UJI KOMPUTASI ALGORITME VARIAN METODE NEWTON PADA PERMASALAHAN OPTIMASI NONLINEAR TANPA KENDALA

N. HAQUEQY[1], B. P. SILALAHI[2], I. S. SITANGGANG[3]

Abstrak

Penelitian ini membahas kombinasi metode untuk menyelesaikan permasalahan optimasi nonlinear tanpa kendala dengan membuat algoritme baru dari kombinasi metode Newton. Algoritme merupakan sebuah prosedur yang digunakan untuk menyelesaikan masalah tertentu dengan cara mengubah *input* ke dalam *output* yang diinginkan. Metode yang akan digunakan adalah metode Newton, Aturan Trapesium dan metode Halley (NTH). Metode Newton merupakan salah satu metode terbuka untuk menentukan solusi akar dari persamaan nonlinear. Persamaan nonlinear adalah persamaan yang variabelnya berpangkat lebih dari satu. Untuk meningkatkan penyelesaian masalah dalam optimasi, maka metode Newton akan dikombinasikan dengan Aturan Trapesium dan metode Halley. Penelitian ini bertujuan untuk membuat algoritme baru dari hasil kombinasi metode dan membandingkan hasil uji komputasi antara algoritme metode kombinasi dengan algoritme metode Newton menggunakan beberapa fungsi nonlinear. Perbandingan uji komputasi memperlihatkan bahwa metode NTH menghasilkan jumlah iterasi yang lebih sedikit daripada metode Newton, berbanding terbalik dengan hasil yang diperoleh untuk *running time*, metode NTH membutuhkan waktu yang lama dibandingkan dengan metode Newton dalam melakukan pencarian akar.

Kata kunci: algoritme, metode newton, metode halley, nonlinear, optimasi.

1 PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan suatu cara yang dapat dilakukan ketika bekerja dalam bidang ilmu eksperimental dan keteknikan, yang meliputi fungsi matematika dan proses industri untuk mendapatkan metode analisis baru (Cerdà *et al.* [1]). Prinsip utama dalam pemodelan optimasi adalah menentukan solusi terbaik yang optimal

[1] Mahasiswa S2, Program Studi Ilmu Komputer, Sekolah Pascasarjana IPB. E-mail: queqy.nh@gmail.com
[2] Departemen Matematika, Fakultas Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor,16680
[3] Departemen Ilmu Komputer, Fakultas Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor,16680

(minimum/maksimum) dari suatu tujuan yang dimodelkan melalui fungsi objektif. Secara garis besar, masalah dalam optimasi dikategorikan menjadi dua bagian, yaitu masalah optimasi dengan kendala dan masalah optimasi tanpa kendala. Masalah optimasi tanpa kendala merupakan masalah optimasi yang tidak memiliki batasan, sedangkan masalah optimasi dengan kendala merupakan masalah pengoptimuman fungsi objektif dengan kendala-kendalanya. Penyelesaian dalam fungsi optimasi dibagi menjadi dua yaitu penyelesaian fungsi optimasi dengan analitik dan penyelesaian fungsi optimasi dengan pendekatan numerik. Metode numerik adalah teknik yang digunakan untuk memformulasikan persoalan matematika sehingga dapat dipecahkan dengan operasi penghitungan atau aritmatika biasa. Penyelesaian menggunakan metode numerik biasanya melibatkan proses iterasi. Apabila proses iterasi dilakukan secara manual, maka akan membutuhkan waktu yang lama dan memungkinkan terjadinya *human error*.

Pemecahan masalah dalam suatu pemrograman merupakan sebuah tujuan dalam proses pembuatan sebuah program. Dalam pemecahan masalah tersebut, terdapat dua macam jenis pemrograman, yaitu pemrograman linear dan pemrograman non-linear. Pemrograman nonlinear adalah suatu bentuk pemrograman yang berhubungan dengan suatu perencanaan aktivitas tertentu yang dapat diformulasikan dalam model matematika, yang memuat fungsi tujuan dan fungsi kendala berbentuk nonlinear yaitu pangkat dan variabelnya lebih dari satu. Hal yang membedakan pemrograman non-linear dengan linear adalah fungsi tujuan dan fungsi kendala yang diasumsikan linear untuk masalah-masalah tertentu. Masalah-masalah pemrograman non-linear dapat dikembangkan dalam berbagai macam model, di antaranya masalah pemrograman non-linear umum yang memuat fungsi tujuan dan fungsi kendala dan masalah pemrograman nonlinear tanpa kendala yang tentunya hanya memuat fungsi tujuan yang akan dioptimalkan. Untuk menyelesaikan permasalahan persamaan nonlinear terdapat banyak metode dan algoritme yang dapat digunakan, tetapi setiap metode dan algoritme mempunyai kelebihan dan kekurangan masing-masing. Salah satunya metode numerik yang digunakan untuk menyelesaikan persoalan apabila penghitungan secara analitik tidak dapat digunakan [14]. Metode numerik merupakan teknik yang digunakan untuk memformulasikan persoalan matematika sehingga dapat dipecahkan dengan operasi penghitungan dan dapat dibuat kedalam bentuk algoritme yang dapat dihitung secara cepat dan mudah. Ada banyak metode numerik yang dapat digunakan untuk menyelesaikan sistem persamaan linear maupun nonlinear, di antaranya metode Newton dan metode Halley. Metode Newton merupakan salah satu metode terbaik untuk menentukan solusi akar dari persamaan nonlinear [11]. Metode Newton sering konvergen dengan cepat menurut Kumar *et al.* [6], terutama bila iterasi dimulai cukup dekat dengan akar yang diinginkan. Metode Halley sendiri merupakan metode dengan orde kekonvergenan tiga [8] di mana metode ini berarti memiliki orde kekonvergenan yang lebih tinggi dibandingkan dengan metode Newton.

Para peneliti umumnya berusaha untuk menemukan metode dengan algoritme yang paling efektif dan efisien untuk dapat menyelesaikan masalah

optimasi linear maupun nonlinear. Seperti yang telah dilakukan oleh Weerakoon *et al.* [15] yang mengembangkan metode Newton dengan cara memodifikasi metode Newton dan aturan trapesium. Modifikasi metode Newton dengan aturan trapesium menghasilkan konvergen tiga, sedangkan metode Newton sendiri masih menghasilkan konvergen dua. Dengan demikian modifikasi metode ini menghasilkan iterasi yang lebih sedikit apabila dibandingkan dengan metode Newton. Homeier [4] melakukan penelitian mengenai metode Newton dengan menggunakan fungsi invers. Ini juga merupakan modifikasi dari metode Newton. Penelitian ini menghasilkan metode dengan konvergen tiga. Jain [5] melakukan modifikasi pada metode Newton yang sebelumnya telah dilakukan oleh Homeir dan Weerakoon dengan menambahkan metode Secant ke dalam metode Newton dengan aturan Trapesium dan metode Newton dengan menggunakan fungsi *invers*. Penelitian ini menghasilkan metode Secant Trapesium Newton dan metode Secant Invers Newton dengan konvergen empat. Selanjutnya Noor [9] melakukan modifikasi metode Halley dan menghasilkan pengembangan baru dari metode Halley sehingga menghasilkan metode yang lebih baik.

Oleh karena itu, dengan kelebihan dan kekurangan yang dimiliki oleh masing-masing metode Newton dan metode Halley dalam menyelesaikan permasalahan persamaan linear maupun nonlinear, maka pada penelitian ini diusulkan metode kombinasi antara varian metode Newton dengan aturan trapesium yang penelitian sebelumnya telah dilakukan oleh Weerakoon *et al.* [15] dan metode Halley.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah, penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana mengkombinasikan metode Newton dengan aturan trapesium dan metode Halley?
2. Bagaimana perbandingan hasil uji komputasi antara algoritme kombinasi metode Newton, Aturan Trapesium dan metode Halley dengan algoritme metode Newton menggunakan beberapa fungsi nonlinear?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini ialah:

1. Mengkombinasikan metode Newton dengan Aturan Trapesium dan metode Halley,
2. Membandingkan hasil uji komputasi antara algoritme kombinasi metode Newton, Aturan Trapesium dan metode Halley dengan algoritme metode Newton menggunakan beberapa fungsi nonlinear.

2 METODE

2.1 Studi Literatur

Pada tahapan ini akan dilakukan studi literatur mengenai penelitian yang telah dilakukan oleh peneliti-peneliti sebelumnya mengenai metode yang akan digunakan yaitu metode Newton dan metode Halley beserta kompleksitas algoritme masing-masing metode tersebut.

2.2 Kombinasi Metode Newton, Aturan Trapesium dan Metode Halley

Tahap ini akan dilakukan kombinasi varian metode Newton dan Aturan Trapesium yang sebelumnya telah dilakukan oleh Weerakoon [15] :

$x_{n+1} = x_n - \frac{2f(x_n)}{[f'(x_n)+f'(x^*_{n+1})]}$, n = 0, 1, 2,..., di mana $x^*_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. (1)

dengan metode optimasi lainnya yaitu metode Halley :

$$x_{n+1} = x_* \frac{2f(x_*)f'(x_*)}{2(f'(x_*))^2 - f(x_*)f''(x_*)} \quad (2)$$

Kombinasi dilakukan dengan cara menyubstitusikan hasil yang diperoleh dari satu metode ke metode lainnya.

2.3 Pembuatan Algoritme Metode Kombinasi

Pada tahap ini, kombinasi metode Newton, Aturan Trapesium dan metode Halley akan dibentuk kesebuah algoritme baru untuk penyelesaian masalah optimasi nonlinear tanpa kendala.

2.4 Implementasi Algoritme Metode Kombinasi

Implementasi algoritme dilakukan dengan cara menginputkan algoritme baru dan mengimplementasikannya menggunakan *software* yang dapat digunakan untuk menjalankan algoritme ini.

2.5 Pengujian Komputasi

Pada tahap ini akan dilakukan uji komputasi dengan menggunakan algoritme metode Newton dan algoritme baru yang dilakukan dari hasil kombinasi metode dengan menggunakan beberapa fungsi nonlinier seperti yang terlihat pada Tabel 1.

Tabel 1
Fungsi nonlinear

Fungsi Nonlinear	
f_1	$x^3 + 4x^2 - 10$
f_2	$\sin^2(x) - x^2 + 1$
f_3	$x^2 - e^x - 3x + 2$
f_4	$\cos(x) - x$
f_5	$(x-1)^3 - 1$
f_6	$x^3 - 10$
f_7	$x\exp(x^2) - \sin^2(x) + 3\cos(x) + 5$
f_8	$x^2\sin^2(x) + \exp[x^2\cos(x)\sin(x)] - 28$
f_9	$\exp(x^2 + 7x - 30) - 1$

Hal ini dilakukan untuk melihat kelebihan dan kekurangan yang dimiliki oleh masing-masing algoritme metode berdasarkan jumlah iterasi yang digunakan, *running time* dan pendekatan hasil yang diperoleh.

3 HASIL DAN PEMBAHASAN

3.1 Kombinasi Metode Newton, Aturan Trapesium dan Metode Halley

Pada penelitian ini dilakukan kombinasi metode dengan menggunakan beberapa metode iterasi, yaitu metode Newton, Aturan Trapesium dan metode Halley yang bertujuan untuk mencari akar dari suatu fungsi nonlinear. Dalam kombinasi metode ini, setiap iterasinya dilakukan tiga kali evaluasi fungsi dengan cara menggabungkan ketiga metode tersebut.

Langkah pertama untuk setiap satu kali iterasi dimulai dengan menggunakan metode Newton seperti persamaan (Luenberger dan Ye [7]) :

$$x^*{}_n = x_n - \frac{f(x_n)}{f'(x_n)} \tag{3}$$

dengan $x_n = x_0$ adalah titik tebakan awal dan $f'(x_n)$ merupakan turunan pertama fungsi dengan mensubstitusikan titik tebakan awal. Pemilihan metode Newton sebagai fungsi pertama pada kombinasi ini dikarenakan metode Newton tidak memerlukan *cost* yang besar dalam pencarian akar.

Tahap selanjutnya, akar yang diperoleh dari langkah pertama akan disubstitusikan ke dalam formula Aturan Trapesium seperti berikut (Weerakoon [15]):

$$x_* = x_n - \frac{2f(x_n)}{[f'(x_n) + f'(x^*n)]} \tag{4}$$

di mana $x^*{}_n$ pada $f'(x^*{}_n)$ merupakan hasil dari metode Newton yang telah dilakukan sebelumnya. Metode yang dikembangkan oleh Weerakoon ini menghasilkan jumlah iterasi yang lebih sedikit dibandingkan dengan menggunakan metode Newton tanpa kombinasi dan menghasilkan orde konvergen tiga.

Dan pada tahap terakhir untuk setiap satu kali iterasi akan ditambahkan dengan menggunakan formula metode Halley. Metode Halley merupakan salah satu metode iterasi dengan orde kekonvergenan tiga. Formula metode Halley akan disubtitusikan pada persamaan (4), sehingga diperoleh :

$$x_{n+1} = x_* - \frac{2f(x_*)f'(x_*)}{2(f'(x_*))^2 - f(x_*)f''(x_*)} \quad (5)$$

dengan x_* merupakan hasil yang diperoleh dari kombinasi metode Newton dan Aturan Trapesium.

3.2 Pembuatan Algoritme Kombinasi Metode Newton, Aturan Trapesium dan Metode Halley

Dalam pembuatan algoritme metode Newton, aturan trapesium dan metode Halley, diperlukan beberapa tahapan, yaitu :

Langkah 1. Definisikan fungsi $f(x)$

Langkah 2. Inputkan titik tebakan awal $x_0 \epsilon\, \mathbb{R}^n$

Langkah 3. Inputkan batas toleransi $0 < \varepsilon < 1$

Langkah 4. Inputkan batas maksimum iterasi

Langkah 5. Hitung $x^*{}_n = x_n - \frac{f(x_n)}{f'(x_n)}$

Hitung $x_* = x_n - \frac{2f(x_n)}{[f'(x_n)+f'(x^*{}_n)]}$

Hitung $x_{n+1} = x_* \frac{2f(x_*)f\prime(x_*)}{2(f\prime(x_*))^2 - f(x_*)f\prime\prime(x_*)}$

Langkah 6. Hitung $b = n(i) - n(i-1)$ atau jika jumlah iterasi mencapai batas maksimum maka berhenti, jika tidak kembali ke langkah 5.

Algoritme metode Newton Trapesium Halley (NTH) membutuhkan titik tebakan awal yaitu x_0 untuk menyelesaikan suatu fungsi persamaan. Semakin dekat titik tebakan awal yang digunakan, maka akan semakin sedikit jumlah iterasi dan semakin kecil *running time* yang diperlukan. Selanjutnya menginputkan batas toleransi untuk menentukan tingkat ketelitian dari hasil yang diperoleh. Semakin kecil nilai toleransi yang digunakan, maka hasil yang diperoleh akan semakin mendekati nilai eksak. Pada dasarnya untuk semua metode numerik kriteria dalam pemberhentian pencarian akar adalah sama. Salah satu kriteria untuk pencarian akar dalam program komputasi adalah selisih dua buah nilai yang disimbolkan dengan $|x_{n+1} - x_n|$.

3.3 Uji Komputasi

Pengujian komputasi dilakukan dengan menggunakan beberapa fungsi nonlinear standar yang telah digunakan oleh peneliti sebelumnya dengan meng-*input*-kan dua buah nilai toleransi yang berbeda, yaitu 10^{-4} dan 10^{-14}. Hal ini dilakukan untuk membandingkan seberapa jauh perbedaan antara jumlah iterasi dan *running time* yang dihasilkan. Metode yang digunakan pada pengujian komputasi menggunakan tiga metode sebagai metode pembanding, yaitu metode Newton (N), metode kombinasi Newton Trapesium Halley (NTH) dan metode Newton Trapesium (NT).

Untuk setiap fungsi dilakukan pengujian komputasi dengan menggunakan titik tebakan awal yang berbeda. Fungsi yang digunakan sebelumnya telah dijelaskan pada bab sebelumnya. Pada persamaan f_1 menggunakan titik tebakan awal dengan $x_0 = -0.5, 1, 2, -0.3$, persamaan f_2 dengan $x_0 = 1, 3$, persamaan f_3 dengan $x_0 = 2, 3$, persamaan f_4 dengan $x_0 = 1, 1.7, -0.3$, persamaan f_5 dengan $x_0 = 3.5, 2.5$, persamaan f_6 dengan $x_0 = 1.5$, persamaan f_7 dengan $x_0 = -2$, persamaan f_8 dengan $x_0 = 5$ dan f_9 dengan $x_0 = 3.5, 3.25$. *Running time* dan jumlah iterasi yang dihasilkan dapat dilihat pada tabel.

Tabel 2
Perbandingan *running time* metode dengan toleransi 10^{-4}

$f(x)$	x_0	***running time* (ms)**		
		N	**NT**	**NTH**
f_1	-0.5	0.1204	0.0212	0.0141
	1	0.0049	0.0044	0.0076
	2	0.0049	0.0062	0.0081
	-0.3	0.0438	0.0147	0.0114
f_2	1	0.0063	0.0072	0.0146
	3	0.0064	0.0064	0.0083
f_3	2	0.0031	0.0072	0.0072
	3	0.0070	0.0106	0.0185
f_4	1	0.0013	0.0041	0.0095
	1.7	0.0044	0.0066	0.0097
	-0.3	0.0053	0.0054	0.0138
f_5	3.5	0.0066	0.0103	0.0154
	2.5	0.0055	0.0076	0.0144
f_6	1.5	0.0054	0.0051	0.0101
$f(x)$	x_0	***Running time* (ms)**		
		N	**NT**	**NTH**
f_7	-2	0.0108	0.0207	0.0166

f_8	5	0,0149	0,2730	0,0324
f_9	3,5	0,0146	0,0236	0,0144
	3,25	0,0092	0,0080	0,0142

Berdasarkan Tabel 2 terlihat bahwa metode yang diusulkan dalam penelitian ini, yaitu metode NTH menghasilkan *running time* yang cukup besar dibandingkan dengan metode Newton dan metode Trapesium pada penggunaan toleransi 10^{-4} dan 10^{-14}. Tetapi jika dibandingkan dengan metode lainnya yang juga merupakan metode kombinasi, *running time* yang dihasilkan oleh metode NTH secara umum terlihat jauh lebih kecil. Hal ini dipengaruhi oleh banyaknya evaluasi fungsi yang dilakukan dalam satu kali iterasi dan besar *cost* yang dihasilkan pada setiap metode kombinasi. Tabel 3 menujukkan perbandingan jumlah iterasi dengan menggunakan toleransi 10^{-4}.

Tabel 3
Perbandingan jumlah iterasi menggunakan toleransi 10^{-4}

$f(x)$	x_0	**jumlah iterasi**		
		N	**NT**	**NTH**
f_1	-0,5	130	6	4
	1	4	3	2
	2	4	3	2
	-0,3	53	6	3
f_2	1	5	4	3
	3	5	3	2
f_3	2	4	4	3
	3	5	4	3
f_4	1	3	2	2
	1,7	4	3	2
	-0,3	5	3	3
f_5	3,5	6	4	3
	2,5	5	4	3
f_6	1,5	5	4	3
f_7	-2	7	5	3
f_8	5	8	118	4
f_9	3,5	11	8	4
	3,25	7	5	3

Berdasarkan Tabel 3, hasil dari jumlah iterasi yang diperoleh oleh masing-masing metode dengan manggunakan dua toleransi yang berbeda, dapat dilihat secara keseluruhan bahwa metode yang diusulkan, yaitu metode NTH menghasilkan jumlah iterasi yang lebih sedikit dibandingkan dengan metode

lainnya. Penggunaan titik tebakan awal x_0 dan batas toleransi sangat berpengaruh. Jika titik tebakan awal x_0 yang digunakan mendekati nilai sebenarnya, maka jumlah iterasi yang diperoleh akan semakin sedikit. Batas toleransi juga mempengaruhi proses pencarian akar, jika kriteria pemberhentian proses iterasi belum terpenuhi yaitu $|x_{n+1} - x_n| < toleransi$ maka proses pencarian solusi akan terus berjalan sampai kriteria pemberhentian proses iterasi terpenuhi. Pada Tabel 4 menunjukkan perbandingan nilai akar menggunakan toleransi 10^{-4}.

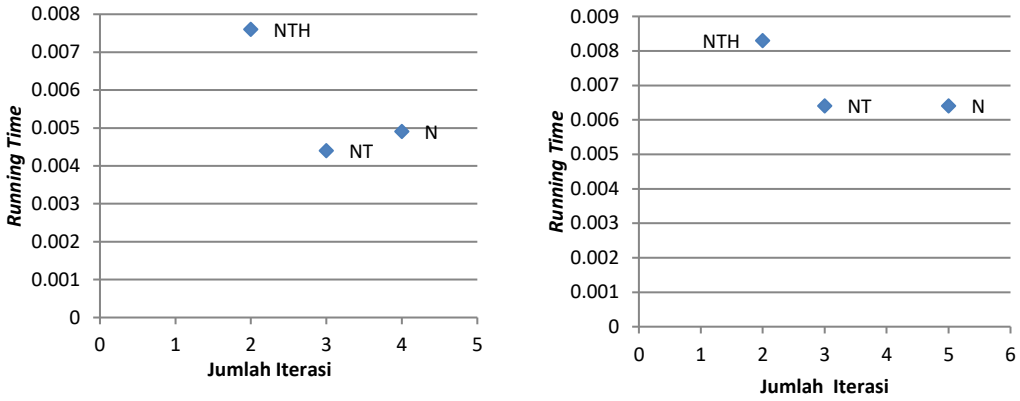
Tabel 4
Perbandingan nilai akar menggunakan toleransi 10^{-4}

$f(x)$	x_0	nilai akar		
		N	**NT**	**NTH**
f_1	-0.5	1.3652	1.3652	1.3652
	1	1.3652	1.3652	1.3652
	2	1.3652	1.3652	1.3652
	-0.3	1.3652	1.3652	1.3652
f_2	1	1.4044	1.4044	1.4044
	3	1.4044	1.4044	1.4044
f_3	2	0.2575	0.2575	0.2575
	3	0.2575	0.2575	0.2575
f_4	1	0.7391	0.7391	0.7391
	1.7	0.7391	0.7391	0.7391
	-0.3	0.7391	0.7391	0.7391
f_5	3.5	2.0000	2.0000	2.0000
	2.5	2.0000	2.0000	2.0000
f_6	1.5	2.1544	2.1544	2.1544
f_7	-2	-2.0000	-2.0000	-2.0000
f_8	5	3.4375	4.8246	4.6221
f_9	3.5	3.0000	3.0000	3.0000
	3.25	3.0000	3.0000	3.0000

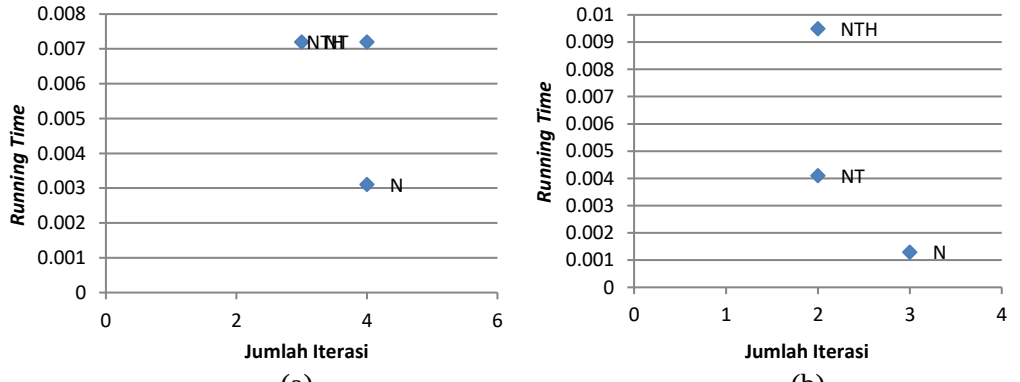
Tabel 4 memperlihatkan perbandingan nilai akar yang dihasilkan masing-masing metode dengan *true value*. *True value* yang digunakan pada penelitian ini berdasarkan hasil yang telah diperoleh oleh penelitian sebelumnya, yaitu penelitian yang dilakukan oleh Weerakoon [15]. Secara umum terlihat bahwa nilai akar yang dihasilkan dengan melakukan uji komputasi metode tidak menunjukkan perbedaan yang signifikan dengan *true value*. Tetapi pada f_8 terlihat perbedaan nilai akar yang cukup jauh antara *true value* dengan hasil yang diperoleh.

Gambar 1-5 menunjukkan grafik perbandingan antara jumlah iterasi dan *running time* yang dihasilkan dengan melakukan percobaan uji komputasi menggunakan batas toleransi 10^{-4}. Perbandingan tersebut memperlihatkan bahwa *running time* tidak bergantung kepada jumlah iterasi yang dihasilkan. Hal ini

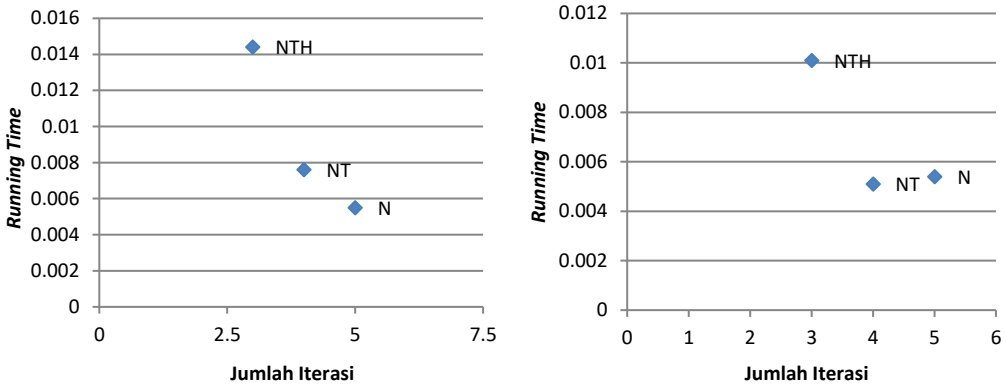
terjadi karena masing-masing metode yang digunakan melakukan jumlah evaluasi fungsi kerja yang berbeda-beda untuk setiap masing-masing iterasi, sehingga memengaruhi waktu yang digunakan untuk melakukan pencarian solusi. Untuk metode yang diusulkan yaitu metode NTH, *running time* yang dimiliki cukup besar dibandingkan beberapa metode lainnya seperti metode Newton dan metode Trapesium. Hal ini dikarenkana metode NTH melakukan tiga kali evaluasi fungsi sedangkan metode Newton hanya melakukan satu kali evaluasi fungsi untuk setiap iterasinya, tetapi jika dilihat dari jumlah iterasi yang dihasilkan, penggunaan metode NTH terlihat lebih unggul dibandingkan metode-metode lainnya, termasuk metode Newton yang memiliki *running time* lebih kecil dibandingkan dengan metode NTH.



Gambar 1 Perbandingan jumlah iterasi dan *running time* menggunakan toleransi 10^{-4} (a) f_1 dengan x_0 = 1 (b) f_2 dengan x_0 = 3

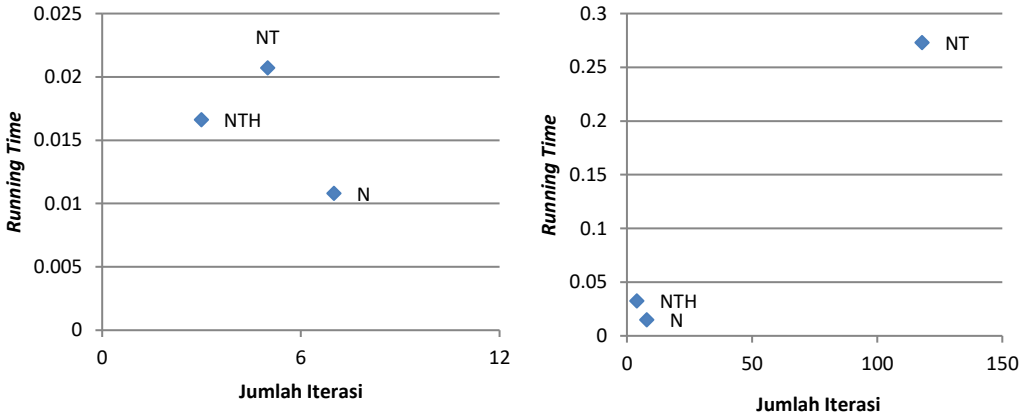


Gambar 2 Perbandingan jumlah iterasi dan *running time* menggunakan toleransi 10^{-4} (a) f_3 dengan x_0 = 2 (b) f_4 dengan x_0 = 1



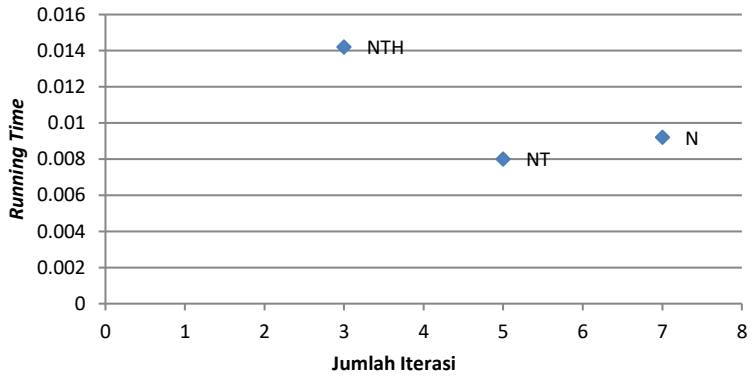
(a) (b)

Gambar 3 Perbandingan jumlah iterasi dan *running time* menggunakan toleransi 10^{-4} (a) f_5 dengan x_0 = 2.5 (b) f_6 dengan x_0 = 1.5



(a) (b)

Gambar 4 Perbandingan jumlah iterasi dan *running time* menggunakan toleransi 10^{-4} (a) f_7 dengan x_0 = -2 (b) f_8 dengan x_0 = 5



Gambar 5 Perbandingan jumlah iterasi dan *running time* dengan toleransi 10^{-4} menggunakan f_9 dengan x_0 = 3.25

4 SIMPULAN

Metode Newton merupakan salah satu metode numerik terbaik yang digunakan untuk mencari nilai akar dari suatu fungsi. Akan tetapi pencarian solusi dengan menggunakan metode Newton masih menghasilkan jumlah iterasi yang cukup banyak. Dengan mengkombinasikan metode Newton dengan Aturan Trapesium dan metode Halley, perbandingan antara beberapa metode memperlihatkan bahwa hasil yang diperoleh dari segi jumlah iterasi pada kombinasi metode NTH lebih bagus. Jika dilihat dari *running time* yang digunakan metode ini membutuhkan waktu yang cukup lama dibandingkan dengan metode yang proses pencarian akarnya hanya melakukan satu kali evaluasi fungsi dalam satu kali iterasi. Oleh sebab itu, hasil yang diperoleh dengan melakukan uji komputasi dalam penelitian ini menunjukkan bahwa kombinasi algoritme metode Newton, Aturan Trapesium, dan metode Halley baik digunakan untuk pencarian nilai akar dari fungsi nonlinear.

DAFTAR PUSTAKA

[1] Cerdà V, Cerdà JL, Idris AM. 2015. *Optimization using the gradient and simplex methods*. *Talanta*. Doi: 10.1016/j.talanta.2015.05.061.

[2] Chavnov JR. 2012. *Introduction to Numerical Method*. The Hongkong University of Science and Technology, Hongkong: Creative Commons Attribution 3.0 Hong Kong

[3] Cormen TH, Leiserson CE, Rivest RL, Stein C. 2009. *Introduction to Algorithm Third Edition.* London, England: The MIT Press.

[4] Homeier HHH. 2005. *On Newton-type Methods with Cubic Convergence*. Journal of Computational and Applied Mathematic. 176: 425-432.

[5] Jain D. 2013. *Families of Newton-Like Methods with Fourth-Order Convergence.* International Journal of Computer Mathematic. 90(5): 1072-1082.

[6] Kumar M, Singh AK, Srivastava A. 2012. *Various Newton-type Iterative Methods for Solving* Nonlinear *Equations.* Journal of the Egyptian Mathematic Society. 21: 334-339.

[7] Luenberger DG, Ye Y. 2008. *Linear and* Nonlinear *Programing Third Edition*. Stanford, California: Springer.

[8] Narang M, Bathia S, Kanwar GV. 2015. *Newton two-parameter Chebyshev-Halley-like family of fourth and sixth-order methods for systems of* nonlinear *equation.* Applied Mathematics and Computation. 275(2016): 394-403.

[9] Noor MA, Khan WA, Hussain A. 2007. *A new modified Halley method without second derivatives for* nonlinear *equation.* 189: 1268–1273.

[10] Putra S, Agusni, Restu YP. 2012. *Kombinasi Metode Newton dengan Metode Iterasi yang Diturunkan Berdasarkan Kombinasi Linear Beberapa Kuadratur untuk Menyelesaikan Persamaan* Nonlinear. Jurnal Sains, teknologi Industri. 10(1): 85-89

[11] Sánchez MG. 2009. *Improving Order and Efficiency: Composition with a Modified Newton's Method.* Journal of Computational and Applied Mathematics. 231: 592-597.

[12] Scavo TR, Thoo JB. 1994. American Mathematical Monthly.

[13] Snyman JA. 2005. *Practical Mathematical Optimization*. An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. University of Pretoria, Pretoria, South Africa : Springer.

[14] Utami NNR, Widan IN, Asih ND. 2013. *Perbandingan Solusi Sistem Persamaan* Nonlinear *Menggunakan Metode Newton-Raphson dan Metode Jacobian*. E-jurnal Matematika. 2(2): 11-17.

[15] Weerakoon S, Fernando TGI. 2000. *A Variant of* Newton *Raphson's Method with* Accelerated *Third-Order Convergence*. Applied Mathematics Letters. 13: 87-93.