{x}(\mu)$	$\overline{x}(\mu)$	$\underline{y}(\mu)$	$\overline{y}(\mu)$
0.6	3.6000	4.8000	2.6000	3.4000
0.7	3.7000	4.6000	2.7000	3.3000
0.8	3.8000	4.4000	2.8000	3.2000
0.9	3.9000	4.2000	2.9000	3.1000
1	4.0000	4.0000	3.0000	3.0000



Gambar 7 a. Representasi Bilangan Fuzzy Segitiga $\tilde{x}$ pada Contoh 3.4.2.



Gambar 7 b. Representasi Bilangan Fuzzy Segitiga $\tilde{y}$ pada Contoh 3.4.2.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Sistem persamaan fuzzy penuh tak linear dapat diselesaikan menggunakan Metode Broyden dengan melibatkan aritmatika bilangan fuzzy segitiga, yang kemudian metode tersebut dikonstruksi menjadi Algoritma Broyden dan ditulis ke dalam tiga penulisan algoritma, yaitu kalimat deskriptif, diagram alir, dan *pseudocode*. Algoritma Broyden yang telah dikonstruksi juga dapat diimplementasikan menggunakan pemrograman Matlab agar lebih mudah digunakan untuk menyelesaikan sistem persamaan fuzzy penuh tak linear dengan kesalahan yang relatif kecil dalam waktu yang singkat. Solusi yang dihasilkan dalam dua contoh yang diberikan masing-masing berupa dua bilangan fuzzy segitiga positif yaitu $\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} 1,4,6 \\ 2,4,7 \end{pmatrix}$ dan $\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} 3,4,6 \\ 2,3,4 \end{pmatrix}$.

5.2. Saran

Metode Broyden yang diteliti dapat dikembangkan untuk memecahkan masalah yang berkaitan dengan himpunan fuzzy *hesitant*.

DAFTAR PUSTAKA

Bartlett, M. S. (1991). An Inverse Matrix Adjustment Arising in Discriminant Analysis. *Annals of Statistics*, *22*(1), 107–111.

Burden, R. L., & Faires, J. D. (2005). *Numerical Analysis Eighth Edition* (8th ed.). THOMSON.

Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis* (M. Julet (ed.); 9th ed.). BROOKS/COLE.

Deswita, Z., & Mashadi. (2019). Alternative Multiplying Triangular Fuzzy Number and Applied in Fully Fuzzy Linear System. *Asrjets*, *56*(2006), 113–123.

Dugopolski, M. (2006). *Elementary and Intermediate Algebra Second Edition*. New York: McGraw-Hill.

Inearat, L., & Qatanani, N. (2018). Numerical methods for solving fuzzy linear systems. *Mathematics*, *6*(2).

Jafari, R., Razvarz, S., & Gegov, A. (2020). A Novel Technique for Solving Fully Fuzzy Nonlinear. *Vietnam Journal of Computer Science*, *7*(1), 93–107.

Jafarian, A., & Jafari, R. (2019). A new computational method for solving fully fuzzy nonlinear matrix equations. *International Journal of Fuzzy Computation and Modelling*, *2*(4), 275–285.

Liliana, B., Castaneda, B., Arunachalam, V., & Dharmaraja, S. (1973). *Introduction To Mathematical*. 497–531.

Lipschutz, S., & Lipson, M. (2009). Schaum's Outline of Linear Algebra, 4ed. In *MC Graw Hill*.

Malkawi, G., Ahmad, N., & Ibrahim, H. (2015). An algorithm for a positive solution of arbitrary fully fuzzy linear system. *Computational Mathematics and Modeling*, *26*(3), 436–465.

Marzuki, C. C., Agustian`, A., Hariati, D., Afmilda, J., Husna, N., & Nanda, P. (2018). Penyelesaian Sistem Persamaan Linier Fully Fuzzy Menggunakan Metode Dekomposisi Nilai Singular (SVD). *Jurnal Matematika "MANTIK," 4*(2), 143–149.

Megarani, W., Zakaria, L., Sutrisno, A., Azis, D., & Nuryaman, A. (2022). Algorithms and Programming: The Jacobi Method For Solving Dual Fully Fuzzy Linear Systems. *Recent Advances in Computer Science and Communications*, *15*.

Ramli, A., Abdullah, M. L., & Mamat, M. (2010). Broyden's method for solving fuzzy nonlinear equations. *Advances in Fuzzy Systems*, *2010*.

Retta, A. M., Isroqmi, A., & Nopriyanti, T. D. (2020). Pengaruh Penerapan Algoritma Terhadap Pembelajaran Pemrograman Komputer. *Indiktika : Jurnal Inovasi Pendidikan Matematika*, *2*(2), 126.

Safitri, Y., & Mashadi. (2019). Alternative fuzzy algebra to solve dual fully fuzzy linear system using st decomposition method. *The Internasional Organization of Scientific Research-Journal of Mathematics*, *15*(2), 32–38.

Senthilkumar, P., & Rajendran, G. (2011). New approach to solve symmetric fully fuzzy linear systems. *Sadhana - Academy Proceedings in Engineering Sciences*, *36*(6), 933–940.

Triatmodjo, B. (2002). *Metode Numerik Dilengkapi dengan Program Komputer*. BETA OFFSET.

Windarto, A. P., Harumy, H., & Sulistianingsih, I. (2016). Belajar Dasar Algoritma dan Pemrograman C++. *Journal of Chemical Information and Modeling*, *August 2017*, 1–209.

Yang, W. Y., Cao, W., Chung, T.-S., & Morris, J. (2005). *Applied Numerical Methods Using Matlab*. John Wiley & Sons, Inc.

Ziqan, A., Ibrahim, S., Marabeh, M., & Qarariyah, A. (2021). Fully fuzzy linear systems with trapezoidal and hexagonal fuzzy numbers. *Granular Computing*, *6*(1998).