

PERBANDINGAN METODE BISEKSI, REGULA FALSI, SECANT, DAN NEWTON-RAPHSON DALAM PENYELESAIAN PERSAMAAN TRANSENDEN MENGGUNAKAN PYTHON

I Gede Eka Agung Agastya Punia^{1,2,3,*}, Krishna Prafidya Romantica¹

¹Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Terbuka, Tangerang Selatan

²Program Studi Doktor Ilmu Kedokteran, Fakultas Kedokteran, Universitas Indonesia, Depok

³Program Studi Sarjana Kedokteran, Fakultas Kedokteran, Universitas Mahasaraswati Denpasar, Denpasar

*Penulis korespondensi: agastya@unmas.ac.id

ABSTRAK

Fungsi transendental salah satu fungsi analitis yang tidak dapat ditentukan menggunakan persamaan polinomial secara rutin. Dengan demikian, fungsi ini lebih kompleks dibandingkan dengan fungsi aljabar. Karena itu, fungsi ini juga dikenal sebagai fungsi yang 'melampaui' aljabar sehingga tidak dapat direpresentasikan menggunakan operasi penjumlahan, pengurangan, perkalian, pembagian, serta operasi eksponen dan akar dalam jumlah tertentu. Beberapa fungsi transendental yang sering digunakan adalah fungsi eksponensial, fungsi logaritma, dan fungsi trigonometri. Penelitian ini berfokus pada analisis dan perbandingan efektivitas empat metode dalam menyelesaikan persamaan transendental yang rumit, yakni $x^8 - 40x^3 + e^{-x} + 10 = 0$ pada interval $[0, 2]$. Empat metode yang dimaksud meliputi Metode Biseksi, Regula Falsi, Secant, dan Newton-Raphson. Untuk proses ini, peneliti menggunakan bahasa pemrograman Python di platform Google Collab. Hasil hitungan diverifikasi menggunakan Claude™. Pada akhirnya, keempat metode tersebut berhasil menghasilkan nilai akar yang sama, yaitu $x \approx 0.641409591970798502$. Meski demikian, jumlah iterasi dan efisiensi dari setiap metode masih bervariasi. Metode yang paling efektif dalam penelitian ini adalah Metode Newton-Raphson dengan hanya memerlukan 8 iterasi. Dari sisi kestabilan, Metode Newton-Raphson dan Secant dikategorikan stabil karena memiliki konvergensi superlinear hingga kuadratik. Ini berarti bahwa kedua metode tersebut dapat diandalkan dalam menemukan solusi persamaan transendental dengan cepat dan akurat.

Kata kunci: Persamaan Transenden, Akar Persamaan, Metode Numerik, Python.

1. Pendahuluan

Fungsi transendental salah satu fungsi analitis yang tidak dapat ditentukan menggunakan persamaan polinomial secara rutin. Fungsi ini lebih kompleks dibandingkan dengan fungsi aljabar. Karena itu, fungsi ini juga dikenal sebagai fungsi yang 'melampaui' aljabar sehingga tidak dapat direpresentasikan menggunakan operasi penjumlahan, pengurangan, perkalian, pembagian, serta operasi eksponen dan akar dalam jumlah tertentu. Beberapa fungsi transendental yang sering digunakan adalah fungsi eksponensial, fungsi logaritma, dan fungsi trigonometri (Widyanto, 2022).

Dalam metode numerik, Pencarian akar $f(x) = 0$ dapat dilakukan secara iteratif. Metode pencarian akar dapat dilakukan dengan dua cara yakni metode braket dan metode terbuka. Biseksi dan Regula Falsi adalah metode tertutup, yakni metode yang memerlukan dua variabel angka dalam penentuan akar dan hasil perhitungan akar yang diperoleh terletak diantara kedua variabel tersebut (Darmanto, 2022). Metode Biseksi adalah metode yang paling gampang dan sederhana

daripada metode lainnya dengan prinsip membagi interval menjadi 2 bagian secara terus-menerus dan dapat diperoleh nilai hampiran yang memenuhi toleransi tertentu (Putri dkk, 2026). Metode regula falsi merupakan metode yang menyerupai metode biseksi yang akan melakukan iterasi dilakukan dengan terus menerus memperbarui rentang untuk memperoleh akan persamaan (Islam, 2022). Namun pencarian akarnya didasarkan pada kemiringan dan selisih tinggi dari dua titik batas (Rosidi, 2019).

Selain dari metode Biseksi dan Regula Falsi terdapat metode Secant dan Newton Rhapson. Metode Secant merupakan sebuah metode yang dikembangkan oleh Newton-Rhapson yang tidak memerlukan sebuah turunan eksplisit dari sebuah fungsi. Teknik Secant menggunakan 2 tebakan awal untuk membuat sebuah garis secant dalam pendekatan sesuai dengan fungsi gradien. Rumusnya adalah sebagai berikut (Ginantara dkk., 2021) dan Metode Newton-Raphson merupakan sebuah metode numerik iteratif yang digunakan untuk menemukan akar persamaan non-linear dengan laju konvergensi yang cepat. Metode ini bekerja dengan memanfaatkan turunan pertama suatu fungsi untuk mengoreksi estimasi akar pada setiap iterasi (Badr dkk., 2021).

Keempat metode ini digunakan karena masing-masing memiliki keunggulan dan karakteristik yang berbeda sehingga dapat memberikan pemahaman yang komprehensif mengenai pendekatan pencarian akar persamaan non-linear. Metode Biseksi dan Regula Falsi digunakan karena sifatnya yang terpercaya dan konvergen selama terdapat perubahan tanda fungsi pada interval awal, meskipun konvergensi relatif lambat. Metode Secant dan Newton-Raphson dipilih karena memiliki laju konvergensi yang jauh lebih cepat dan lebih efisien untuk permasalahan yang membutuhkan akurasi tinggi dengan iterasi yang sedikit, akan tetapi memerlukan tebakan awal yang cukup baik dan informasi turunan fungsi. Penggunaan keempat metode ini memungkinkan perbandingan antara kestabilan, kecepatan, dan kemudahan implementasi dalam menyelesaikan berbagai jenis persamaan non-linear. Terdapat perbedaan signifikan dalam beberapa kali iterasi pada dalam sebuah persamaan yang sama (Bhardwaj, R. 2025). Metode biseksi umumnya menggunakan iterasi lebih banyak dibandingkan regula falsi untuk menemukan akar persamaan yang sama (Wigati J., 2020). Dalam makalah ini perbandingan antara metode biseksi dan regula falsi dilakukan pada persamaan transenden yakni $x^8 - 40x^3 + e^{-x} + 10 = 0$. karena persamaan tersebut merupakan sebuah persamaan transenden yang memuat bilangan euler (Kurudirek, A., 2026). Kajian ini memiliki tujuan untuk menganalisis keakuratan hasil, jumlah iterasi, dan galat dari masing-masing metode dalam menentukan akar persamaan tersebut serta menemukan metode yang paling cepat dan memiliki galat yang paling rendah.

2. METODE

Penelitian ini merupakan penelitian yang menggunakan studi komparatif numerik, yakni membandingkan empat metode penyelesaian akar persamaan transenden secara iteratif terhadap fungsi yang sama. Jenis penelitian adalah metode terapan, yakni metode yang mengaplikasikan pengetahuan numerik dalam menyelesaikan masalah nyata. Objek kajian adalah persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$ yang merupakan persamaan polinomial berderajat tiga dengan akar yang tidak dapat ditentukan secara analitik dan langsung (Ismuniyarto, 2016) dengan interval $[0,2]$. Penelitian akan dilaksanakan melalui beberapa tahap yang meliputi identifikasi fungsi. Setelah itu menghitung nilai galat absolut dengan rumus,

$$\varepsilon = |x - x^*| \quad (1)$$

di mana:

x = Nilai Absolut

x^* = Nilai aproksimasi

dan stabilitas metode diukur menggunakan kriteria konvergensi iterasi dengan rumus,

$$\varepsilon = |x_{n+1} - x_n| < \delta \quad (2)$$

di mana:

x_n = nilai aproksimasi pada iterasi ke-n,

x_{n+1} = nilai aproksimasi pada iterasi ke-(n+1),

δ = batas toleransi

Selain itu penelitian ini juga menghitung berdasarkan jumlah maksimum iterasi sebagai kriteria pemberhentian proses iterasi (Yi, dkk, 2020). untuk mengurangi kesalahan, penelitian ini akan menggunakan bahasa pemrograman Python dalam implementasi metode numerik untuk mendapatkan hasil perhitungan dalam bentuk tabel dan nilai hampiran akar persamaan secara otomatis dan terstruktur (Widiyanto dkk, 2022). Untuk melakukan validasi hasil perhitungan python yang dilakukan di Google Collab, penelitian ini akan menggunakan ClaudeTM dalam melakukan perhitungan validasi sehingga meminimalisasi kesalahan.

3. HASIL DAN PEMBAHASAN

3.1 Metode Biseksi

Prinsip dasar dari metode biseksi adalah mengurung akar fungsi dalam interval $[a,b]$, lalu interval tersebut terus menerus dibagi dua hingga sekecil mungkin sehingga nilai hampiran yang dicari dapat ditentukan dengan tingkat toleransi tertentu. Pada setiap titik iterasi, titik Tengah dapat dihitung dengan rumus

$$x_{tengah} = \frac{x_n + x_{n+1}}{2} \quad (3)$$

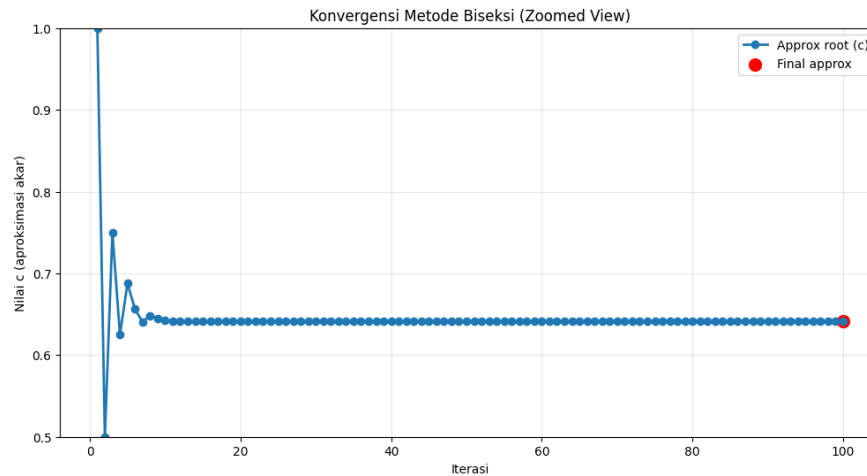
Lalu dilakukan pengecekan tanda apabila $f(x_{tengah}) \cdot f(x_n) < 0$ maka akar berada di subinterval kiri sehingga batas atas harus diperbaharui. Apabila yang terjadi sebaliknya maka batas bawah yang diperbarui. Proses ini diulang hingga mencapai kriteria toleransi (Ersöz,2013). Pada persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$, Kita dapat menggunakan python untuk perbantuan yang hasilnya dapat dilihat di tabel berikut

Tabel 1. Hasil aproksimasi dengan metode biseksi

Iter	a	b	c	$f(c)$	Galat	Konvergensi	Kestabilan
1	0	2	1	-28.6321	0	1	STABIL
2	0	1	0.5	5.61044	0.5	0.5	TIDAK STABIL
3	0.5	1	0.75	-6.30252	0.25	0.25	TIDAK STABIL
4	0.5	0.75	0.625	0.792919	0.125	0.125	TIDAK STABIL
5	0.625	0.75	0.6875	-2.44531	0.0625	0.0625	TIDAK STABIL
...	...	...	...	...	...	...	...
98	0.64141	0.64141	0.64141	-1.98883 $\times 10^{-28}$	6.31089 $\times 10^{-30}$	6.31089 x 10^{-30}	TIDAK STABIL
99	0.64141	0.64141	0.64141	-4.25688 $\times 10^{-29}$	3.15544 $\times 10^{-30}$	3.15544 x 10^{-30}	TIDAK STABIL
100	0.64141	0.64141	0.64141	-3.55884 $\times 10^{-29}$	1.57772 $\times 10^{-30}$	1.57772 x 10^{-30}	TIDAK STABIL

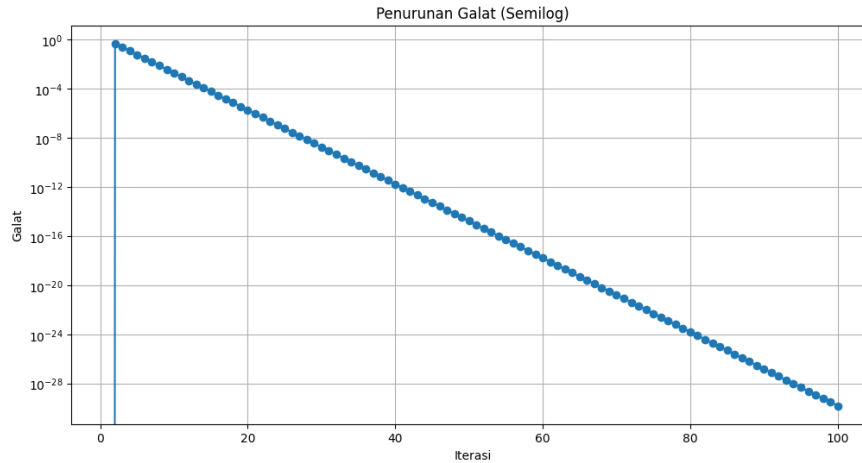
Setelah dilakukan proses iterasi sebanyak 100 kali menggunakan metode biseksi, diperoleh akar aproksimasi dari persamaan yang ditinjau, yakni $x \approx 0.641409591970798502071634345978$. Selama proses 100 iterasi, metode menunjukkan status tidak stabil secara konsisten sejak iterasi ke-2. Ini terlihat dari nilai $f(c)$ yang berfluktuasi ekstrem bahkan pada iterasi ke-98 dan ke-99, nilai $f(c)$ meledak hingga orde 10^{-28} sebelum tiba-

tiba turun kembali di iterasi ke-100. Ketidakstabilan ini bukan berarti akar yang ditemukan salah, melainkan menunjukkan bahwa perilaku fungsi di sekitar akar sangat tidak linear atau bermasalah secara numerik bisa disebabkan oleh fungsi yang memiliki turunan sangat besar, atau akumulasi kesalahan pembulatan dalam komputasi. Galat akhir yang diperoleh adalah $\varepsilon = 1.57772 \times 10^{-30}$. Artinya secara numerik interval pencarian sudah menyempit hingga hampir nol dan nilai a , b , serta c hampir identik (0.64141). Meskipun galat akhirnya sangat kecil, status "tidak stabil" tetap dicatatkan karena selama proses iterasi tidak ada pola penurunan galat yang konsisten dan monoton. Nilai ini merupakan titik di mana fungsi $f(x)$ mendekati nol, artinya x tersebut adalah solusi numerik dari persamaan yang diselesaikan.



Gambar 1. Hasil Konvergensi dengan metode biseksi

Gambar 1 menampilkan perjalanan nilai aproksimasi akar c dari iterasi pertama hingga ke-100. Fase awal pada iterasi 1-10 terlihat nilai c bergerak sangat dramatis yang dimulai dari 1.0, lalu turun ke 0.5, naik ke 0.75, turun ke 0.625, naik lagi ke 0.6875, dan seterusnya. Ini adalah perilaku normal metode biseksi bahwa setiap langkah membelah interval menjadi dua, sehingga nilai c seakan "memantul" bolak-balik di sekitar akar sejati sambil terus menyempit. Setelah sekitar iterasi ke-10, nilai c terlihat datar dan tidak berubah di angka 0.6414. Ini bukan karena komputasi berhenti, melainkan karena perubahan antar iterasi sudah sangat kecil sehingga tidak terlihat dalam skala grafik biasa. Titik merah menandai akar akhir yang ditemukan di iterasi ke-100, tepat di $x \approx 0.6414$ yang mengonfirmasi bahwa nilai c telah konvergen ke titik tersebut. Grafik ini membuktikan bahwa metode biseksi berhasil menemukan akar meskipun dalam tabel iterasi tercatat "tidak stabil". Ketidakstabilan yang dimaksud dalam tabel lebih mengacu pada perilaku nilai $f(c)$ yang berfluktuasi.



Gambar 2. Penurunan galat dengan metode biseksi

Pada Gambar 2, terlihat dari iterasi ke-3 hingga ke-100, titik-titik galat membentuk garis lurus yang hampir sempurna. Setiap iterasi mengurangi galat dengan faktor konstan karena dalam hal ini faktor $\frac{1}{2}$ karena interval selalu dibelah dua. Pada penurunan galat, Terlihat garis vertikal tegak di awal grafik. Ini mencerminkan langkah pertama di mana galat langsung turun dari nilai besar ke 0.5 secara mendadak sebelum kemudian menurun secara teratur. Penurunan galat yang monoton, teratur, dan linier dalam skala log. Nilai galat akhir $\approx 1.578 \times 10^{-30}$

3.2 Metode Regula Falsi

Metode Regula Falsi adalah sebuah komodifikasi dari Metode Biseksi yang kinerjanya lebih cepat dalam mencapai akar aproksimasi. Metode ini bekerja menjadi 2 bagian yang sama. Metode ini dikenal juga dengan metode interpolasi linier yakni metode yang digunakan untuk mencari akar-akar persamaan non-linier melalui proses iterasi berdasarkan kemiringan jenis yang menghubungkan 2 titik batas pada kurva. Rumus titik aproksimasinya adalah,

$$x = b - \frac{f(b)(b-a)}{f(b)-f(a)} \quad (4)$$

di mana:

a dan b adalah dua titik batas interval awal yang memuat akar $f(a)$ dan $f(b)$

$f(a)$ = Nilai fungsi pada titik a .

$f(b)$ = Nilai fungsi pada titik b .

$(b - a)$ = Panjang interval saat ini.

$f(b) - f(a)$ = Selisih nilai fungsi di kedua titik batas

x : Nilai aproksimasi akar yang baru

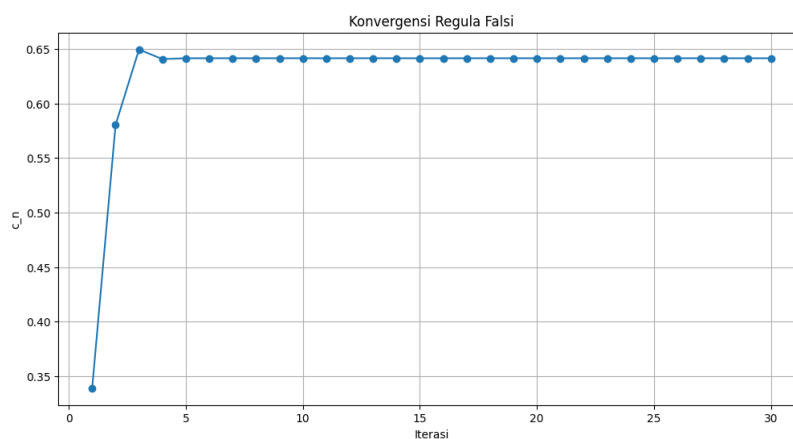
Kriteria penghentian iterasi pada metode regula falsi apabila $|f(x_n)| \leq \varepsilon$ yang berarti nilai fungsi pada titik hampiran sudah sangat mendekati nol sesuai dengan toleransi yang ditetapkan (Bandiyah, S.N.,2024) Pada persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$, Kita dapat menggunakan python untuk perbantuan yang hasilnya dapat dilihat di tabel berikut.

Tabel 2. Hasil aproksimasi dengan metode Regula Falsi

Iter	a	b	x	$f(x)$	Galat	Stabilitas
1	0	2	0.339168	9.1519	9.1519	STABIL
2	0.339168	2	0.58037	2.75312	0.241203	STABIL

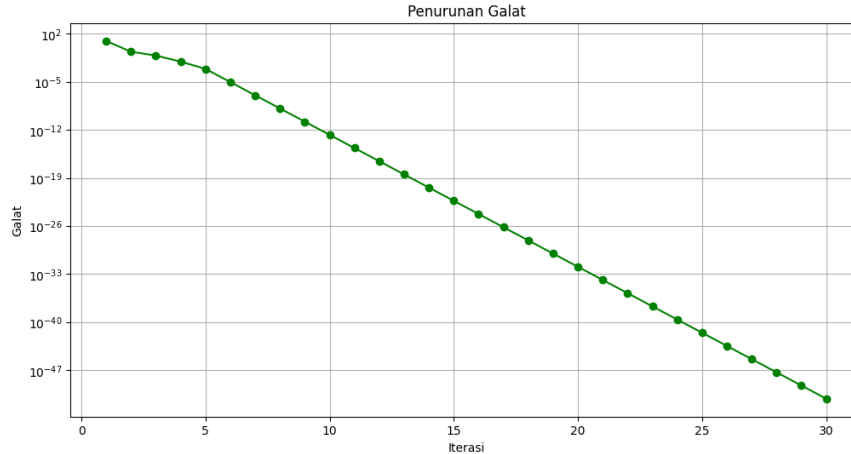
Iter	a	b	x	$f(x)$	Galat	Stabilitas
3	0.58037	2	0.649402	-0.400717	0.0690315	STABIL
4	0.58037	0.649402	0.640631	0.0385248	0.00877093	STABIL
5	0.640631	0.649402	0.6414	0.000461664	0.000769276	STABIL
...	...	...	...	...	...	...
28	0.64141	0.649402	0.64141	2.83199×10^{-48}	4.72279×10^{-48}	STABIL
29	0.64141	0.649402	0.64141	3.38704×10^{-50}	5.64843×10^{-50}	STABIL
30	0.64141	0.649402	0.64141	4.05087×10^{-52}	6.75548×10^{-52}	STABIL

Ditemukan akar aproksimasi $\approx 0.641409591970798502$ dalam iterasi sebanyak 30 dan Galat akhir: 6.75548×10^{-52} Kestabilan Stabil. Pada iterasi ke-1, titik awal $x = 0.339168$ masih jauh dari akar $f(c)$ masih bernilai besar. Namun hanya dalam 5 iterasi, nilai x sudah mencapai 0.6414 dengan galat di bawah 0.001. Ini menunjukkan konvergensi yang sangat agresif di fase awal. Setelah iterasi ke-5, nilai c sudah sangat dekat dengan akar. Iterasi selanjutnya secara bertahap menyempurnakan presisi hingga orde yang sangat kecil. Batas atas $b = 0.649402$ tidak pernah berubah sejak iterasi ke-3. Pada iterasi ke 28-30, Galat turun sekitar faktor ~ 60 per iterasi di fase akhir ini.



Gambar 3. Hasil Konvergensi dengan metode Regula-Falsi

Pada Gambar 3 kita lihat bahwa grafik ini menampilkan nilai aproksimasi akar c dari iterasi 1 hingga 30. Kurva berbentuk seperti huruf "J" terbalik yang terlihat naik cepat di awal lalu mendatar sempurna. Pada iterasi 1-4, Nilai c dimulai sangat rendah di ≈ 0.339 pada iterasi pertama, lalu melompat tajam ke ≈ 0.580 (iterasi 2), kemudian naik lagi ke ≈ 0.649 di iterasi 3-4. Pada awal iterasi, garis interpolasi masih jauh dari akar sehingga koreksinya besar. Mulai iterasi ke-4, nilai c langsung datar di sekitar 0.6414 dan tidak bergeser lagi secara visual hingga iterasi ke-30. Ini menunjukkan bahwa metode sudah "mengunci" akar dengan sangat cepat. Seluruh iterasi setelah ke-4 hanyalah penyempurnaan presisi di digit-digit desimal yang sangat dalam yang tidak terlihat di skala grafik ini. Perlu diperhatikan bahwa kurva hanya naik dari bawah menuju akar, tidak pernah melampaui nilai ≈ 0.649 dan tidak bolak-balik seperti biseksi yang menunjukkan konvergensi satu sisi.



Gambar 4. Penurunan galat dengan metode Regula-Falsi

Pada Gambar 4 kita dapat melihat penurunan galat menggunakan metode Regula-Falsi. Kita melihat bahwa penurunan membentuk sebuah garis lurus pada grafik ini berarti penurunan galat yang eksponensial dan konsisten. Dari iterasi ke-1 hingga ke-30, titik-titik galat membentuk garis lurus menurun yang sangat rapi. Kemiringan grafik terlihat lebih curam karena mencapai 54 orde hanya dalam 30 iterasi yang artinya rata-rata setiap iterasi mengurangi galat jauh lebih efisien dari biseksi yang hanya faktor 2 per iterasi.

Berbeda dengan grafik galat biseksi yang sempat menunjukkan lonjakan besar di iterasi akhir, grafik Regula Falsi ini mulus sempurna dari awal hingga akhir. tidak ada fluktuasi, tidak ada lonjakan yang menunjukkan metode berjalan stabil.

3.3 Metode Secant

Metode Secant merupakan sebuah metode yang dikembangkan oleh Newton-Rhapson yang tidak memerlukan sebuah turunan eksplisit dari sebuah fungsi. Teknik Secant menggunakan 2 tebakan awal untuk membuat sebuah garis secant dalam pendekatan sesuai dengan fungsi gradien. Rumusnya adalah sebagai berikut (Ginantara dkk, 2021),

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (5)$$

dengan:

x_{n-1} = Aproksimasi akar dari iterasi sebelumnya

x_n = Aproksimasi akar sekarang

x_{n+1} = Aproksimasi akar baru setelah iterasi secant

Metode secant akan berhenti ketika perubahan nilai akar antara dua iterasi berturut-turut sangat kecil dan lebih kecil dari toleransi yang ditentukan, dirumuskan sebagai $|x_n - x_{n-1}| < \varepsilon$ atau nilai fungsi pada taksiran akar mendekati nol atau berada di bawah nilai toleransi error, yaitu $|f(x_n)| < \varepsilon$ (Ginantara dkk, 2021).

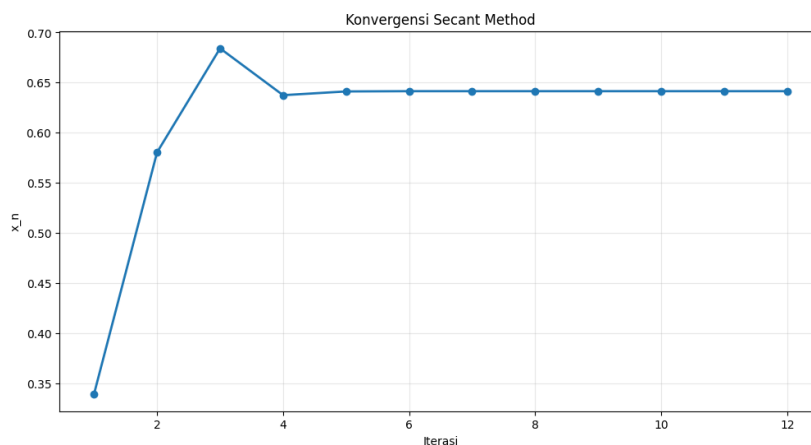
Pada persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$, Kita dapat menggunakan python untuk perbantuan yang hasilnya dapat dilihat di tabel berikut.

Tabel 3. Hasil aproksimasi metode Secant

Iter	x_{n-1}	x_n	x_{n+1}	$f(x)$	Galat	Stabilitas
1	0	2	0.339168	-53.8647	1.66083	STABIL
2	0.339168	0.58037	9.1519	0.241203	-	STABIL

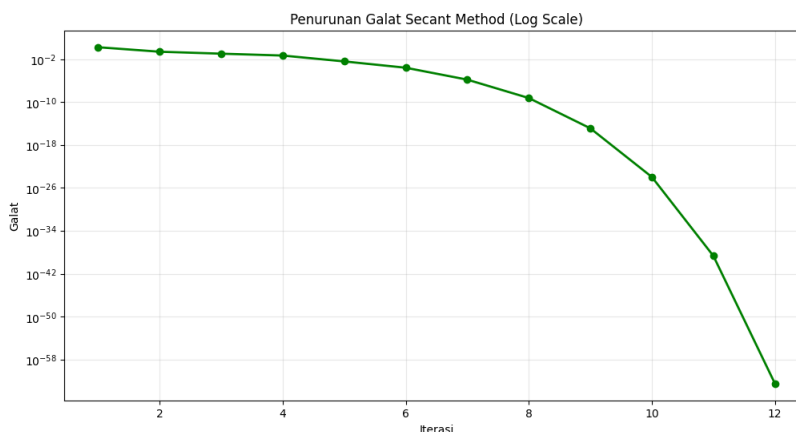
Iter	x_{n-1}	x_n	x_{n+1}	$f(x)$	Galat	Stabilitas
3	0.339168	0.58037	0.68415	2.75312	0.103779	STABIL
4	0.58037	0.68415	0.637405	-2.25643	0.0467448	STABIL
5	0.68415	0.637405	0.641162	0.197191	0.00375676	STABIL
6	0.637405	0.641162	0.641411	0.0122808	0.000249505	STABIL
7	0.641162	0.641411	0.64141	-7.45804×10^{-5}	1.50608×10^{-6}	STABIL
8	0.641411	0.64141	0.64141	2.79188×10^{-8}	5.63582×10^{-10}	STABIL
9	0.64141	0.64141	0.64141	6.34292×10^{-14}	1.28042×10^{-15}	STABIL
10	0.64141	0.64141	0.64141	-5.39456×10^{-23}	1.08897×10^{-24}	STABIL
11	0.64141	0.64141	0.64141	1.04235×10^{-37}	2.10415×10^{-39}	STABIL
12	0.64141	0.64141	0.64141	1.71294×10^{-61}	3.45783×10^{-63}	STABIL

Ditemukan akar aproksimasi $\approx 0.641409591970798502$ dalam iterasi sebanyak 12 dan Galat akhir: 3.45783×10^{-63} Kestabilan stabil. Metode Secant hanya membutuhkan 12 iterasi untuk mencapai presisi yang sangat tinggi. Ini karena berdasarkan rumus metode Secant menggunakan dua titik sekaligus untuk memperkirakan turunan fungsi secara numerik, sehingga laju konvergensinya bersifat superlinear dengan orde ≈ 1.618 . Galat akhir sebesar 3.45783×10^{-63} adalah nilai yang sangat kecil yang jauh melampaui standar toleransi rekayasa maupun sains. Ini berarti akar yang ditemukan memiliki presisi hingga 63-digit desimal, menjadikannya hasil yang sangat akurat secara numerik. Metode Secant menunjukkan kestabilan yang konsisten dan baik. Hal ini terlihat dari pola galat yang menurun secara teratur di setiap iterasi yang dimulai dari 1.66 (iterasi 1) hingga 10^{-63} (iterasi 12) tanpa menunjukkan fluktuasi atau lonjakan yang tidak wajar.



Gambar 5. Hasil konvergensi metode Secant

Pada grafik konvergensi di Gambar 5 terlihat bahwa grafik menampilkan nilai aproksimasi x dari iterasi 1 hingga 12. Kurva berbentuk "busur naik lalu datar" namun dengan satu keunikan bahwa adanya overshoot di iterasi ke-3. overshoot ini disebabkan karena metode Secant menggunakan dua titik untuk memperkirakan turunan, teknik ini kadang "melompat terlalu jauh" sebelum terkoreksi. Setelah iterasi ke-4, nilai x langsung datar sempurna di ≈ 0.6414 tanpa guncangan lebih lanjut. Seluruh 8 iterasi berikutnya hanya menyempurnakan digit-digit presisi yang tidak terlihat di skala grafik ini.



Gambar 6. Hasil penurunan galat metode Secant

Perhatikan bahwa pada Gambar 6 pola kurva melengkung ke bawah. Ini disebabkan karena teknik konvergensi secant yang bersifat konvergensi superlinear, di mana laju penurunan galat semakin mempercepat dirinya sendiri setiap iterasi. Dari grafik kita dapat lihat fase lambat yakni fase iterasi 1-7 yang terlihat galat turun dari $\sim 10^{-2}$ (iterasi 1) hingga $\sim 10^{-9}$ (iterasi 7) yakni penurunan sekitar 7 orde dalam 7 langkah. Terlihat relatif landai di bagian kiri grafik. Setelah itu masuk fase akselerasi. Dalam 5 iterasi terakhir saja, galat turun 53 orde besaran nyaris sepanjang yang dicapai Regula Falsi dalam 30 iterasi penuh. Inilah bukti nyata konvergensi superlinear orde ≈ 1.618 . Kurva turun mulus tanpa fluktuasi dari awal hingga akhir yang mengonfirmasi kestabilan numerik yang baik sepanjang proses iterasi.

3.4 Metode Newton-Rhapson

Metode Newton-Raphson merupakan sebuah metode numerik iteratif yang digunakan untuk menemukan akar persamaan non-linear dengan laju konvergensi yang cepat. Metode ini bekerja dengan memanfaatkan turunan pertama suatu fungsi untuk mengoreksi estimasi akar pada setiap iterasi (Badr dkk., 2021). Akan tetapi, metode ini memiliki keterbatasan karena memerlukan perhitungan turunan fungsi yang tidak selalu mudah diperoleh, serta sensitivitas terhadap pemilihan tebakan awal (Chen dkk., 2021). Rumusnya adalah sebagai berikut,

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

dengan deskripsi:

x_n = Perkiraan nilai akar (titik optimal) pada iterasi ke-n

x_{n+1} = Estimasi baru setelah koreksi rumus Newton

$f(x)$ = Fungsi non-linear yang berasal dari model logistik.

$f'(x)$ = Turunan dari $(f(x))$

Metode Newton-Rhapson dapat dihentikan ketika metode divergen atau konvergensinya sangat lambat. Pada persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$, Kita dapat menggunakan python untuk perbantuan yang hasilnya dapat dilihat di tabel berikut.

Tabel 4. Hasil aproksimasi metode Newton-Rhapson

Iter	x	$f(x)$	$f'(x)$	x_{new}	Galat	Stabilitas
1	5.0	5.610437	-3.054403×10^1	0.6836836	1.836836×10^{-1}	STABIL
2	0.6836836	-2.230294	-5.603697×10^1	0.6438832	3.980040×10^{-2}	STABIL

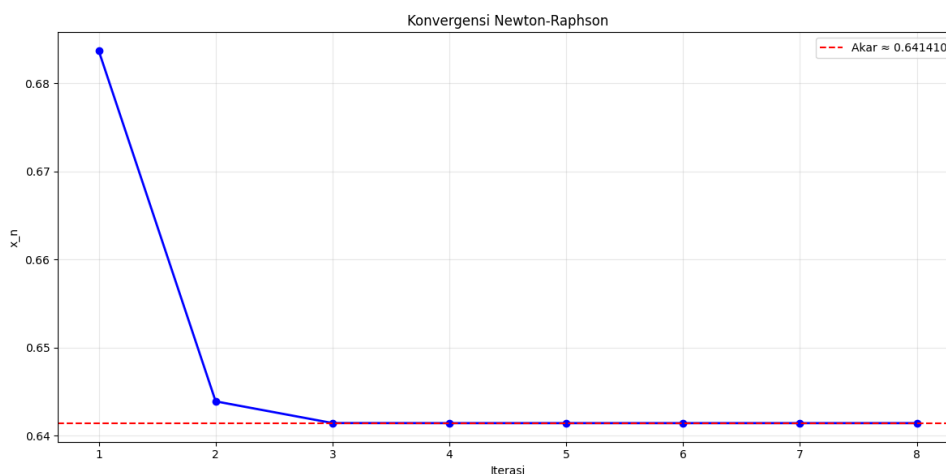
Iter	x	$f(x)$	$f'(x)$	x_{new}	Galat	Stabilitas
3	0.6438832	-0.1229946	-4.990845×10^1	0.6414188	2.464403×10^{-3}	STABIL
4	0.6414188	-4.550378×10^{-4}	-4.953937×10^1	0.6414096	9.185376×10^{-6}	STABIL
5	0.6414096	-6.307315×10^{-9}	-4.953800×10^1	0.6414096	1.273228×10^{-10}	STABIL
6	0.6414096	$-1.211879 \times 10^{-18}$	-4.953800×10^1	0.6414096	2.446362×10^{-20}	STABIL
7	0.6414096	$-4.473923 \times 10^{-38}$	-4.953800×10^1	0.6414096	9.031296×10^{-40}	STABIL
8	0.6414096	$-6.097435 \times 10^{-77}$	-4.953800×10^1	0.6414096	1.230860×10^{-78}	STABIL

Metode Newton-Raphson diterapkan untuk menyelesaikan persamaan nonlinear $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$ dengan titik awal yang ditetapkan pada $x_0 = 5$, yang merupakan titik cukup jauh dari akar sejati. Meskipun demikian, metode ini berhasil menemukan akar aproksimasi pada nilai $x \approx 0.6414096$ hanya dalam 8 iterasi, dengan galat akhir yang sangat kecil yakni 1.231×10^{-78} .

Pada iterasi pertama, meskipun titik awal berada jauh dari akar, metode Newton-Raphson langsung memberikan koreksi besar dengan melompat dari $x = 5$ ke $x \approx 0.6837$. Hal ini dimungkinkan karena metode ini menggunakan informasi turunan $f'(x)$ untuk mengarahkan pencarian secara lebih tepat dan efisien. Memasuki iterasi kedua dan ketiga, nilai x terus menyempurnakan diri mendekati 0.6414 dengan galat yang turun dari 3.98×10^{-2} menjadi 2.46×10^{-3} .

Keunggulan metode ini mulai terlihat sangat jelas pada iterasi keempat hingga kedelapan, di mana galat mengalami penurunan yang bersifat kuadratik yang artinya jumlah digit presisi yang benar hampir berlipat dua setiap iterasi. Galat berurutan turun dari 10^{-6} lalu ke 10^{-10} lalu ke 10^{-20} lalu ke 10^{-40} dan lalu ke 10^{-78} .

Sepanjang delapan iterasi, nilai $f'(x)$ konsisten berada di kisaran -49.5 hingga -56 dan tidak pernah mendekati nol, sehingga tidak terjadi risiko pembagian dengan nol yang dapat mengganggu kestabilan komputasi. Oleh karena itu, metode ini dikategorikan stabil selama proses berlangsung, tanpa fluktuasi maupun anomali numerik yang berarti.

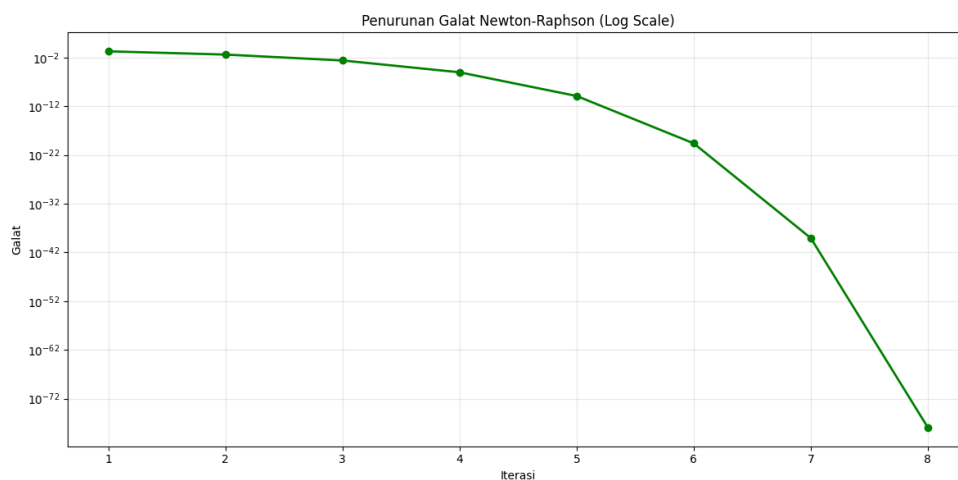


Gambar 7. Hasil konvergensi metode Newton Rhapsion

Pada Gambar 7, terlihat menampilkan perjalanan nilai aproksimasi x dari iterasi 1 hingga 8. Kurva dimulai dari titik tertinggi di iterasi pertama pada nilai ≈ 0.684 dimana hasil ini merupakan hasil koreksi pertama dari titik awal $x_0 = 5$. Walaupun titik awal berada sangat jauh di $x = 5$, grafik ini hanya menampilkan rentang 0.62 hingga 0.70 yang menunjukkan bahwa lompatan besar

dari $x = 5$ ke ≈ 0.684 sudah terjadi sebelum iterasi pertama yang tampil, dan sejak iterasi pertama nilai x sudah berada sangat dekat dengan akar.

Dari iterasi pertama ke iterasi kedua kurva terlihat turun cukup tajam yakni dari konvergensi ≈ 0.684 menuju ≈ 0.644 , kemudian pada iterasi ketiga kembali turun lebih landai menuju ≈ 0.641 yang nyaris menyentuh garis merah putus-putus. Penurunan ini mencerminkan proses koreksi Newton-Raphson yang semakin akurat setiap langkahnya. Sejak iterasi ke-4, nilai x sudah relatif dengan akar sejati dalam presisi yang dapat ditampilkan oleh skala grafik. Kurva Newton-Raphson ini turun dari atas secara monoton tanpa pernah melampaui garis akar dan ini mencerminkan pendekatan yang lebih terkontrol dan terarah berkat penggunaan informasi turunan.



Gambar 8. Hasil penurunan galat metode Newton Rhapson

Pada Gambar 8, Grafik kedua menampilkan penurunan galat dari iterasi 1 hingga 8 dalam skala logaritmik dengan sumbu Y membentang dari 10^{-2} hingga 10^{-72} . Ciri paling mencolok dari grafik ini adalah bentuk kurvanya yang tidak lurus melainkan melengkung semakin curam ke bawah. Pada iterasi pertama hingga keempat, kurva terlihat relatif landai dengan penurunan galat yang bertahap dari $\sim 10^{-2}$ menuju $\sim 10^{-7}$. Mulai iterasi kelima, kurva mulai menekuk tajam yakni galat melompat dari $\sim 10^{-10}$ pada iterasi 5 ke $\sim 10^{-20}$ pada iterasi 6, lalu ke $\sim 10^{-40}$ pada iterasi 7, dan akhirnya mencapai $\sim 10^{-78}$ pada iterasi kedelapan. Dalam tiga iterasi terakhir saja galat turun sebesar 68 orde besaran.

Pola melengkung ini secara matematis mencerminkan sifat konvergensi kuadratik Newton-Raphson, di mana jumlah digit presisi yang benar hampir berlipat dua setiap iterasi begitu metode sudah berada dekat dengan akar. kurva Newton-Raphson ini melengkung paling dramatis dan paling curam yang membuktikan superioritas konvergensi kuadratiknya secara visual yang sangat jelas dan meyakinkan.

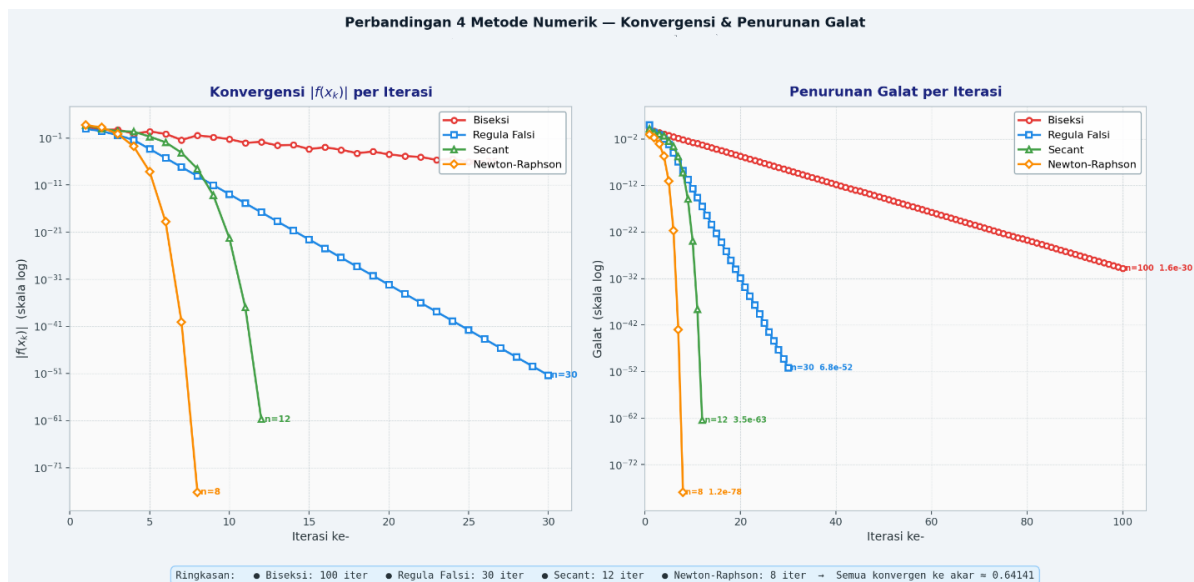
3.5 Pembahasan

Berdasarkan hasil iterasi kedua metode terhadap persamaan $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$, dengan rentang interval $[0,2]$ dapat dikumpulkan hasil sebagai berikut.

Tabel 5. Perbandingan metode aproksimasi

No	Nama Metode	Iterasi terakhir	Kestabilan	Akar Aproksimasi	Galat
1	Biseksi	100	Tidak Stabil	0.641409591970798502	1.57772×10^{-30}
2	Regula Falsi	30	Stabil	0.641409591970798502	6.75548×10^{-52}
3	Secant	12	Stabil	0.641409591970798502	3.45783×10^{-63}
4	Newton Rhapson	8	Stabil	0.641409591970798502	1.23×10^{-78}

Dari keempat metode, perbedaan dilihat pada grafik berikut.



Gambar 9. Hasil penurunan galat metode Newton Rhapson

Dari tabel perbandingan di atas, Seluruh keempat metode berhasil menemukan akar persamaan yang sama, yaitu $x \approx 0.641409591970798502$ akan tetapi dengan jumlah iterasi dan tingkat efisiensi yang sangat berbeda. Newton-Raphson pada penelitian ini menjadi metode paling efisien dengan hanya membutuhkan 8 iterasi dan menghasilkan galat terkecil sebesar 1.23×10^{-78} kemudian Metode Secant dengan 12 iterasi dan galat 3.46×10^{-63} , kemudian Regula Falsi dengan 30 iterasi dan galat 6.76×10^{-52} , serta terakhir Biseksi sebagai metode yang paling banyak membutuhkan iterasi yakni 100 iterasi dengan galat 1.58×10^{-30} .

Tentang Kestabilan, Metode Secant dan Newton-Raphson dikategorikan stabil karena konvergensi bersifat superlinear hingga kuadratik yang menjelaskan bahwa galat berkurang secara eksponensial yang sangat cepat pada setiap iterasinya. Akan tetapi, Metode Biseksi dan Regula Falsi dikategorikan tidak stabil dalam konteks laju konvergensi karena penurunan galatnya bersifat linier dan lebih lambat meskipun keduanya tetap menjamin konvergensi selama syarat awal terpenuhi.

Temuan penelitian ini sejalan dengan hasil yang dilaporkan oleh hasil penelitian Apriansyah dkk di tahun 2025. dengan menyatakan bahwa metode Newton-Raphson lebih cepat karena memiliki konvergensi tinggi, terutama jika tebakan awal mendekati akar, sementara metode biseksi unggul dalam kestabilan dan kepastian konvergensi selama akar berada dalam

selang yang ditentukan (Apriansyah dkk, 2024). Meskipun demikian, terdapat sejumlah literatur yang memberikan perspektif berbeda dan perlu dipertimbangkan secara kritis. Batarius pada tahun 2019 dalam penelitiannya tentang perbandingan metode Newton-Raphson dan Secant yang dimodifikasi juga mengingatkan bahwa metode Newton-Raphson dan metode Secant memiliki kelebihan dan kekurangan masing-masing yang perlu ditinjau dari segi kecepatan konvergensi dan keduanya merupakan metode terbuka yang tidak selalu menjamin konvergensi pada setiap kasus (Batarius, 2019).

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, persamaan polinomial transenden $f(x) = x^8 - 40x^3 + e^{-x} + 10 = 0$ pada interval $[0, 2]$ berhasil diselesaikan menggunakan empat metode numerik yaitu Biseksi, Regula Falsi, Secant, dan Newton-Raphson. Keempat metode tersebut menghasilkan akar aproksimasi yang sama dan konsisten satu sama lain, sehingga validitas akar yang ditemukan dapat dikonfirmasi. Dari segi jumlah iterasi dan efisiensi, metode Biseksi membutuhkan iterasi paling banyak dengan galat akhir yang paling besar, sedangkan metode Newton-Raphson terbukti paling efisien dengan jumlah iterasi paling sedikit dan galat akhir terkecil. Dari aspek kestabilan, metode Biseksi menunjukkan laju konvergensi paling lambat karena bersifat linier, sementara Regula Falsi, Secant, dan Newton-Raphson menunjukkan konvergensi yang lebih stabil dan cepat dengan penurunan galat yang lebih konsisten. Apabila kecepatan konvergensi dan tingkat akurasi menjadi prioritas utama, metode Newton-Raphson lebih direkomendasikan. Namun, jika kesederhanaan implementasi dan jaminan konvergensi lebih diutamakan tanpa memerlukan informasi turunan fungsi, Metode Biseksi atau Regula Falsi menjadi pilihan yang lebih tepat. Secara komputasional, keempat metode ini dapat diimplementasikan melalui perangkat lunak matematis, pemrograman komputer, maupun sistem komputasi ilmiah lainnya untuk menyelesaikan permasalahan pencarian akar persamaan nonlinear dalam berbagai bidang aplikasi teknik dan sains, seperti pemodelan *Reinforcement Learning* dan *perceptron* (Bose, S., Hilmarsson, 2026).

DAFTAR PUSTAKA

- Widyanto, W., Saputra, F. A. R., & Rachman, A. (2022). Perancangan program perhitungan solusi numerik menggunakan metode Bolzano pada bahasa Python. *KERNEL: Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika*, 3(1), 57–63. <https://doi.org/10.31284/j.kernel.2022.v3i1.2557>
- Darmanto, Kartono, & Friska. (2022). Penerapan metode tertutup pada perancangan aplikasi perhitungan persamaan non linear berbasis Android. *Jurnal InTekSis*, 9(2), 66–74.
- Rosidi, M. (2019). Metode numerik menggunakan R untuk teknik lingkungan. Bookdown. https://bookdown.org/moh_rosidi2610/Metode_Numerik/
- Putri, A. D., Setiawan, A., & Rahmawati, I. (2026). Menentukan akar persamaan menggunakan metode bagi dua (bisection). *Sinergi: Jurnal Ilmiah Multidisiplin*, 2(1), 1014–1019
- Islam, M. M., Islam, M. S., & Saiduzzaman, M. (2022). An advanced approach of Regula-Falsi method exploiting the bisection method. *Journal of Interdisciplinary Mathematics*, 25(7), 1999–2006. <https://doi.org/10.1080/09720502.2022.2120015>
- Bhardwaj, R. (2025). Numerical simulation of nonlinear equations by modified bisection and Regula Falsi method. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, 62(1), 11–19.

- Wigati, J. (2020). Solusi numerik persamaan non-linier dengan metode Bisection dan Regula Falsi. *G-Tech: Jurnal Teknologi Terapan*, 1(1), 5–17. <https://doi.org/10.33379/gtech.v1i1.262>
- Ismuniyarto, I. (2016). Perbandingan metode pengapitan akar (Bisection, Regula Falsi, dan Secant) persamaan non linear dalam menyelesaikan analisis break even [Skripsi/Tesis tidak diterbitkan]. Universitas Muhammadiyah Surakarta.
- Yi, J., Zhou, C., Lu, Z., & Pan, Q. (2020). Efficient adaptive Kriging-based reliability analysis combining new learning function and error-based stopping criterion. *Structural and Multidisciplinary Optimization*, 62(5), 2517–2536. <https://doi.org/10.1007/s00158-020-02621-3>
- Ersöz, A. (2013). Algorithmic approach and an application for bisection method using Python. In *Proceedings of the International Conference on Applied Mathematics* (pp. 1–5). Publisher.
- Bandiyah, S. N., & Hidayat, R. (2024). Determining the optimal chemical concentration with the Regula Falsi method. *International Journal of Technology and Modeling*, 3(3), 152–160.
- Kurudirek, A. (2026). An approach to solving transcendental equations for high school students using technology. *Eurasia Journal of Mathematics, Science and Technology Education*, 22(1), Article em2398. <https://doi.org/10.29333/ejmste/XXXXX>
- Ginatra, N. L. W. S. R., Giri, P. A. S., Wanto, A., & Suwardika, G. (2021). Performance one-step secant training method for forecasting cases. *Journal of Physics: Conference Series*, 1933(1), Article 012032. <https://doi.org/10.1088/1742-6596/1933/1/012032>
- Badr, E., Almotairi, S., & El Ghamry, A. (2021). A comparative study among new hybrid root finding algorithms and traditional methods. *Mathematics*, 9(11), Article 1306. <https://doi.org/10.3390/math9111306>
- Chen, Q., Jiao, X., & Yang, O. (2021). Robust and efficient multilevel-ILU preconditioning of hybrid Newton–GMRES for incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 93(12), 3405–3423. <https://doi.org/10.1002/flid.5039>
- Apriansyah, Dynova, A. P., Wicaksana, A. A., Alfian, P. N. S., Misrianto, & Jalanidi, N. (2024). Perbandingan metode biseksi dan Newton-Raphson dalam memprediksi jumlah penduduk Kota Balikpapan. *Jutin: Jurnal Teknik Industri Terintegrasi*, 8(3), 1234–1242. <https://doi.org/10.31004/jutin.v8i3.47633>
- Batarius, P. (2018). Perbandingan metode Newton-Raphson modifikasi dan metode secant modifikasi dalam penentuan akar persamaan. Dalam *Prosiding Seminar Nasional Riset dan Teknologi Terapan 8 (RITEKTRA 8)* (hlm. IK-53–IK-58). Universitas Katolik Widya Mandira.
- Bose, S., Hilmarsson, H., & Suri, D. (2026). Newton's lantern: A reinforcement learning framework for finetuning AC power flow warm start models (arXiv:2605.11102). *arXiv*. <https://doi.org/10.48550/arXiv.2605.11102>