RITEKTRA X
BANDUNG 2021

Prosiding Seminar Nasional
Riset dan Teknologi Terapan (RITEKTRA) 2021
Menuju *Society* 5.0: Teknologi Cerdas yang Berpusat pada Manusia
Bandung, 12 Agustus 2021

ISSN: 2807-999X

PERBANDINGAN METODE BRENT DAN BISECTION DALAM PENENTUAN AKAR GANDA PERSAMAAN BERBENTUK POLINOMIAL

Patrisius Batarius[1*]
[1]*Program Studi Ilmu Komputer, Fakultas Teknik, Universitas Katolik Widya Mandira, Jl. San Juan Penfui Kupang - NTT*

[*]*E-mail: patrisbatarius@unwira.ac.id*

ABSTRAK
Persamaan polinomial sering dijumpai pada persoalan-persoalan matematika dan bidang teknik. Tidak jarang persamaan polinomial tersebut memiliki akar kembar, ada yang jumlahnya genap dan ada yang jumlahnya ganjil. Tidak semua metode numerik bisa mencari akar kembar dari persamaan polinomial. Metode Bisection salah satu metode dalam pencarian akar dengan yang paling sederhana. Metode bisection berdasarkan pada teori nilai untuk fungsi kontinyu antara x_l *dan* x_u *Interval* $[x_l, x_u]$ *ganti dengan* $[x_r, x_u]$ *atau* $[x_l, x_r]$ *bergantung pada tanda perkalian* $f(x_l)*f(x_r)$. *Selain metode Bisection, metode Brent juga yang digunakan dalam pencarian akar. Metode Brent merupakan metode hybrid yaitu gabungan metode tertutup Bisection dan metode Inverse Quadratic Interpolation (IQI) dan metode terbuka Secant. Penelitian ini bertujuan membandingkan kedua metode, yaitu metode Brent dan metode Bisection dalam mencari akar persamaan yang berbentuk polinomial yang memiliki akar ganda. Persamaan polinomial yang digunakan dalam penelitian ini memiliki 3 akar yang mana 2 diantaranya adalah akar ganda dan persamaan polinomial yang memiliki 4 akar dan 3 diantaranya adalah akar ganda. Pemilihan nilai awal baik pada metode Brent maupun metode Bisection akan mempengaruhi akar polinomial yang dicari. Kedua metode memiliki model hasil pencarian akar yang sama terhadap persamaan polinomial yang diberikan.*

Kata kunci: *Metode Brent, metode bisection, persamaan polynomial, pencarian akar*

1. PENDAHULUAN

Fungsi f(x) = 0 sama artinya mencari akar persamaan tersebut. Bentuk persamaan bisa bermacam-macam seperti berbentuk polynomial, sinusoidal, eksponensial atau gabungan dari 2 atau lebih model persamaan. Pencarian akar persamaan yang berbentuk polynomial merupakan masalah penting bukan saja dalam bidang matematika tetapi juga dalam bidang lainnya seperti bidang sains dan ilmu rekayasa.

Metode *Bisection* dan juga metode posisi salah termasuk dalam metode tertutup dalam proses pencarian akar suatu persamaan. Hal ini terkait dengan pemilihan nilai awal sebagai batas bawah dan batas atas yang mengurung akar suatu persamaan. Berbeda dengan metode terbuka, seperti metode *Newton-Raphson* dan metode *Secant*, tidak terpaku pada pengurungan akar dalam proses penentuan nilai awal. Baik metode terbuka maupun metode tertutup memiliki kelemahan dan kelebihan masing-masing.

Pengembangan metode dalam pencarian akar juga banyak dilakukan. Penerapan dilakukan tidak saja dalam bidang matematika tetapi pada bidang teknik. Metode *Bisection* misalnya, dikembangan dengan metode *Newton-Raphson* untuk pencarian akar persamaan non-linear. Hasil dari algoritma hibrid antara dua metode tersebut lebih konvergensi dalam pencarian akar (Kim, dkk., 2017).

Pencarian akar ganda persamaan polinomial memiliki tingkat kesukaran tersendiri. Pengembangan metode *Newton-Raphson* dan metode *Secant* bisa menjadi solusi dalam mencari akar persamaan yang memiliki akar ganda baik yang jumlahnya genap maupun yang jumlahnya ganjil (Chapra, dkk., 2010).

Metode-metode dan pendekatan dalam mencari akar persamaan telah banyak digunakan. Pengembangan berbagai metode pun dilakukan untuk mencari akar persamaan dengan berbagai kelebihan dan kekurangan polynomnya masing-masing. Salah satu metode yang digunakan dalam penentuan akar persamaan adalah metode *Brent*. Metode *Brent* merupakan pengembangan metode interpolasi kuadratik (*Inverse Quadratic Interpolation* = IQI) dan metode *Dekker*. Sementara metode *Dekker* sendiri merupakan penggabungan metode *Bisection* dan metode *Secant*.

Pencarian akar ganda dengan metode *Secant* yang dimodifikasi atau metode *Newton-Raphson* yang dimodifikasi lebih mudah dicari daripada dengan metode *Newton-Raphson* atau metode *Secant* biasa (Chapra, dkk., 2010). Nilai awal yang dipilih dalam menyelesaikan persamaan polinomial yang memiliki akar kembar berpengaruh pada pencarian akar pada metode *Secant* yang dimodifikasi. Hasil yang diperoleh bisa akar kembar atau salah satu akar selain akar kembar (Batarius, dkk., 2019).

 ISSN: 2807-999X

Metode *Brent* yang merupakan gabungan metode terbuka dan tertutup, yaitu metode *Bisection*, metode *Secant* dan metode IQI, perlu dicoba kinerjanya dalam mencari akar kembar. Dengan demikian rumusan penelitian ini adalah bagaimana kinerja metode *Brent* dalam menentukan akar persamaan polinomial yang memiliki akar ganda. Penelitian ini bertujuan untuk membandingkan metode *Brent* dan metode *Bisection* dalam penentuan akar persamaan berbentuk polinomial. Parameter pengukuran dari kedua metode tersebut adalah, input yang sama untuk nilai tebakan awal. Nilai a sebagai tebakan batasan bawah dan b tebakan batasan atas pada metode *Brent* sama dengan nilai x_l xl dan x_u untuk metode *Bisection*. Selain itu, kriteria berhenti masing-masing metode jika sesuai dengan algoritma masing-masing metode. Algoritma yang efisien dilihat dari jumlah iterasi dalam mencapai atau menemukan akar dari fungsi yang dicari.

Pendekatan linearisasi untuk menyelesaikan persamaan non-linear menggunakan ekspansi deret *Taylor* (Temelkan, dkk., 2020). Algoritma *hybrid* antara metode *Bisection* dan kombinasi dari inverse dari deret sinus dan metode *Newton* menunjukkan efisiensi lebih baik dalam menemukan akar ke-n dari bilangan real (Pinkham, dkk., 2020).

Metode *gradient* konjugasi Dai-Liao dalam menyelesaikan persamaan non-linear skala besar. Metode ini menggabungkan metode *Secant* yang dimodifikasi menunjukan lebih efisien. Beberapa perbandingan dilakukan untuk membuktikan hal tersebut (Waziri, dkk., 2020). Sebuah metode konjugasi *gradient* yang berdasarkan pada modifikasi metode *Secant*, menunjukan kinerja komputasi yang efisien (Dehghani, dkk., 2019).

Skema iteratif yang efisien secara komputasi dalam menemukan akar ganda persamaan non-linear. Teknik yang digunakan memiliki konvergensi yang konsisten dan urutan ke-delapan dalam komputasi konvergensi. Metode ini lebih efisien dan waktu komputasi lebih sedikit (Sharma, dkk., 2020).

Metode Brent merupakan penggabungan atau hybrid dari beberapa metode yaitu metode *Secant*, metode Bagi Dua dan metode IQI (*Interval Quadratic Inverse*). Penelitian yang dilakukan Vakkalagadda, dengan mengganti metode *Secant* dengan metode posisi salah. Sehingga metode *Brent-Dekker* menjadi penggabungan yang melibatkan metode posisi salah, IQI (Interval Quadratic Inverse), dan metode bagi dua. Hasilnya menunjukan bahwa pencarian akar yang dilakukan dengan modifikasi tersebut lebih baik karena membutuhkan waktu eksekusi yang lebih sedikit daripada metode *Brent-Dekker* sebelumnya (Vakkalagadda, 2020).

Metode *Secant* dengan satu tebakan awal tingkat konvergen dalam komputasinya lebih baik daripada metode secant konvensional yang membutuhkan dua titik tetap atau dua titik awal dalam proses komputasi (Etin-Osa, dkk., 2020). Metode *Brent* juga digunakan untuk menyelesaikannya beberapa persoalan *polynomial fuzzy interval* tipe 2. Metode *Brent* yang digunakan, dimodifikasi terlebih dahulu. Metode *Brent* hasil modifikasi dapat menyelesaikan persamaan secara efisien dengan jumlah iterasi yang lebih rendah, lebih kecil kesalahan dan waktu komputasi lebih singkat dibandingkan dengan metode *Brent* biasa (Rahman, dkk., 2017).

Penelitian-peneltian dengan metode *Brent* yang diuraikan diatas belum mencoba persamaan berbentuk polinomial yang memiliki akar ganda. Pada penelitian ini membandingkan kinerja metode *Brent* dengan metode lainnya.

2. METODE

Penelitian ini dilakukan dengan proses eksperimen dengan mengambil 2 buah persamaan polinomial yang memiliki akar ganda. Persamaan pertama memiliki 2 akar ganda (genap) dan persamaan kedua memiliki 3 akar ganda (ganjil). Selanjutnya dilakukan proses simulasi menggunakan 3 metode yaitu metode *Brent*, metode *Bisection* dan metode *Secant* modifikasi.

Berikut langkah-langkah penelitian yang dilakukan.

1. Tentukan persamaan polynomial yang memiliki akar kembar.
   Pada penelitian ini, ada 2 persamaan untuk proses penelitian.
   a. Pertama persamaan yang memiliki memiliki 3 buah akar yang terdiri dari 1 akar tunggal dan 2 akar kembar (genap), $f(x) = (x+3)(x-1)^2$
   b. Persamaan kedua yaitu persamaan yang memiliki 4 buah akar yang terdiri dari 1 akar tunggal dan 3 akar kembar (ganjil), $f(x) = (x+3)(x-1)^3$
2. Tentukan nilai awal yang sama untuk ke-2 metode.
   Untuk metode *Brent* batas bawah adalah a sama dengan x_l untuk metode *Bisection*. Demikian juga untuk batas atas pada metode *Brent*, yaitu nilai b sama dengan x_u pada metode *Bisection*. Untuk metode *Brent* dan *Bisection*, pemilihan dua angka awal sebagai usaha untuk mengurung akar agar benar terdapat pada interval yang dipilih.
3. Lakukan proses simulasi pada ke-2 metode yang ada, yaitu dengan metode *Brent* dan metode *Bisection*. Proses simulasi dilakukan untuk mencari akar persamaan berbentuk polynomial yang memiliki akar kembar dengan menerapkan metode *Brent* dan metode *Bisection*.

ISSN: 2807-999X

4. Tampilkan hasil masing-masing metode dan lakukan proses perbandingan beberapa parameter. Parameter perbandingan yang dilakukan yaitu jumlah iterasi dari kedua metode yang merupakan indikasi dari kecepatan konvergensi dari metode yang digunakan. Selain itu, akurasi akar dari masing-masing metode dengan acuan nilai tebakan awal yang dipilih.
5. Penarikan kesimpulan.

2.1 Metode *Brent*

Metode *Brent* merupakan salah satu metode yang *powerful* dan mempunyai jangkauan yang luas pada permasalahan yang cukup sulit dalam penyelesaian pencarian akar persamaan. Metode *Brent* secara langsung tersedia pada *software* MATLAB yaitu dengan fungsi fzero yang dimilikinya. Adapun algoritma metode *Brent*, sebagai berikut.

Input a, b: real; var c: real
calculate f(a)
calculate f(b)
if f(a) f(b) >= 0 then error-exit end if
if |f(a)| < |f(b)| then swap (a,b) end if
c := a
set mflag
repeat until f(b or s) = 0 or |b − a| is
small enough /* convergence */
if f(a) ≠ f(c) and f(b) ≠ f(c) then

$$s \coloneqq \frac{af(b)f(c)}{\big(f(a)-f(b)\big)\big(f(a)-f(c)\big)} + \frac{bf(a)f(c)}{\big(f(b)-f(a)\big)\big(f(b)-f(c)\big)} + \frac{cf(a)f(b)}{\big(f(c)-f(a)\big)\big(f(c)-f(b)\big)}$$

else

$$s \coloneqq b - f(b)\,\frac{b-a}{f(b)-f(a)}$$

end if
if (condition 1) s is not between (3a + b)/4 and b
or (condition 2) (mflag is set and |s−b| ≥ |b−c| / 2)
or (condition 3) (mflag is cleared and |s−b| ≥ |c−d| / 2)
or (condition 4) (mflag is set and |b−c| < |δ|)
or (condition 5) (mflag is cleared and |c−d| < |δ|)
then
s := (a+b)/2
set mflag
else
clear mflag
end if
calculate f(s)
d := c
c := b
if f(a)*f(s) < 0 then b := s else a := s
end if
if |f(a)| < |f(b)| then swap (a,b) end if
end repeat
output b or s

2.2 Metode *Bisection*

Algoritma metode *Bisection*.

1. Tentukan nilai xl dan xu untuk mengurung akar, dan pastikan nilai $f(x_l).f(x_u) < 0$, jika lebih besar maka akar tidak terletak antara interval x_l sampai x_u
2. Hitung akar dengan rumus $x_r = \frac{x_l + x_u}{2}$
3. Hitung nilai $f(x_l).f(x_r)$
   a. Jika $f(x_l).f(x_r) < 0$, akar terletak pada sub interval pertama, maka $x_u = x_r$, lanjut ke langkah 4
   b. Jika $f(x_l).f(x_r) > 0$, akar terletak pada sub interval kedua, maka $x_l = x_r$, lanjut ke langkah 4
   c. Jika $f(x_l).f(x_r) = 0$, akar = x_r, komputasi dihentikan

ISSN: 2807-999X

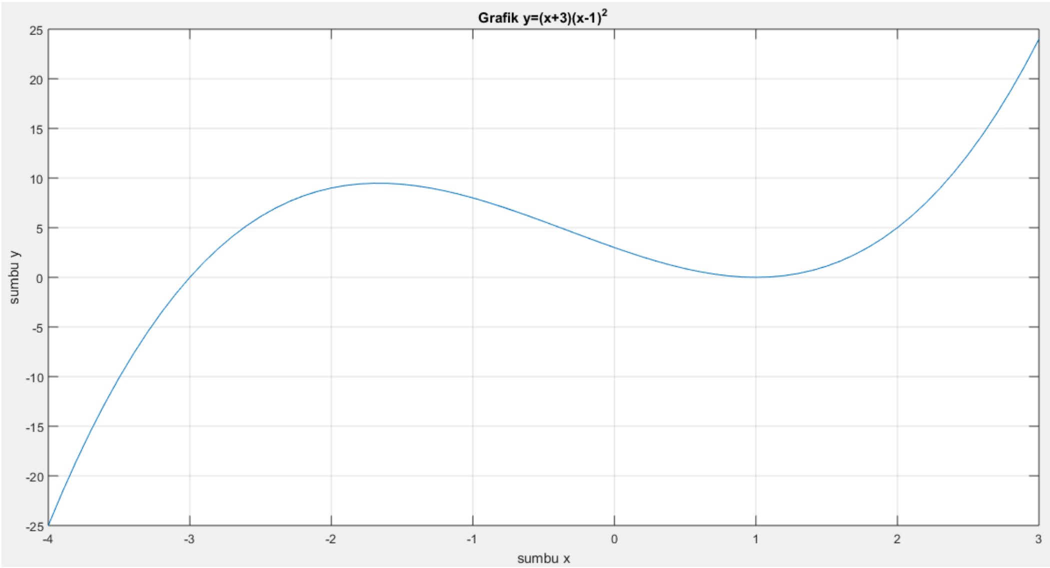
4. Hitung akar baru dengan persamaan $x_r = \frac{x_l + x_u}{2}$
5. Jika akar baru cukup akurat, hentikan komputasi, jika tidak kembali ke langkah 3.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Berikut ini hasil yang diperoleh

1. Untuk persamaan yang memiliki 3 akar dimana 1 akar tunggal dan 2 akar kembar (genap)
   Gambar grafik fungsi $f(x) = (x+3)(x-1)^2$ ditunjukan pada Gambar 1.
   Dari Gambar 1, secara visual bisa dilihat bahwa akar dari persamaan *polynomial* tersebut adalah pada $x = -3$ dan menyinggung sumbu x pada titik $x = 1$ yang merupakan akar kembar. Hasil pencarian akar, jumlah iterasi dan kriteria berhenti dari algoritma pencarian akar untuk metode *Brent* dan *Bisection* ditunjukan pada Tabel 1.



Gambar 1. Grafik fungsi $f(x) = (x+3)(x-1)^2$

Tabel 1. Perbandingan hasil Metode *Brent* dan *Bisection* untuk persamaan $f(x) = (x+3)(x-1)^2$

No	Nilai Awal		Metode Brent			Metode Bisection		
	$a = x_l$	$b = x_u$	Nilai akar = s	Jumlah iterasi	Kriteria berhenti	Nilai akar = x_r	Jumlah iterasi	Kriteria berhenti
1	0,5	3	Tidak dilanjutkan, $f(a) * f(b) > 0$			Tidak dilanjutkan, $f(x_l) * f(x_u) > 0$		
2	-1	2	Tidak dilanjutkan, $f(a) * f(b) > 0$			Tidak dilanjutkan, $f(x_l) * f(x_u) > 0$		
3	-4	0,5	-3,00000	10	$f(s) = 0,0000$	-3,00000	17	$f(x_l) * f(x_r) = 0,00000$ $e_a = 0,00001$
4	-4	-1	-3,00000	6	$f(s) = 0,0000$	-3,00002	16	$f(x_l) * f(x_r)$=0,00000, $e_a = 0,00002$
5	-4	3	-3,00000	8	$f(s) = 0,0000$	-2,99998	17	$f(x_l) * f(x_r)$=0,00000, $e_a = 0,00002$
6	-5	5	-3,00000	9	$f(s) = 0,0000$	-2,99999	18	$f(x_l) * f(x_r)$=0,00000, $e_a = 0,00002$

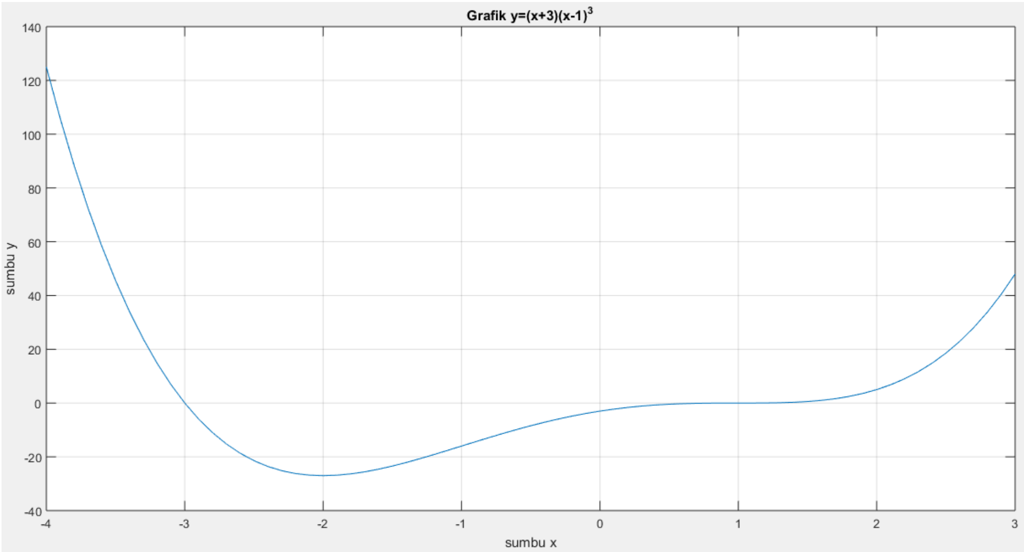
Keterangan:
$a = x_l$: nilai batas bawah
$a = x_u$: nilai batas atas
s : nilai akar yang dicari, dengan metode *Brent*
x_r : nilai akar yang dicari dengan metode *Bisection*

RITEKTRA X BANDUNG 2021
Prosiding Seminar Nasional
Riset dan Teknologi Terapan (RITEKTRA) 2021
Menuju *Society* 5.0: Teknologi Cerdas yang Berpusat pada Manusia
Bandung, 12 Agustus 2021
ISSN: 2807-999X

$f(a)$: nilai fungsi pada titik $x = a$
$f(b)$: nilai fungsi pada titik $x = b$
$f(s)$: nilai fungsi pada titik $x = s$, untuk metode *Brent.*
$f(x_l)$: nilai fungsi pada titik $x = x_l$, untuk metode *Bisection.*
$f(x_r)$: nilai fungsi pada titik $x = r$, untuk metode *Bisection.*
e_a : *error approximation* (kesalahan aproksimasi) dengan perhitungannya menggunakan persamaan

$$e_a = \frac{aproksimasi\_sekatang - aproksimasi\_sebelumnya}{aproksimasi\_sekarang} * 100\%$$

2. Untuk persamaan yang memiliki 4 akar, dimana 1 akar tunggal dan 3 akar kembar. Gambar grafik fungsi $f(x) = (x+3)(x-1)^3$ ditunjukkan pada Gambar 2.



Gambar 2. Grafik fungsi $f(x) = (x+3)(x-1)^3$

Dari Gambar 2 diatas, secara visual terlihat jelas bahwa akar dari persamaan *polynomial* tersebut adalah pada x = -3 dan titik x = 1 yang merupakan akar kembar. Hasil pencarian akar, jumlah iterasi dan kriteria berhenti dari algoritma pencarian akar untuk metode *Brent* dan *Bisection* ditunjukan pada Tabel 2.

Tabel 2. Perbandingan hasil Metode *Brent* dan *Bisection* untuk persamaan $f(x) = (x+3)(x-1)^3$

No	Nilai Awal		Metode *Brent*			Metode *Bisection*		
	$a = x_l$	$b = x_u$	Nilai akar = s	Jumlah iterasi	Kriteria berhenti	Nilai akar = x_r	Jumlah iterasi	Kriteria berhenti
1	0,5	3	0,98968	9	$f(s) = 0{,}0021$	0,96875	4	$f(x_l) * f(x_r) = 0{,}00000$ $e_a = 0{,}00001$
2	-1	2	1,01053	10	$f(s) = 0{,}0000$	0,96875	5	$f(x_l) * f(x_r) = 0{,}00000$ $e_a = 0{,}00001$
3	-4	0,5	-3,00000	10	$f(s) = 0{,}0001$	-3,00000	17	$f(x_l) * f(x_r) = 0{,}00000$ $e_a = 0{,}00001$
4	-4	-1	-3,00000	6	$f(s) = 0{,}0001$	-3,00002	16	$f(x_l) * f(x_r) = 0{,}00000$ $e_a = 0{,}00001$
5	-4	3	Tidak bisa dilanjutkan, $(a) * f(b) > 0$			Tidak bisa dilanjutkan, $(x_l) * f(x_u) > 0$		
6	-5	5	Tidak bisa dilanjutkan, $(a) * f(b) > 0$			Tidak bisa dilanjutkan, $(x_l) * f(x_u) > 0$		

3.2 Pembahasan

Dari hasil 2 tabel diatas, beberapa hal yang perlu dijelaskan bahwa:

1. Pada Tabel 1, untuk persamaan polinomial dengan jumlah akar ganda genap
   a. Metode Brent dan metode *Bisection* sesuai dengan algoritma yang diberikan bahwa jika nilai $f(a) * f(b) > 0$ (metode *Brent*) dan $f(x_l) * f(x_u) > 0$ (metode *Bisection*) maka proses pencarian akar tidak bisa dilanjutkan.
   b. Metode *Brent* dan metode *Bisection* tidak bisa mencari nilai akar ganda (genap) meskipun pemilihan nilai awal mengurung akar ganda tersebut.
   c. Metode *Brent* dan metode *Bisection* mempunyai hasil nilai akar yang dicari jika pemilihan nilai awal mengurung semua akar baik salah satu akar maupun akar ganda dengan hasil akar yang dicari adalah salah satu akar dan bukan akar ganda.
   d. Pada kasus point c diatas, metode *Brent* lebih efektif dengan jumlah iterasi yang sedikit. Metode *Brent* lebih cepat konvergensi dalam mencari akar untuk *polynomial* yang memiliki akar kembar.
2. Tabel 2, untuk persamaan *polynomial* dengan jumlah ganjil
   a. Persamaan polinomial yang memiliki 4 buah akar dan 3 diantaranya akar kembar (ganjil), metode *Brent* dan metode *Bisection* tidak bisa menemukan salah satu akarnya jika pemilihan nilai awalnya mengurung kedua akar tersebut. Sesuai dengan algoritma yang diberikan bahwa jika nilai $f(a) * f(b) > 0$ (metode *Brent*) dan $f(x_l) * f(x_u) > 0$ (metode *Bisection*) maka proses pencarian akar tidak bisa dilanjutkan. Pada persamaan polinomial, nilai akarnya adalah -3 dan 1 (akar kembar) dan nilai awal yang dipilah -4 dan 3 serta -5 dan 5.
   b. Metode *Brent* dan metode *Bisection* akan mendapatkan akar yang dicari untuk polinomial yang memiliki akar kembar jika pemilihan nilai awal kedua metode tersebut mengurung salah satu akar.
   c. Metode *Brent* dan metode *Bisection* tidak bisa mencari nilai akar ganda (genap) meskipun pemilihan nilai awal mengurung akar ganda tersebut.
   d. Pada kasus point c diatas, pencarian akar polinomial yang memiliki 4 akar dan 3 diantaranya adalah akar ganda (ganjil), metode *Brent* lebih cepat konvergensi dengan jumlah iterasi yang sedikit untuk pencarian akar ganda. Sementara salah satu akar lain metode *Bisection* lebih cepat konvergensi dalam mencari akar.

4. KESIMPULAN

Beberapa kesimpulan yang diambil dari penelitian ini:

1. Persamaan berbentuk polinomial yang memiliki akar ganda dengan jumlah genap, baik metode *Brent* maupun metode *Bisection* tidak bisa menemukan nilai akar ganda (genap) meskipun pemilihan nilai awal mengurung akar ganda tersebut. Berbeda dengan persamaan polinomial yang memiliki akar ganda dengan jumlah ganjil, metode *Brent* dan metode *Bisection* bisa menemukan akar tersebut.
2. Pemilihan nilai tebakan awal dengan mengurung semua akar yaitu akar ganda dan salah satu akar pada persamaan tersebut yang berbentuk polinomial, baik metode *Bisection* maupun metode *Brent,* hasil pencarian akarnya adalah akar ganda. Hal ini berlaku pada persamaan polinomial yang jumlah akar gandanya genap. Tetapi untuk persamaan polinomial yang jumlah akar gandanya ganjil hasil pencarian akarnya tidak bisa ditemukan karena sesuai algoritma pencarian kedua metode tersebut dibatasi jika nilai $f(a) * f(b) > 0$ (metode Brent) dan $f(x_l) * f(x_u) > 0$ (metode Bisection), maka pencarian akar tidak dilanjutkan.

5. SARAN DAN PENGEMBANGAN

Dalam penentuan akar ganda suatu persamaan berbentuk *polynomial*, *hybrid* metode *Brent* dengan metode lainnya perlu dikembangkan. Berbagai metode bisa ada kemungkinan dikombinasi dalam pencarian akar persamaan. Selanjutnya, percobaan tidak sebatas pada persamaan berbentuk *polynomial* tetapi juga pada bentuk persamaan lainnya seperti eksponensial, sinusoidal dan kombinasi berbagai bentuk persamaan lainnya.

ISSN: 2807-999X

PUSTAKA

Batarius, P. & SinLae, A.A. 2019. Nilai Awal Pada Metode Secant Yang Dimodifikasi Dalam Penentuan Akar Ganda Persamaan Non Linear. *Jurnal Ilmiah MATRIK,* 21 (1).

Chapra, C.S. & Canale, P.R. 2010. *Numerical Methods for Engineers*, Sixth Edition, hlm. 164-167. McGraw Hill Companies Inc.

Dehghani, R., Bidabadi, N., Fahs H., & Hosseini, M.M. 2019. A Conjugate Gradient Method Based on a Modified Secant Relation for Unconstrained Optimization, Numerical Functional Analysis And Optimization, (https://doi.org/10.1080/01630563.2019.1669641).

Etin-Osa, B. F. & Noah Ch.T. 2020. On A One Fixed Point Improved Secant Method for Solving Roots of Polynomials. *Earthline Journal of Mathematical Sciences*, 3 (1): 83-93.

Kim, J., Noh, T., Oh, W., Park S. 2017. An Improved Hybrid Algorithm to Bisection Method and Newton-Raphson. *HIKARI, Method Applied Mathematical Sciences*, 11 (56): 2789 – 2797.

Pinkham, S. & Sansiribhan, S. 2020. A New Hybrid Algorithm of Bisection and Modified Newton's Method for The Nth Root-finding of A Real Number. *Journal of Physics*: Conference Series 1593.

Rahman, N. Ab., Abdullah, L., Ghani, A.T.Ab. 2017. Modified Brent's Method for Solving Interval Type-2 Fuzzy Polynomials. *Far East Journal of Mathematical Sciences*, 102 (6): 1099-1113.

Sharma, J.R. & Kumar, S. 2021. *Excellent Numerical Technique for Multiple Roots, Mathematics and Computers in Simulation*, 182: 316–324.

Temelcan, G., Sivri, M., Albayrak, I. 2020. A New Iterative Linearization Approach for Solving Nonlinear Equations Systems. *An International Journal of Optimization and Control: Theories & Applications*, 10 (1): 47-54.

Vakkalagadda, S.S.P. 2020. A Better Root Finding Method using False Position and Inverse Quadratic Interpolation Methods. *International Research Journal of Engineering and Technology (IRJET)*, 07 (04): 4178-4180.

Waziri, M.Y., Ahmed, K., Sabi'u, J. 2020. A Dai–Liao Conjugate Gradient Method Via Modified Secant Equation for System of Nonlinear Equations. *Arabian Journal of Mathematics, Arab. J. Math*, 9: 443–457.