Algoritma Penyelesaian Persamaan *Nonlinear Fuzzy* dengan Metode Modifikasi Newton-Raphson

Wahyu Megarani[1], Dorrah Aziz[1], Amanto[1], La Zakaria[1*]

[1]Jurusan Matematika, FMIPA Universitas Lampung
Jl. Prof. Soemantri Brojonegoro No 1, Gedong Meneng, Bandar Lampung

***Email korespondensi:** lazakaria1969@fmipa.unila.ac.id

Abstrak

Persamaan *fuzzy nonlinear* merupakan permasalahan matematika yang dapat dijumpai dalam banyak aplikasi matematika, misalnya robotik dan teknik. Persamaan tersebut dapat diselesaikan dengan metode metode numerik. Terdapat sejumlah metode numerik yang dapat digunakan untuk menylesaikan persamaan *fuzzy nonlinear*, misalnya metode Modifikasi Newton-Raphson. Artikel ini mendeskripsikan konstruksi algoritma Metode Modifikasi *Newton-Raphson* untuk menyelesaikan persamaan *fuzzy nonlinear* dan mengimplementasikan algoritma tersebut ke dalam program MATLAB. Sebagai aplikasi algoritma yang dikontruksi sebuah studi kasus disertai dalam artikel ini. Kasus yang diselesaikan dengan algoritma yang dikontruksi memberikan efesiensi penggunaan waktu dan akurasi solusi sehingga proses perhitungan dapat dilakukan dengan mudah dan dalam waktu yang relatif singkat.

Kata kunci: *algoritma, persamaan* fuzzy nonlinear, *modifikasi* Newton-Raphson.

Abstract

Nonlinear fuzzy equations are mathematical problems that can be found in many mathematical applications, such as robotics and techniques. The equation can be solved by a numerical method. There are a number of numerical methods that can be used to solve nonlinear fuzzy equations, for example the Newton-Raphson Modification method. This article describes the construction of the Newton-Raphson Modification Method algorithm to complete nonlinear fuzzy equations and implement that algorithm into the MATLAB program. As an algorithmic application contracted a case study is included in this article. The case solved by a controlled algorithm gives the efficiency of time use and accuracy of the solution so that the calculation process can be done easily and in relatively short time.

Keywords: algorithm, nonlinear fuzzy equation, modified Newton-Raphson.

1. Pendahuluan

Model-model persamaan *nonlinier* telah memainkan peran utama dalam bidang teknik, matematika, kedokteran, dan robotika. Salah satu penerapan dalam masalah nonlinier adalah pada cuaca yang secara inheren bersifat nonlinier. Pada umumnya, menentukan solusi eksak berdasarkan sebuah metode analitik terhadap permasalahan persamaan *nonlinear* cendrung sulit dilakukan karena ketidaksederhanaan bentuk persamaan *nonlinear* tersebut [1]. Oleh karena itu dilakukan dengan cara numerik. Salah satu tipe persamaan *nonlinear* saat ini dapat dijumpai dalam berbagai terapan matematika adalah persamaan *fuzzy* [2,3,4,5,6,7].

Terdapat sejumlah metode yang digunakan dalam menyelesaikan persamaan *nonlinear fuzzy* [2,3,4]. Pada metode-metode numerik untuk menyelesaikan persamaan *nonlinear*, terdapat sejumlah metode numerik yang khusus digunakan untuk mengatasi persoalan program *nonlinear* yang berskala besar yang memerlukan *tools* program komputer untuk melakukan proses iteratif dalam komputasinya, misalnya metode Newton [8], metode *Newton-Raphson* [5], dan metode *Broyden* [7]. Metode yang terakhir disebutkan digunakan untuk menyelesaikan sistem persamaan nonlinier dan dirancang untuk menyempurnakan metode Newton [9]. Sementara itu, metode *Newton-Raphson* yang dimodifikasi untuk persamaan *fuzzy* adalah metode numerik yang digunakan untuk menyelesaikan persamaan non linier. Artikel ini ini bertujuan untuk mendeskripsikan secara lengkap algoritma metode modifikasi *Newton-Raphson* untuk menyelesaikan persamaan *nonlinier fuzzy* dalam bentuk sebuah persamaan polinom.

Artikel ini terdiri dari 4 (empat) bagian yang dimulai dengan Pendahuluan di bagian pertama. Sementara itu pada bagian kedua dideskripsikan konsep persamaan nonlinear fuzzy dan metode modifikasi Newton. Algoritma yang dikonstruksi sebagai bagian utama artikel itu diberikan dalam bagian ketiga. Artikel ini diakhiri dengan sebuah kesimpulan yang diberikan dalam bagian ke empat..

2. Persamanaan *Nonlinear Fuzzy* dan Metode Modifkasi *Newton-Raphson*

2.1 Persamanaan *Nonlinear Fuzzy*

Pada himpunan klasik, nilai keanggotaan hanya memasangkan nilai 0 dan 1 untuk unsur-unsur pada semesta pembicaraan, yang menyatakan anggota atau bukan anggota [2]. Jika X adalah himpunan semesta, maka nilai keanggotaan untuk himpunan A adalah fungsi $\mu_A(x): X \rightarrow \{0,1\}$ dengan

$$\mu_A(x) = \begin{cases} 1, & jika\ x \in A \\ 0, & jika\ x \notin A \end{cases}$$

Fungsi ini pada himpunan *fuzzy* diperluas sehingga nilai yang dipasangkan pada unsur-unsur dalam semesta pembicaraan tidak hanya 0 dan 1 saja, tetapi keseluruhan nilai dalam interval $[0,1]$ yang menyatakan derajat keanggotaan suatu unsur pada himpunan yang dibicarakan

Bilangan *fuzzy* didefinisikan sebagai himpunan *fuzzy* dalam semesta himpunan semua bilangan riil R yang memenuhi empat sifat yaitu normal, mempunyai pendukung (*support*) yang terbatas, semua potongan α-nya adalah selang tertutup dalam $\mathbb{R}$, dan konvek [6]. Terdapat beberapa bentuk bilangan *fuzzy*, namun yang paling popular adalah bilangan *fuzzy* segitiga yang dipresentasikan sebagai berikut [3]:

$$A = (a_1, a_2, a_3)$$

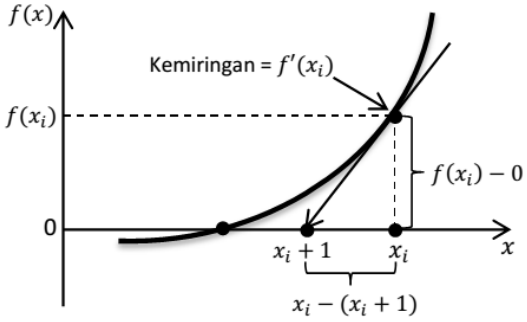
dengan fungsi keanggotaan sebagai berikut:

$$\mu_{(A)}(x) = \{\frac{x-a_1}{a_2-a_1}, \quad a_1 \le x \le a_2 \ \frac{a_3-x}{a_3-a_2}, \quad a_2 \le x \le a_3\ 0, \quad x < a_1\ atau\ x > a_3$$

2.2 Metode *Modifikasi Newton-Raphson*

Dalam metode numerik terdapat beberapa bentuk proses hitungan atau algoritma untuk menyelesaikan suatu tipe persamaan matematis [1]. Algoritma merupakan metode umum yang digunakan untuk menyelesaikan kasus-kasus tertentu yang dapat ditulis menggunakan notasi matematika dan masih belum dapat dijalankan pada komputer [10]. Dalam kehidupan sehari-hari, sudah dilakukan penyusunan algoritma untuk menyelesaikan permasalahan atau tantangan yang dihadapi. Dalam membuat algoritma diperlukan suatu mekanisme atau alat bantu untuk menuangkan hasil pemikiran mengenai langkah-langkah penyelesaian masalah yang sistematis dan terurut. Pada dasarnya untuk bisa menyusun solusi diperlukan kemampuan *problem-solving* yang baik. Oleh karena itu, sebagai sarana untuk melatih kemampuan tersebut terdapat sebuah *tool* (alat) yang dapat digunakan, yakni *flowchart*. Secara formal, *flowchart* didefinisikan sebagai skema penggambaran dari algoritma atau proses.

Metode *Newton-Raphson* merupakan metode yang paling banyak digunakan dari semua formula penempatan akar [9].



Gambar 1. Grafik Metode *Newton-Raphson*

Metode *Newton-Raphson* dapat diturunkan berdasarkan interpretasi geometrik. Seperti pada Gambar 1, turunan pertama pada x_i adalah ekuivalen terhadap kemiringan *(slope)*:

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

yang dapat diatur kembali menjadi:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{2}$$

Metode Newton mensyaratkan bahwa turunan dihitung langsung. Hal ini jelas berdasarkan rumus metode Newton bahwa itu akan gagal dalam kasus di mana turunannya bernilai nol. Untuk menghindari perhitungan $f'(x)$ mungkin tidak selalu ada atau mungkin sulit untuk menghitung dan mempertahankan sifat konvergensi dari metode *Newton-Raphson*, maka $f'(x_n)$ diganti dengan $f[x_n, x_{n-1}]$ pada persamaan (2). Sehingga persamaan (2) menjadi:

$$x_{n+1} = x_n - \frac{f(x_n)}{f[x_n, x_{n-1}]}$$

$$= x_n - \frac{f(x_n)}{\frac{f(x_n)-f(x_{n-1})}{(x_n - x_{n-1})}} \tag{3}$$

di mana $f[x_n, x_{n-1}] = \frac{f(x_n)-f(x_{n-1})}{(x_n - x_{n-1})}$ adalah diferensiasi terbagi. Dimulai dengan dua perkiraan awal x_0 dan x_1, lalu hitung x_2 dengan metode Secant. Kemudian terapkan metode Modifikasi *Newton-Raphson* untuk menghitung x_3, x_4, x_5, ... menggunakan tabel interpolasi. Ganti $f'(x_n)$ pada persamaan (3) dengan $g_{n,k}(x)$ yang ditulis dalam bentuk Newtonian sebagai berikut:

$$g_{n,k}(x) = f(x_n) + \sum_{i=1}^{k} f[x_n, x_{n-1}, \dots, x_{n-i}] \prod_{j=0}^{i-1} (x_n - x_{n-j}) \tag{4}$$

di mana $f[x_n, x_{n-1}, \dots, x_{n-i}]$ adalah diferensiasi terbagi dari $f(x)$. Diferensiasi terbagi didefinisikan sebagai $f[x_n] = f(x_n)$ dan $f[a, b] = \frac{f(a)-f(b)}{a-b}, a \neq b$. Dari persamaan (4), $g_{n,k}(x)$ dihitung dengan menata x_i sebagai $x_n, x_{n-1}, \dots, x_{n-i}$ untuk $i = 1,2, \dots, k$. Kita dapat menghitung $g_{n,k}(x)$ dan diberikan $x = x_n$, menjadi:

$$g_{n,k}(x) = f(x_n) + \sum_{i=2}^{k} f[x_n, x_{n-1}, \dots, x_{n-i}] \prod_{j=1}^{i-1} (x_n - x_{n-j}) \tag{5}$$

Di mana diferensiasi pertama:

$$\nabla f_1 = \frac{f_1 - f_0}{x_1 - x_0}, \nabla f_2 = \frac{f_2 - f_1}{x_2 - x_1}, \dots, \nabla f_n = \frac{f_n - f_{n-1}}{x_n - x_{n-1}}$$

Diferensiasi kedua:

$$\nabla^2 f_2 = \frac{\nabla f_2 - \nabla f_1}{x_2 - x_0}, \nabla^2 f_3 = \frac{\nabla f_3 - \nabla f_2}{x_3 - x_1}, \dots, \nabla^2 f_n = \frac{\nabla f_n - \nabla f_{n-1}}{x_n - x_{n-2}}$$

Diferensiasi ketiga:

$$\nabla^3 f_3 = \frac{\nabla^2 f_3 - \nabla^2 f_2}{x_3 - x_0}, \nabla^3 f_4 = \frac{\nabla^2 f_4 - \nabla^2 f_3}{x_4 - x_1}, \dots, \nabla^3 f_n = \frac{\nabla^2 f_n - \nabla^2 f_{n-1}}{x_n - x_{n-3}}$$

Dan seterusnya hingga didapat diferensiasi ke i:

$$\nabla^i f_n = \frac{\nabla^{i-1} f_n - \nabla^{i-1} f_{n-1}}{x_n - x_{n-i}}$$

Untuk $k = 2$, persamaan (5) menjadi

$$g_{n,2}(x) = f[x_n, x_{n-1}] + f[x_n, x_{n-1}, x_{n-2}](x_n - x_{n-1}) \tag{6}$$

dengan menulis $f[x_n, x_{n-1}] = \nabla f_n$ dan $f[x_n, x_{n-1}, x_{n-2}] = \nabla^2 f_n$ sehingga persamaan (5) menjadi $g_{n,2}(x) = \nabla f_n + \nabla^2 f_n (x_n - x_{n-1})$. Formula Modifikasi *Newton-Raphson* untuk $k = 2$ adalah sebagai berikut:

$$x_{n+1} = x_n - \frac{f(x_n)}{\nabla f_n + \nabla^2 f_n (x_n - x_{n-1})} \tag{7}$$

Dengan cara yang sama, dapat ditentukan formula Modifikasi *Newton-Raphson* untuk $k = 3$ adalah sebagai berikut [5]:

$$x_{n+1} = x_n - \frac{f(x_n)}{\nabla f_n + \nabla^2 f_n(x_n - x_{n-1}) + \nabla^3 f_n(x_n - x_{n-1})(x_n - x_{n-2})} \quad (8)$$

3. Hasil dan Pembahasan

Algoritma yang dikonstruksi dalam bagian ini ditulis dalam format *Pseudo Code* dan diagram alir (*flow chart*) yang digunakan untuk menyelesaikan persamaan *fuzzy non linear* menggunakan metode Modifikasi *Newton-Raphson*. Bagian-bagian yang diperlukan dalam algoritma tersebut adalah diferensiasi terbagi. Selain itu algoritma yang dibangun merlu meperhatikan proses komputasi pada masing-masing variabel secara terpisah, yaitu x_1, x_2, dan x_3. Pada bagian input nilai awal, terdapat hal yang perlu diperhatikan yaitu pilihan nilai awal yang tepat yang tidak mengakibatkan terjadinya nilai menjadi *blow up* menuju tak berhingga. Nilai awal yang dibutuhkan yaitu dua bilangan *fuzzy* segitiga, misalkan (x_1^1, x_2^1, x_3^1) dan (x_1^2, x_2^2, x_3^2). Untuk menghindari terjadinya *blow up*, pilihan nilai awal memenuhi interval $x_1^1 < x_2^1 < x_3^1$ dan $x_1^2 < x_2^2 < x_3^2$, di mana $x_1^1 < x_1^2$, $x_2^1 < x_2^2$, dan $x_3^1 < x_3^2$. Secara lengkap algoritma yang dimaksud dinyatakan dalam bagian berikut ini.

Nama algoritma: Algoritma Modifikasi Newton-Raphson untuk menyelesaikan persamaan nonlinear fuzzy dengan persamaan berbentuk polinom.
Kondisi awal: Diberikan suatu persamaan fuzzy *non linear*. Lalu ditentukan dua nilai awal pada x_1, x_2, dan x_3 serta nilai toleransi dan batas iterasi maksimal.
Kondisi akhir: Nilai-nilai peubah x_1, x_2, dan x_3 sebagai solusi penyelesaian persamaan fuzzy *non linear* menggunakan metode modifikasi *Newton-Raphson* ditampilkan pada *screen*/monitor.
Deklrasi : z,k,n,i,j : integer
x1,x2,x3, f1,f2,f3 : array[1..z] of real
g1,g2,g3 : array[1..z,1..z] of real;
eps,s1,s2,s3,p1,p2,p3,err1,err2,err3 : real
Deskripsi:

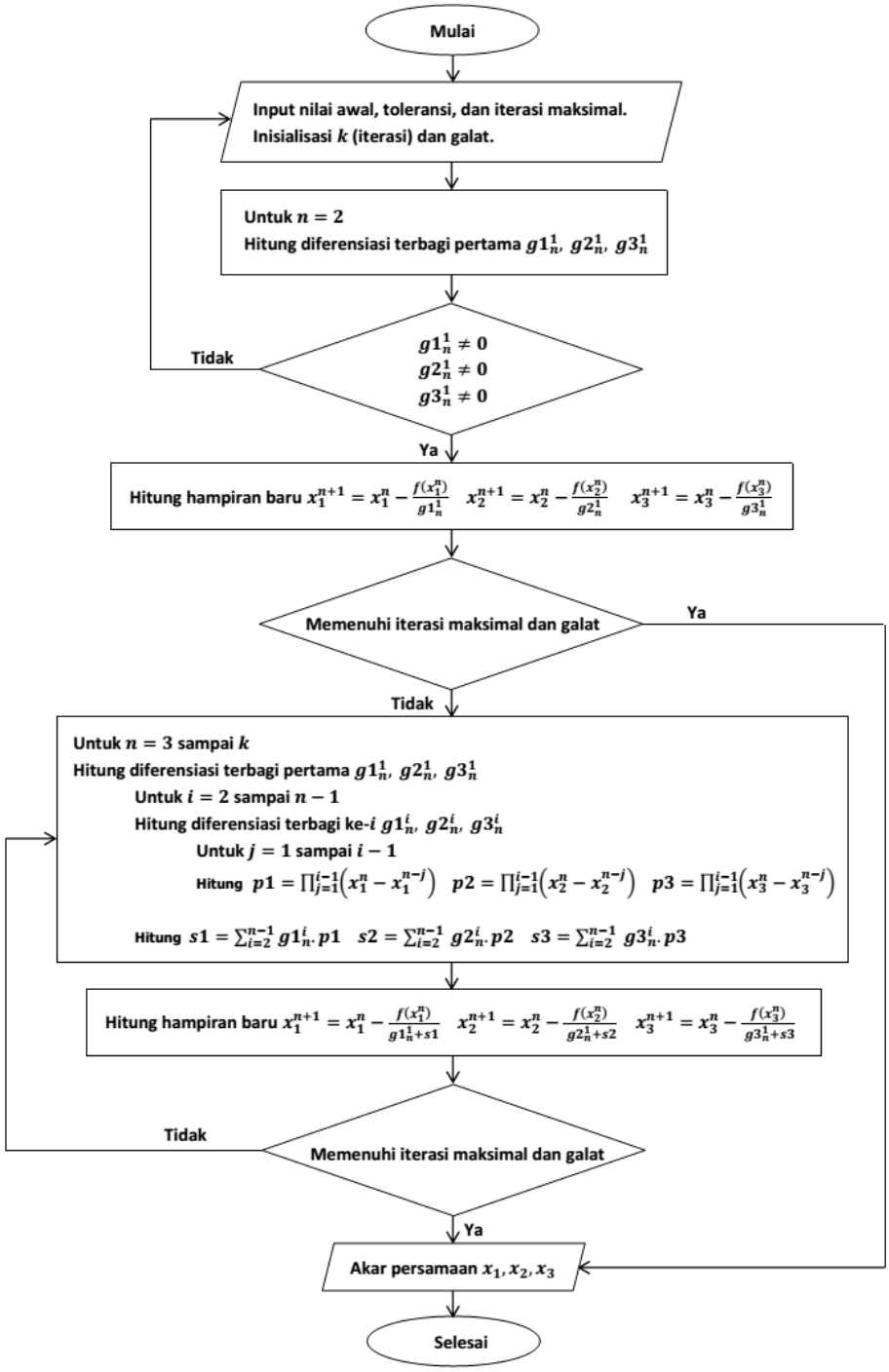
```
function algoritma_modifikasi_NewtonRaphson (x1,x2,x3, f1,f2,f3 : array[1..z] of real; g1,g2,g3 : array[1..z,1..z] of real; eps,s1,s2,s3,p1,p2,p3,err1,err2,err3 : real; z,k,n,i,j : integer)
Read (x1(1),x2(1),x3(1),x1(2),x2(2),x3(2),eps,z,k,err1,err2,err3)
while (k<=z&&err1>=eps&&err2>=eps&&err3>=eps)
  for n = 2 to 2 do
  begin
  g1(1,n) ← (f1(n)-f1(n-1))/(x1(n)-x1(n-1));          g2(1,n) ← (f2(n)-f2(n-1))/(x2(n)-x2(n-1));
  g3(1,n) ← (f3(n)-f3(n-1))/(x3(n)-x3(n-1));
  x1(n+1) ← x1(n)-(f1(n)/g1(1,n));                    x2(n+1) ← x2(n)-(f2(n)/g2(1,n));
  x3(n+1) ← x3(n)-(f3(n)/g3(1,n));
  k ← k+1;
  for n = 3 to k do
  begin
  g1(1,n) ← (f1(n)-f1(n-1))/(x1(n)-x1(n-1));          g2(1,n) ← (f2(n)-f2(n-1))/(x2(n)-x2(n-1));
  g3(1,n) ← (f3(n)-f3(n-1))/(x3(n)-x3(n-1));
  s1 ← 0; s2 ← 0; s3 ← 0;
  for i = 2 to (n-1) do
  begin
    g1(i,n) ← (g1(i-1,n)-g1(i-1,n-1))/(x1(n)-x1(n-i));  g2(i,n) ← (g2(i-1,n)-g2(i-1,n-1))/(x2(n)-x2(n-i));
    g3(i,n) ← (g3(i-1,n)-g3(i-1,n-1))/(x3(n)-x3(n-i));
    p1 ← 1; p2 ← 1; p3 ← 1;
    for j = 1 to (i-1) do
    begin
      p1 ← p1*(x1(n)-x1(n-j));    p2 ← p2*(x2(n)-x2(n-j));  p3 ← p3*(x3(n)-x3(n-j));
    endfor
    s1 ← s1+(g1(i,n).*p1);        s2 ← s2+(g2(i,n).*p2);        s3 ← s3+(g3(i,n).*p3);
  endfor
  x1(n+1) ← x1(n)-(f1(n)/(g1(1,n)+s1));     x2(n+1) ← x2(n)-(f2(n)/(g2(1,n)+s2));
  x3(n+1) ← x3(n)-(f3(n)/(g3(1,n)+s3));     err1 ← abs(x1(n+1)-x1(n));
  err2 ← abs(x2(n+1)-x2(n));                err3 ← abs(x3(n+1)-x3(n));
  endfor
```

```
   endfor
   endwhile
function a ← f1(n) (n : integer; a,x1 : array[1..z] of real)
a ← {fungsi f1(n)}
endfunction
function b ← f2(n) (n : integer; b,x2 : array[1..z] of real)
b ← {fungsi f2(n)}
endfunction
function c ← f3(n) (n : integer; c,x3 : array[1..z] of real)
c ← {fungsi f3(n)}
endfunction
tabel(:,1) ← x1';
tabel(:,2) ← x2';
tabel(:,3) ← x3';
write(tabel)
endfunction
```

{kondisi selesai pengulangan : k > z && err1 < eps && err2 < eps && err3 < eps, maka solusi penyelesaiannya berupa akar persamaan x1, x2, dan x3 yang ditampilkan dalam bentuk tabel}
Secara deskriptif, algoritma tersebut dapat dijabarkan sebagai berikut.

1. Input:
   Nilai awal, toleransi, iterasi maksimal. Inisialisasi k (iterasi) dan galat.
2. Perhitungan inti:
   Ketika galat ≥ toleransi dan $k \leq$ iterasi maksimal
   Iterasi pertama (untuk $n = 2$)
   Hitung diferensiasi terbagi pertama
   $g1_n^1 = \frac{f(x_1^n)-f(x_1^{n-1})}{x_1^n-x_1^{n-1}}$; $g2_n^1 = \frac{f(x_2^n)-f(x_2^{n-1})}{x_2^n-x_2^{n-1}}$; $g3_n^1 = \frac{f(x_3^n)-f(x_3^{n-1})}{x_3^n-x_3^{n-1}}$
   Hitung nilai hampiran baru
   $x_1^{n+1} = x_1^n - \frac{f(x_1^n)}{g1_n^1}$; $x_2^{n+1} = x_2^n - \frac{f(x_2^n)}{g2_n^1}$; $x_3^{n+1} = x_3^n - \frac{f(x_3^n)}{g3_n^1}$
   Perbarui iterasi
   Iterasi kedua dan seterusnya (untuk $n = 3$ sampai k)
   Hitung diferensiasi terbagi pertama
   $g1_n^1 = \frac{f(x_1^n)-f(x_1^{n-1})}{x_1^n-x_1^{n-1}}$ $g2_n^1 = \frac{f(x_2^n)-f(x_2^{n-1})}{x_2^n-x_2^{n-1}}$ $g3_n^1 = \frac{f(x_3^n)-f(x_3^{n-1})}{x_3^n-x_3^{n-1}}$
   Untuk $i = 2$ sampai $n - 1$
   Hitung diferensiasi terbagi ke-i
   $g1_n^i = \frac{g1_n^{i-1}-g1_{n-1}^{i-1}}{x_1^n-x_1^{n-1}}$; $g2_n^i = \frac{g2_n^{i-1}-g2_{n-1}^{i-1}}{x_2^n-x_2^{n-1}}$ $g3_n^i = \frac{g3_n^{i-1}-g3_{n-1}^{i-1}}{x_3^n-x_3^{n-1}}$
   Untuk $j = 1$ sampai $i - 1$
   Hitung
   $p1 = \prod_{j=1}^{i-1}(x_1^n - x_1^{n-j}); p2 = \prod_{j=1}^{i-1}(x_2^n - x_2^{n-j}); p3 = \prod_{j=1}^{i-1}(x_3^n - x_3^{n-j})$
   Selesai untuk j
   Hitung
   $s1 = \sum_{i=2}^{n-1} g1_n^i.p1; s2 = \sum_{i=2}^{n-1} g2_n^i.p2;\ s3 = \sum_{i=2}^{n-1} g3_n^i.p3$
   Selesai untuk i
   Hitung nilai hampiran baru
   $x_1^{n+1} = x_1^n - \frac{f(x_1^n)}{g1_n^1+s1}$; $x_2^{n+1} = x_2^n - \frac{f(x_2^n)}{g2_n^1+s2}$; $x_3^{n+1} = x_3^n - \frac{f(x_3^n)}{g3_n^1+s3}$
   Hitung galat
   Selesai
3. Input fungsi $f(x_1^n), f(x_2^n), f(x_3^n)$
   Output: Nilai akar persamaan x_1, x_2, x_3.

Algoritma yang dikemukan di atas, dalam diagram alir (*flowchart*) dapat dinyatakan sebagai berikut:



Gambar 2. Diagram Alir Metode Modifikasi *Newton-Raphson* untuk menyelesaikan persamaan *nonlinear fuzzy*

Contoh pemakaian.

Pandang persamaan *fuzzy nonlinear* yang terdiri dari bilangan *fuzzy* segitiga berikut ini [4].

$$(3,4,5)x^2 + (1,2,3)x = (1,2,3) \quad (10)$$

Definisikan x merupakan bilangan *fuzzy* segitiga dengan anggota (x_1, x_2, x_3) sehingga persamaan tersebut dapat ditulis menjadi

$$(3,4,5)(x_1, x_2, x_3)^2 + (1,2,3)(x_1, x_2, x_3) + (-1,-2,-3) = 0 \quad (11)$$

dan akan diselesaikan dengan proses perhitungan pada masing-masing variabel. Selanjutnya tentukan dua nilai awal $(x_1^1, x_2^1, x_3^1) = (0.1, 0.2, 0.3)$ dan $(x_1^2, x_2^2, x_3^2) = (0.2, 0.3, 0.4)$, toleransi sebesar 10^{-4}, serta iterasi maksimal 50.

Dengan menggunakan algoritma yang telah didesain dan diimplementasikan menggunakan program Matlab, dapat dihasilkan *output* (keluaran) berupa tabel nilai hampiran dari masing-masing iterasi sebagai berikut.

Tabel . Hasil Nilai Hampiran

x1	x2	x3
0.100000000000000	0.200000000000000	0.300000000000000
0.200000000000000	0.300000000000000	0.400000000000000
0.557894736842105	0.560000000000000	0.553846153846154
0.444806932585702	0.502222222222222	0.530977130977131
0.434349529774549	0.500003282455277	0.530662445898873
0.434258552797387	0.500000000007183	0.530662386291810

Sehingga perhitungan berhenti pada iterasi keempat, dengan solusi penyelesaian yaitu $x_1 = 0.43425$, $x_2 = 0.5$, dan $x_3 = 0.53066$. Jika ditulis dalam bentuk bilangan *fuzzy* segitiga, maka solusi penyelesaiannya yaitu $x = (0.43425, 0.5, 0.53066)$.

4. Kesimpulan

Kami telah mendeskripsikan langkah-langkah konstruksi algoritma penyelesaian persamaan *fuzzy nonlinear* menggunakan metode modifikasi Newton-Raphson pada bagian hasil dan pembahasan. Algoritma tersebut telah diimplementasikan ke dalam MATLAB dan digunakan untuk menyelesaikan sebuah kasus persamaan *fuzzy nonlinear*. Terhadap kasus tersebut diperoleh bahwa hasil penggunaan algoritma kami memberikan solusi yang akurat hingga tingkat ketelitian perseratus ribu dan waktu komputasi yang relatif cepat.

Ucapan Terima Kasih

Ucapan terimakasih disampaikan kepada reviewer yang telah memberikan saran dan masukan berharga untuk memperbaiki kualitas artikel ini.

Daftar Pustaka:

[1] Triatmodjo, B. 2002. Metode Numerik Dilengkapi dengan Program Komputer. Beta Offset, Yogyakarta.
[2] Klir, G.J. & Yuan, B. 1995. *Fuzzy Set and Fuzzy Logic: Theory and Applications*. Prentice Hall International, New Jersey.
[3] Kwang, H.L. 2005. *First Course on Fuzzy Theory and Applications*. Springer, Berlin.
[4] Shokri, J. 2008. Numerical Method for Solving Fuzzy Nonlinear Equations. *Applied Mathematical Sciences.* **2**(24): 1191-1203
[5] Subash, R. & Sathya, S. 2011. Numerical Solution Of Fuzzy Modified Newton-Raphson Method For Solving Nonlinear Equations. *International Journal of Current Research*. **3**(11): 390-392..
[6] Utomo, T. 2012. *Operasi Aritmetika Dasar pada Bilangan Fuzzy dan Sifat Sifatnya*. (Skripsi). Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, Malang
[7] Ramli A., Abdullah M.L, and Mamat M. 2010. Broyden's Method for Solving Fuzzy Nonlinear Equations, *Advances in Fuzzy Systems* **2010**. 6 pages doi:10.1155/2010/763270
[8] S. Abbasbandy, B. Asady, Newton_s method for solving fuzzy nonlinear equations, Appl. Math. Comput. 159 (2004) 349–356.
[9] Chapra, S.C. & Canale, R.P. 1991. *Metode Numerik Untuk Teknik dengan Penerapan pada Komputer Pribadi*. Universitas Indonesia (UI-Press), Jakarta.
[10] Harumy, T.H.F., dkk. 2016. *Belajar Dasar Algoritma & Pemrograman C++*. Pemula, Medan.