KOMBINASI VARIAN METODE NEWTON DAN METODE HALLEY UNTUK MENYELESAIKAN PERSAMAAN TAK LINIER

D. A. PRATAMASYARI[1], B. P. SILALAHI[2], S. GURITMAN[3]

Abstrak

Menemukan akar dari suatu fungsi tak linier sering menjadi masalah dalam berbagai disiplin ilmu. Dalam kenyataannya menemukan akar dari suatu fungsi tak linier tidak mudah ditemukan secara analitik. Menetukan akar dari suatu fungsi tak linier yang lebih sulit dapat dilakukan dengan pendekatan numerik. Metode Newton merupakan salah satu metode yang baik dalam menentukan nilai akar. Metode Newton tanpa modifikasi menghasilkan lebih banyak iterasi yang dampaknya adalah memperbanyak eksekusi waktu, hal ini menyebabkan metode Newton menjadi tidak efisien. Penelitian ini bertujuan untuk mengombinasikan varian metode Newton dan halley yang diberi nama Newton, midpoint, halley (NMH). Hasil numerik dari penelitian ini menunjukkan bahwa metode NMH bisa mereduksi jumlah iterasi dan *running time*.

Kata kunci*:* akar, iterasi, metode Halley, metode Midpoint, metode Newton

1 PENDAHULUAN

Dalam matematika terapan sering ditemui masalah untuk mencari penyelesaian persamaan yang berbentuk $(x) = 0$, di mana persamaan dapat berbentuk sebagai persamaan aljabar, persamaan transenden atau persamaan campuran. Nilai-nilai x yang memenuhi disebut akar persamaan. Masalah menemukan akar dari suatu persamaan tak linier ini merupakan masalah yang sering muncul dalam berbagai disiplin ilmu. Dalam kenyataannya, akar-akar persamaan tak linier tersebut tidak mudah untuk ditemukan secara analitik, kecuali pada kasus-kasus sederhana. Oleh sebab itu, alasan utama mengapa penyelesaian masalah pencarian akar persamaan tak linier memerlukan pendekatan numerik disebabkan karena penyelesaian menggunakan cara analitik biasanya akan menemui kesulitan, meskipun persamaan tersebut kelihatannya sederhana. Hal inilah yang menjadi sebab mengapa metode numerik menjadi sangat diperlukan dalam memecahkan persoalan-persoalan dalam bidang sains dan teknologi bahkan ekonomi sekalipun. Praktek rekayasa di bidang ekonomi baik yang mensyaratkan bahwa semua proyek, produksi, dan perencanaan harus didekati dengan cara biaya yang efektif. Masalah ini dinamakan ”masalah pulang-pokok”. Masalah ini dipergunakan untuk menentukan titik pada mana dua pilihan alternatif setara. Pilihan-pilihan demikian dihadapi dalam semua bidang rekayasa. Walaupun terlihat sederhana namun akan sangat rumit apabila masalah tersebut tidak dapat diselesaikan secara analitik atau manual, oleh karena itu masalah

---

[1] Mahasiswa S2, Program Studi Matematika Terapan, Sekolah Pascasarjana IPB, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. Email: dedearseyani@gmail.com

[2] Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. E-mail : bibparuhumsilalahi@gmail.com

[3] Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680. E-mail: guritman@yahoo.com

pulang pokok pada perusahaan-perusahaan dapat diselesaikan dengan menggunakan metode numerik.

Metode Newton merupakan satu dari teknik terbaik untuk menyelesaikan persamaan nonlinear. Metode ini sangat mudah untuk diimplementasikan dan sering konvergen dengan cepat [3], bila iterasi dimulai cukup dekat dengan akar yang diinginkan. Metode Newton juga merupakan salah satu metode terbaik untuk menentukan solusi akar dari persamaan tak linier [8]. Pada perkembangannya metode ini telah mengalami banyak kemajuan, tidak hanya mencari akar dari suatu fungsi, namun metode ini juga digunakan untuk mencari titik optimal dari suatu persamaan dalam optimasi tak linear. Dalam beberapa tahun terakhir pengembangan metode iterasi untuk menentukan akar persamaan tak linier telah dilakukan dengan cara memodifikasi metode yang ada ataupun mengombinasikan pemakaiannya secara bersama-sama. Seperti yang telah di kembangkan oleh beberapa peneliti diantaranya mereka memodifikasi metode Newton menggunakan aturan trapesium sehingga menghasilkan metode Trapesium Newton yang memiliki orde kekonvergenan kubik metode Werakon Fernando (WF) lebih baik dari metode Newton [9]. Hasil dari metode WF telah memicu banyak penelitian terhadap metode Newton. Penelitian tersebut dilakukan untuk mendapatkan algoritma pencarian nilai akar fungsi tak linear dan memungkinkan untuk meningkatkan orde kekonvergenannya kesuatu nilai. Peneliti berikutnya Frontini dan Ozban dengan mengaproksimasikan integral Newton menggunakan aturan *midpoint* yang hasilnya mendapatkan metode *midpoint* Newton [6], rata-rata Aritmatik Newton untuk metode Aritmatik Newton dan rata-rata Harmonik Newton untuk metode Harmonik Newton. Seluruh metode ini juga menghasilkan orde kekonvergenan kubik yang sama dengan metode WF [1]. Peneliti berikutnya Putra [7] mengombinasikan metode *secant* dan *midpoint* Newton yang hasil dari iterasinya cukup sebanding dengan metode *secant*-Trapesium Newton yang sebelumnya diusulkan Jain [2], namun hasil dari penelitian ini salah satu fungsi yang diuji cobakan yaitu $x\exp(x^2) - sin^2(x) + 3\cos(x) + 5$ menghasilkan iterasi yang cukup besar, maka perlu perbaikan modifikasi agar fungsi-fungsi yang lebih rumit bisa menghasilkan iterasi yang kecil. Berikutnya Noor memodifikasi metode Halley dengan konsep dasar metode Newton dan menghasilkan beberapa pengembangan baru dari metode Halley sehingga menghasilkan performa yang lebih baik [5]. Dari penelitian Noor menghasilkan metode Halley yang performanya cukup baik, oleh karena itu metode Halley bisa menjadi pilihan kombinasi dengan metode lainnya agar hasil dari modifikasi menghasilkan metode yang lebih baik lagi. Merujuk hasil-hasil penelitian sebelumnya maka penulis mencoba untuk mengobinasikan metode numerik antara lain metode Newton, *midpoint*, dan Halley.

Berdasarkan beberapa penelitian yang dilakukan sebelumnya, banyaknya iterasi yang dihasilkan masih cukup besar dan tidak menunjukkan hasil r*unning time* program, maka tujuan penelitian ini adalah mengombinasikan metode Newton, metode *midpoint* Newton dan metode Halley guna memperoleh iterasi yang cepat namun konvergen mendekati nilai eksak. Serta membandingkan secara ekperimental hasil uji komputasi dari kombinasi Metode Newton Midpoint Halley (NMH) dengan metode Newon tanpa kombinasi dari segi iterasi, NOFE (*Number of Function Evaluations*), dan *running time*.

2 TINJAUAN PUSTAKA

Metode numerik yang paling sering digunakan untuk mencari akar persamaan tak linier adalah metode Newton,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \text{ n} = 0,1,2..... \quad (1)$$

yang memiliki orde kekonverganan kuadratik. Metode ini sangat populer sehingga para peneliti banyak melakukan modifikasi terhadap metode ini. Seperti yang telah dilakukan Wereakoon dan Fernando dengan menggunakan teorema Newton

$$f(x) = f(x_n) + \int_{x_n}^{x} f'(t)\ dt \quad (2)$$

dan mengaproksimasi nilai integral dengan aturan Trapesium, yaitu

$$\int_{x_n}^{x} f'(t) = (\frac{x-x_n}{2})(f'(x) + f'(x_n)) \quad (3)$$

Apabila persamaan (3) disubstitusi ke persamaan (2) maka diperoleh varian lain dari metode Newton yang dikenal dengan nama metode Trapesium Newton

$$x_{n+1} = x_n - \frac{2\,f(x_n)}{f'(x_n^*) + f'(x_n)} \quad (4)$$

di mana x_n^* dihitung dengan menggunakan metode Newton. Metode Trapesium Newton ini telah ditunjukkan Weerakoon dan Fernando memiliki orde kekonverganan kubik. Apabila nilai integral diaproksimasi dengan metode *Midpoint* (titik tengah)

$$\int_{x_n}^{x} f'(t)\ dt = (x - x_n) f'(\frac{x-x_n}{2}) \quad (5)$$

Peneliti selanjutnya yang bernama Ozban berhasil memperoleh formula yang dikenal dengan nama metode *midpoint* Newton dengan bentuk sebagai berikut, di mana x_n^* dihitung dengan menggunakan metode Newton

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(\frac{x_n^*+x_n}{2})} \quad (6)$$

Metode Halley merupakan salah satu metode perluasan dari metode Newton, formula metode Halley diperoleh dengan menggunakan turunan dari deret taylor polynomial tingkat dua. Bentuk umum dari formula metode Halley diformulasikan oleh Noor sebagai berikut [5] :

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)} \quad (7)$$

3 METODE

3.1 Tahapan Penelitian

Penelitian ini disusun melalui tiga tahap, pertama dilakukan telaah pustaka mengenai metode varian Newton dan Metode Halley. Pada tahap pertama ini akan mengombinasikan tiga metode yaitu metode Newton, metode Midpoint dan metode Halley. Selanjutnya pada tahap kedua mengombinasikan ketiga metode tersebut. Kemudian pada tahap ketiga, mengimplementasikan metode ke dalam bahasa pemograman dengan menggunakan *software* Matlab. Setelah itu, dilakukan pengujian untuk beberapa fungsi tak linier kemudian ditinjau dari segi iterasi, NOFE, dan *running time*.

3.2 Kombinasi Varian Metode Newton dan Metode Halley

Tahap ini dilakukan untuk mengombinasikan metode Newton, *midpoin*t dan metode Halley. Setelah diperoleh hasil kombinasi metode Newton, *midpoint* dan metode Halley kemudian metode-metode tersebut akan digabungkan sehingga dapat diketahui proses evaluasi fungsi dalam satu kali iterasi. Proses penggabungan metode yaitu dengan menyubstitusikan hasil perhitungan satu metode ke metode berikutnya dalam setiap iterasi.

3.3 Pembuatan Algoritma

Pada tahap ini akan dilakukan kombinasi algoritma dari ketiga metode yaitu metode Newton, metode Midpoint, dan metode Halley yang dirancang untuk menyelesaikan persamaan tak linier.

3.4 Pengujian Komputasi

Pada tahap ini akan dilakukan uji komputasi untuk membandingkan kemampuan metode yang diusulkan (NMH) dengan metode Newton tanpa kombinasi dengan menggunakan beberapa contoh fungsi tak linier. Hasil uji komputasi ini digunakan untuk melihat jumlah iterasi, NOFE, dan waktu tempuh (*running time*) yang dibutuhkan metode NMH untuk menyelesaikan beberapa fungsi yang telah diberikan. Berikut adalah persamaan yang digunakan utuk uji komputasi (Weerakoon [9]) :

1. $f_1(x) = x^3 + 4\,x^2 - 10$
   dengan $\alpha = 1.36523001341448$
2. $f_2(x) = sin^2(x) - x^2 + 1$
   dengan $\alpha = 1.40449164821621$
3. $f_3(x) = x^2 - e^x - 3x + 2$
   dengan $\alpha = 0.257530285439771$
4. $f_4(x) = \cos(x) - x$
   dengan $\alpha = 0.739085133214758$
5. $f_5(x) = (x-1)^3 - 1$
   dengan $\alpha = 2.0$
6. $f_6(x) = x^3 - 10$
   dengan $\alpha = 2.15443469003367$
7. $f_7(x) = x\exp(x^2) - sin^2(x) + 3\cos(x) + 5$
   dengan $\alpha = -1.20764782713013$
8. $f_8(x) = x^2 sin^2(x) + \exp[x^2 \cos(\mathrm{x})\sin(\mathrm{x})] - 28$
   dengan $\alpha = 4.82458931731526$
9. $f_9(x) = \exp(x^2 + 7x - 30) - 1$
   dengan $\alpha = 3.0$

4 HASIL DAN PEMBAHASAN

4.1 Kombinasi Varian Metode Newton dan Metode Halley

Kombinasi metode yang digunakan pada penelitian ini adalah metode Newton, metode Midpoint dan metode Halley yang mana ketiga metode tersebut tergolong dalam metode iteratif. Formula dari kombinasi ketiga metode tersebut adalah :

$$x_{n+1} = \bar{x}_n - \frac{2f(\bar{x}_n)f'(\bar{x}_n)}{2\left(f'(\bar{x}_n)\right)^2 - f(\bar{x}_n)f''(\bar{x}_n)} \tag{8}$$

di mana, $\bar{x}_n$ dihitung dengan mneggunakan metode midpoint Newton yang telah diformulasikan oleh peneliti bernama Ozban [6] sebagai berikut :

$$\bar{x}_n = x_n - \frac{f(x_n)}{f'\left(\frac{x_n^* + x_n}{2}\right)} \tag{9}$$

Kemudian nilai x_n^* dihitung dengan menggunakan metode Newton.

4.2 Pembuatan Algoritma

Hasil kombinasi metode direkonstruksi dalam bentuk algoritma yang mana setiap langkah penyelesaian untuk mengeksekusi sebuah fungsi disesuaikan dengan langkah-langkah pada formula metode yang diusulkan. Adapun algoritma metode Newton, metode midpoint Newton dan metode Halley dituliskan dalam tahap-tahap sebagai berikut :

Algoritma NMH (Newton Midpoint Halley)

Langkah 1. Diberikan fungsi awal $f(x)$

Langkah 2. Diberikan titik tebakan awal $x_n \epsilon \mathbb{R}^n$ dan batas toleransi $0 < \varepsilon < 1$

Langkah 3. Diberikan batas maksimum iterasi

Langkah 4. Hitung $x_n^* = x_n - \frac{f(x_n)}{f'(x_n)}$

Hitung $\bar{x}_n = x_n - \frac{f(x_n)}{f'\left(\frac{x_n^* + x_n}{2}\right)}$

Hitung $x_{n+1} = \bar{x}_n - \frac{2f(\bar{x}_n)f'(\bar{x}_n)}{2(f'(\bar{x}_n))^2 - f(\bar{x}_n)f''(\bar{x}_n)}$

Langkah 5. Hitung $b = |x_{n+1} - x_n| < \varepsilon$

Langkah 6. Kembali ke langkah 4

Penentuan kriteria pemberhentian untuk solusi numerik pada setiap metode mempunyai kriteria pemberhentian yang sama. Salah satu kriteria untuk pembatasan proses iterasi program komputasi adalah selisih dua nilai titik x terakhir kurang dari batas toleransi ($\varepsilon = 10^{-4}$) yang diberikan. Jika kondisi $|x_{n+1} - x_n| < \varepsilon$ terpenuhi maka proses iterasi komputasi akan berhenti.

Pada algoritma metode Newton Midpoint Halley membutuhkan titik tebakan awal yaitu x_n pada metode Newton, maka dari itu proses metode newton diletakkan di awal, kemudian hasil dari metode newton disubstitusi ke metode *midpoint,* setelah itu hasil dari metode *midpoint* disubstitusi lagi ke metode Halley di mana pada metode Halley membutuhkan turunan fungsi pertama dan kedua dari hasil substitusi metode *midpoint,* di mana *cost* dari metode halley lebih banyak dibandingkan metode lainnya sehingga ekesekusi pada metode halley diposisikan pada bagian akhir.

4.3 Pengujian Komputasi

Pada bagian ini, dilaukan uji komputasi untuk membandingkan iterasi, NOFE (banyaknya fungsi yang dievaluasi), dan *running time.* Pengujian komputasi dilakukan dengan menggunakan beberapa fungsi tak linier yang telah digunakan oleh peneliti sebelumnya [9]. Adapun beberapa metode yang diuji yaitu, Newton, NMS (Newton Midpoint secant), Halley, NH (Newton Halley), MH (Midpoint Halley), dan NMH (Newton Midpoint Halley).

Uji komputasi dilakukan untuk kasus fungsi tak linier dengan akar tunggal. Untuk masing-masing fungsi nonlinear pada uji komputasi ini memiliki nilai tebakan awal yang berbeda-beda yang telah dilkukan oleh peneliti sebelumnya [9].

Tabel 1
Perbandingan jumlah iterasi masing-masing metode untuk toleransi 10^{-4}

$f(x)$	x_0	iterasi					
		N	**NMS**	**H**	**NH**	**MH**	**NMH**
f_1	-0.5	130	6	74	8	34	3
	1	4	4	4	3	3	3
	2	4	4	4	3	3	3
	-0.3	53	63	53	25	6	6
f_2	1	5	5	5	3	3	3
	3	5	5	5	4	4	4
f_3	2	2	div	2	2	2	2
	3	5	5	4	4	4	4
f_4	1	3	3	44	3	3	3
	1.7	4	4	4	3	3	3
	-0.3	5	4	5	4	3	3
f_5	3.5	6	6	4	4	2	4
	2.5	5	4	4	4	3	3
f_6	1.5	5	5	4	3	3	3
f_7	-2	7	6	5	4	4	4
f_8	5	8	8	7	5	4	3
f_9	3.5	11	9	5	4	6	6
	3.25	7	6	4	4	4	4
JUMLAH		269	147	197	90	94	64

Berdasarkan hasil uji komputasi dari aspek banyak iterasi pada Tabel 1 untuk beberapa metode yang berbeda menunjukkan bahwa kombinasi metode yang diusulkan yaitu metode NMH lebih unggul dibandingkan metode Newton dan metode lainnya dengan total banyak terasi yang lebih sedikit. Untuk beberapa kasus fungsi nonlinear pada metode NMS. Selanjutnya jika metode NMH dibandingkan dengan kombinasi metode H, NH dan MH untuk beberapa kasus fungsi, banyak iterasi yang diperoleh tidak jauh berbeda nyata namun kombinasi metode NMH bisa dikatakan masih lebih baik dari segi iterasinya. Nilai tebakan awal x_0 dan toleransi juga berpengaruh pada banyaknya iterasi. Jika menggunakan titik awal atau tebakan awal yang cukup dekat dengan nilai akarnya, maka iterasi menjadi lebih sedikit. Untuk besar toleransi yang diberikan juga berpengaruh terhadap jalannya proses komputasi. Jika kriteria pemberhentian proses iterasi terpenuhi yaitu $|x_{k+1} - x_k| < \varepsilon$ maka proses komputasi akan berhenti sehingga pada beberapa kasus fungsi tidak memperoleh solusi atau *divergen* seperti hasil yang diperoleh metode NMS yaitu pada f_3 dengan dengan $x_0 = 2$ dikarenakan selang yang dibentuk oleh metode sebelumnya yaitu Newton dan *midpoint* tidak memuat nilai akarnya, sehingga metode NMS tidak menemukan solusi atau dikatakan *divergen.*

Tabel 2
Perbandingan NOFE masing-masing metode

$f(x)$	x_0	NOFE (Number Of Function Evaluation)					
		N	**NMS**	**H**	**NH**	**MH**	**NMH**
f_1	-0.5	260	24	222	40	170	18
	1	8	16	12	15	15	18
	2	8	6	12	15	15	18
	-0.3	106	252	159	125	30	36
f_2	1	10	20	15	15	15	18
	3	10	20	15	20	20	24
f_3	2	4	div	6	10	10	12
	3	10	20	12	20	20	24
f_4	1	6	12	12	15	15	18
	1.7	8	6	12	15	15	18
	-0.3	10	6	15	20	15	18
f_5	3.5	12	24	12	20	10	24
	2.5	10	16	12	20	15	18
f_6	1.5	10	20	12	15	15	18
f_7	-2	14	24	15	20	20	24
f_8	5	16	32	21	25	20	18
f_9	3.5	22	36	15	20	30	36
	3.25	14	24	12	20	20	24
JUMLAH		538	534	369	450	470	384

Berdasarkan perhitungan nilai NOFE atau jumlah fungsi yang dievaluasi untuk setiap metode berbeda-beda. Pada setiap iterasi, metode Newton memerlukan dua kali evaluasi, metode NMS memerlukan empat kali evaluasi fungsi, metode Halley memerlukan tiga kali evaluasi fungsi, metode NH memerlukan lima kali evaluasi fungsi, metode MH memerlukan lima kali evaluasi, dan metode NMH memerlukan enam kali evaluasi fungsi, setelah itu jumlahan evaluasi fungsi dikalikan dengan banyaknya iterasi. Dari ke enam metode di atas metode NMH yang paling banyak melakukan evaluasi fungsi pada setiap iterasi, yaitu sebanyak 6 kali namun iterasi yang dihasilkan oleh metode NMH paling kecil sehingga NOFE yang dihasilkan metode NMH paling sedikit dibandingkan dengan metode lainnya kecuali metode Halley. Sehingga total nilai NOFE untuk sembilan buah fungsi yang dipilih, paling sedikit dihasilkan oleh metode Halley sebanyak 369 dan metode NMH sebanyak 384.

Tabel 3
Perbandingan *running time* masing-masing metode untuk toleransi 10^{-4}

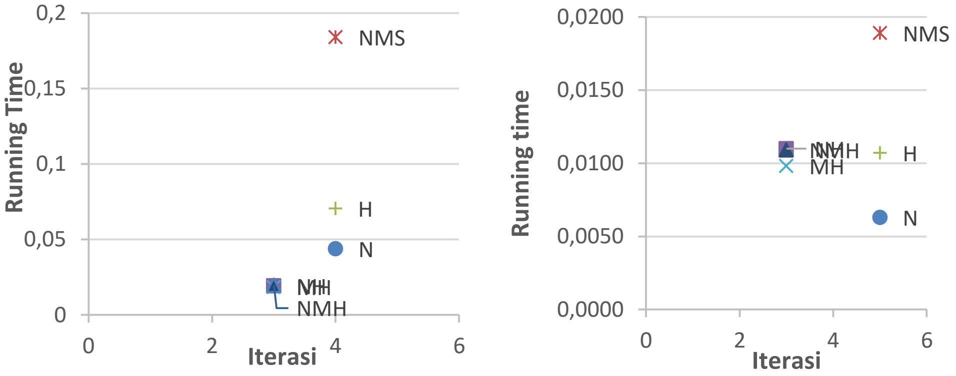
$f(x)$	x_0	*Running time (ms)*					
		N	**NMS**	**H**	**NH**	**MH**	**NMH**
f_1	-0.5	0.1204	0.0220	0.0822	0.0280	0.1052	0.0134
	1	0.0049	0.0243	0.0072	0.0096	0.0091	0.0097
	2	0.0049	0.0138	0.0059	0.0100	0.0091	0.0107
	-0.3	0.0438	0.1839	0.0705	0.0193	0.0184	0.0196
f_2	1	0.0063	0.0189	0.0107	0.0110	0.0098	0.0109
	3	0.0064	0.0158	0.0108	0.0098	0.0131	0.0160
f_3	2	0.0031	3.0553	0.0064	0.0073	0.0072	0.0108
	3	0.0070	0.0167	0.0122	0.0123	0.0141	0.0180
f_4	1	0.0013	0.0086	0.0067	0.0062	0.0086	0.0107
	1.7	0.0044	0.0107	0.0087	0.0061	0.0080	0.0121
	-0.3	0.0053	0.0108	0.0087	0.0089	0.0085	0.0130
f_5	3.5	0.0066	0.0174	0.0089	0.0093	0.0819	0.0099
	2.5	0.0055	0.0143	0.0088	0.0093	0.0086	0.0093
f_6	1.5	0.0054	0.0141	0.0074	0.0090	0.0086	0.0089
f_7	-2	0.0108	0.0590	0.0143	0.0144	0.6774	0.0219
f_8	5	0.0149	0.0247	0.0246	0.0299	0.0229	0.0243
f_9	3.5	0.0146	0.0309	0.0182	0.0137	0.1556	0.0203
	3.25	0.0092	0.0208	0.0151	0.0166	0.0159	0.0165
JUMLAH		0.2748	3.5620	0.3273	0.2307	1.1820	0.2560

Berdasarkan hasil uji komputasi dari aspek *running time* menggunakan enam kombinasi metode. Secara umum dapat dilihat bahwa metode NMH memperoleh nilai *running time* yang terkecil dibandingkan dengan metode Newton pada saat eksekusi program. Namun pada kasus fungsi transenden *running time* metode NMH cukup besar. Hal ini terjadi karena dalam satu kali proses iterasi, metode NMH melakukan tiga kali pengulangan metode dalam setiap iterasi dan metode halley membutuhkan cost yang lebih banyak dibandingkan metode lain sehingga *running time* program menjadi lebih besar daripada metode Newton. Besarnya *cost* setiap kombinasi metode juga berpengaruh untuk eksekusi *running time*. Semakin besar *cost* suatu formula metode, maka *running time* yang dihasilkan suatu kombinasi metode juga akan meningkat.

Tabel 4
Perbandingan nilai akar masing-masing metode dengan toleransi 10^{-4}

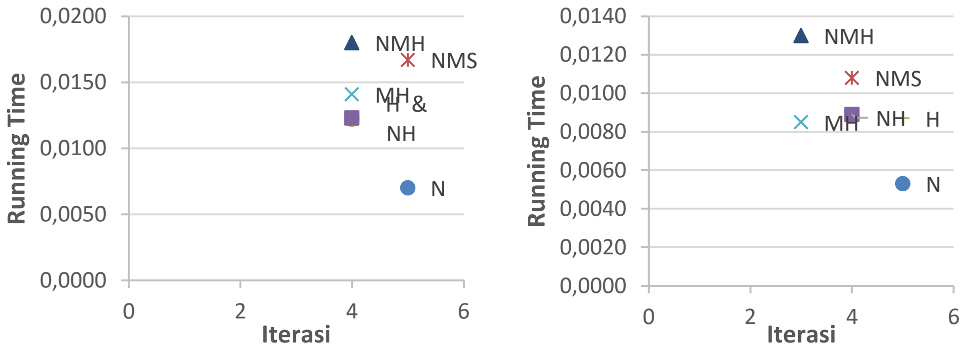
$f(x)$	x_0	Nilai akar						
		True Value	**N**	**NMS**	**H**	**NH**	**MH**	**NMH**
f_1	-0.5	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652
	1	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652
	2	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652
	-0.3	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652	1.3652
f_2	1	1.4044	1.4044	1.4044	1.4044	1.4044	1.4044	1.4044
	3	1.4044	1.4044	1.4044	1.4044	1.4044	1.4044	1.4044
f_3	2	0.2575	0.2575	0.2575	0.2575	0.2575	0.2575	0.2575
	3	0.2575	0.2575	0.2575	0.2575	0.2575	0.2575	0.2575
f_4	1	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391
	1.7	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391
	-0.3	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391	0.7391
f_5	3.5	2	2	2	2	2	2	2
	2.5	2	2	2	2	2	2	2
f_6	1.5	2.1544	2.1544	2.1544	2.1544	2.1544	2.1544	2.1544
f_7	-2	-1.2076	-1.2076	-1.2076	-1.2076	-1.2076	-1.2076	-1.2076
f_8	5	4.8246	3.4375	3.4375	3.4375	3.4374	4.7119	4.7119
f_9	3.5	3	3	3	3	3	3	3
	3.25	3	3	3	3	3	3	3

Berdasarkan hasil uji komputasi pada Tabel 4 diatas menunjukkan bahwa hampir semua kombinasi metode yang dibandingkan menemukan nilai akar yang diharapkan Selanjutnya nilai akar yang diperoleh dari hasil uji komputasi pada Tabel 4 menggunakan toleransi sebesar 10^{-4} tidak terdapat perbedaan nyata jika dibandingkan dengan nilai akar yang sebenarnya. Namun pada f_8 metode Newton menghasilkan nilai akar yang agak jauh dengan nilai sebenarnya, tetapi pada metode NMH menemukan nilai akar yang dihasilkan hampir mendekati nilai akar yang sebenarnya, karena proses evaluasi melalui tiga metode yang dikombinasikan sehingga menghasilkan nilai akar yang lebih dekat dengan nilai yang diharapkan.



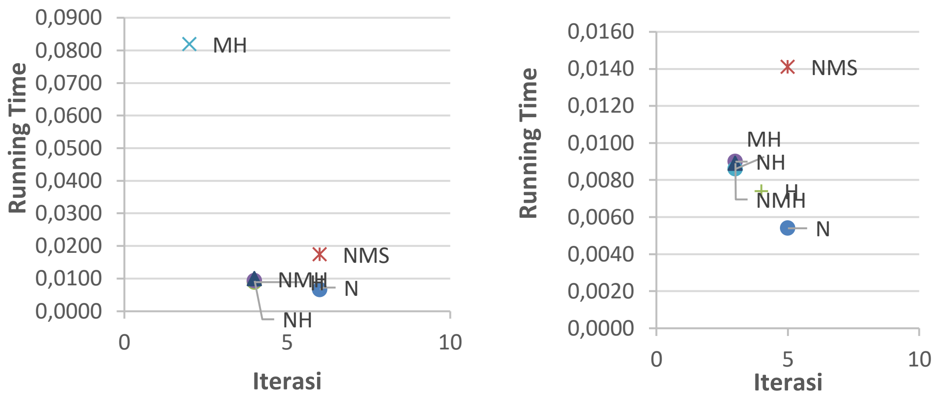
(a) (b)

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



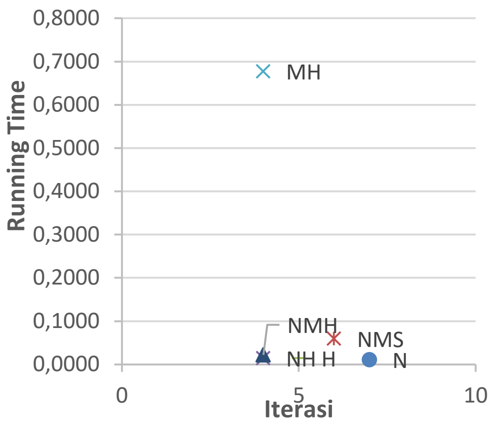
(a) (b)

Gambar 2 Perbandingan banyak iterasi dan *running time* (a) f_3 dengan $x_0 = 3$ toleransi 10^{-4} dan (b) f_4 dengan $x_0 = -0.3$ toleransi 10^{-4}

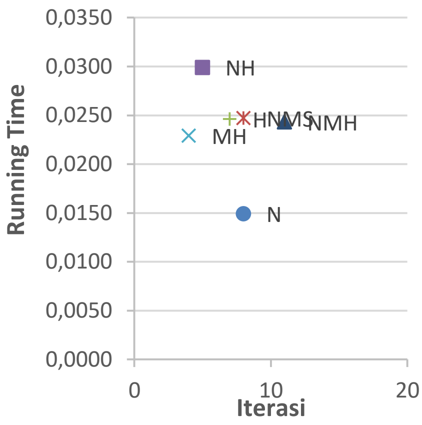


(a) (b)

Gambar 3 Perbandingan banyak iterasi dan *running time* (a) f_5 dengan $x_0 = 3.5$ toleransi 10^{-4} dan (b) f_6 dengan $x_0 = 1.5$ toleransi 10^{-4}

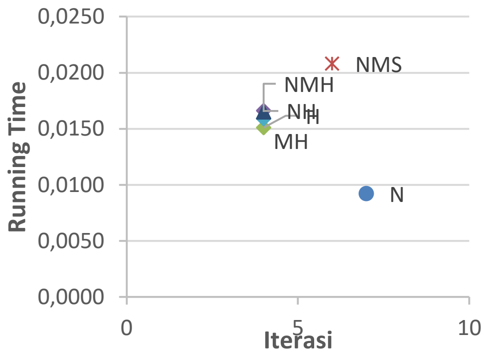


(a)



(b)

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



Gambar 5 Perbandingan banyak iterasi dan *running time* f_9 dengan $x_0 = 3$.

Berdasarkan hasil pengujian komputasi, dapat dilihat hubungan antara jumlah iterasi dan *running time* program yang dieksekusi. Untuk kasus toleransi 10^{-4} berdasarkan Gambar grafik semua fungsi, dapat dilihat bahwa masing-masing metode memiliki kecepatan tersendiri terhadap jalannya *running time*. Semakin kecil jumlah iterasi maka tidak menjamin bahwa semakin kecil *running time*. Sebaliknya jika jumlah iterasi besar, tidak menjamin akan semakin besar pula *running time*. Namun ada beberapa hal yang dapat mempengaruhi besarnya iterasi yaitu tebakan nilai awal fungsi dan kombinasi metode yang digunakan untuk menyelesaikan suatu fungsi.

4 SIMPULAN DAN SARAN

4.1 Simpulan

Hasil penelitian menunjukkan bahwa dengan menggunakan kombinasi metode Newton, midpoint Newton dan Halley (NMH) mampu mencari solusi akar yang diharapkan. Berdasarkan hasil percobaan uji komputasi, secara umum metode NMH mempunyai performa yang lebih unggul jika dibandingkan dengan metode Newton, NMS, H, NH, dan MH. Dapat dilihat dari total iterasi yang lebih sedikit dan total *running time* yang lebih singkat, serta total nilai NOFE yang paling sedikit.

4.2 Saran

Untuk penelitian selanjutnya, metode NMH (Newton Midpoint Halley) bisa diuji cobakan terhadap fungsi tak linier yang memuat 2 peubah untuk memperoleh suatu pasangan akar.

DAFTAR PUSTAKA

[1] Frontini M, Sormain E. 2003. Some Variant of Newton's Method with Third-Order Corvergence. *Applied Mathematic and Computation.* 140: 419-426.
[2] Jain D. 2013. Families of Newton-Like Methods with Fourth-Order Convergence. *International Journal of Computer Mathematic*. 90(5): 1072-1082.
[3] 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.
[4] Luenberger DG, Ye Y. 2008. *Linear and Nonlinear Programing Third Edition*. Stanford (CA) : Springer.
[5] Noor MA, Khan WA, Hussain A. 2006. A new modified Halley method without second derivatives for nonlinear equation. *Applied Mathematics and Computation.* 189: 1268–1273
[6] Ozban AY. 2004. Some New Variants of Newton's Methods. *Applied Mathematic Letters.* 17: 677-682.
[7] Putra. 2014. *Metode Secant-Midpoint Newton untuk menyelesaikan persamaan nonlinier.*[Ulasan]. *Jain*. 90(5): 1072-1082.
[8] Sánchez MG. 2009. Improving Order and Efficiency: Composition with a Modified Newton's Method. *Journal of Computational and Applied Mathematics.* 231: 592-597.
[9] Weerakoon S, Fernando TGI. 2000. A Variant of Newton's Method With Accelerated Third-Order Convergence. *Applied Mathematic Letters.* 13: 87-93.