

Original Article



e-ISSN: 2774-2016 - <https://journal.itera.ac.id/index.php/indojam/>
p-ISSN: 2774-2067

Received 19th March 2026
Accepted 14th April 2026
Published 30th April 2026

Open Access

DOI

<https://doi.org/10.35472/indojam.v6i1.2450>

Studi Komparatif Efisiensi Komputasi antara Metode Newton-Raphson dan Metode Birge-Vieta pada Pencarian akar Polinomial

Hafnani^{a,*}, Muna Irawan^a, T. Murdani Saputra^a, Taufiq Iskandar^a, Syarifah Meurah Yuni^a, Saiful Amri^a

^{a,b} Departemen Matematika, Universitas Syiah Kuala

* Corresponding E-mail: hafnani@usk.ac.id

Abstract: The Newton-Raphson method is a commonly used numerical method for finding polynomial roots, but its computational efficiency can be a concern. This paper compares the computational efficiency of the Newton-Raphson and Birge-Vieta methods by analyzing their computational time. Both methods have similar convergence rates and complexity. For low-degree polynomials ($n \leq 8$), the Newton-Raphson method shows faster computational time. This is likely due to the larger algorithmic overhead of the Birge-Vieta method.

Keywords: *polynomial roots, Newton-Raphson method, Birge-Vieta method, complexity, computational times*

Abstrak: Metode Newton-Raphson adalah salah satu metode numerik yang umum digunakan dalam pencarian akar polinomial namun efisiensi komputasinya dapat menjadi perhatian. Pada tulisan ini, dilakukan perbandingan efisiensi komputasi metode Newton-Raphson dan metode Birge-Vieta dengan menganalisa waktu komputasi. Kedua metode mempunyai laju konvergensi dan kompleksitas yang sama. Berdasarkan hasil komputasi, untuk polinomial berderajat rendah ($n \leq 8$), diperoleh metode Newton-Raphson menunjukkan waktu komputasi yang lebih cepat. Hal ini mungkin disebabkan oleh *overhead* algoritma yang berasal dari proses pembagian sintesis pada Birge-Vieta.

Kata Kunci: *akar polinomial, metode Newton-Raphson, metode Birge-Vieta, kompleksitas, waktu komputasi*

Introduction

Metode numerik adalah perhitungan yang dilakukan secara berulang-ulang (iteratif) untuk mendapatkan hasil yang mendekati nilai solusi eksak dengan bantuan operasi aritmatika [1]. Salah satu penerapan metode numerik dalam perhitungan aritmatika yaitu menentukan akar-akar persamaan polinomial. Metode numerik menawarkan hampiran solusi yang andal yang dapat dihitung secara efisien menggunakan komputer untuk mengatasi kendala dalam mencari solusi analitik. Kendala tersebut dinyatakan oleh Teorema Abel, yaitu tidak ada rumus

aljabar umum (menggunakan operasi aritmatika dan akar) untuk mencari akar polinomial berderajat lebih dari 4 [2].

Metode Newton-Raphson merupakan salah satu metode yang paling umum digunakan untuk pencarian akar polinomial, memiliki orde konvergensi kuadratik. Artinya, galat pada iterasi berikutnya berbanding lurus dengan kuadrat galat pada iterasi sebelumnya [3, 4]. Metode Birge-Vieta adalah metode khusus untuk pencarian akar polinomial. Metode ini sebenarnya adalah penyederhanaan metode Newton-Raphson untuk kasus polinomial dengan menggunakan pembagian sintetik (metode Horner) untuk menghitung nilai fungsi polinomial dan turunan



Original Article

pertama fungsinya [5]. Sama seperti metode Newton-Raphson, metode ini juga konvergen secara kuadratik. Meskipun kedua metode tersebut mempunyai laju konvergensi sama, efisiensi komputasi keduanya diduga berbeda. Dugaan tersebut akan diuji dengan memilih polinomial berderajat 3, 4, 5, 6, 7, dan 8 untuk melihat waktu komputasi.

Penelitian sebelumnya telah melakukan perbandingan antara dua atau lebih metode numerik untuk mencari akar persamaan atau polinomial. Nurman membandingkan metode Muller dan metode Birge-Vieta untuk pencarian akar polinomial contoh kasus polinomial derajat 3, 4, 5, 6, dan 7. Hasil penelitian menunjukkan pada metode Muller membutuhkan iterasi yang sedikit dan memperoleh akar real dan kompleks, tetapi galatnya masih besar dan membutuhkan waktu komputasi yang panjang, sedangkan pada metode Birge-vieta membutuhkan iterasi yang lebih banyak dan akar yang diperoleh berupa akar real saja, tetapi galatnya kecil dan membutuhkan waktu komputasi yang lebih sedikit [6]. Selanjutnya metode Newton-Raphson dengan dekomposisi Adomian yang dimodifikasi lebih baik dalam pencarian akar polinomial dibandingkan metode Muller, yaitu ditunjukkan dengan iterasi dan galat yang lebih kecil untuk kasus polinomial derajat 5 [7]. Dalam [8], membandingkan dua metode *bracketing* (metode yang menjepit akar), yaitu metode Bagi-Dua dan metode Regula-Falsi. Prosedur diujikan pada polinomial berderajat 3 dan hasil menunjukkan bahwa iterasi dengan metode Regula-Falsi menunjukkan akselerasi yang lebih baik dibandingkan iterasi dengan metode Bagi-Dua, dengan galat relatif yang lebih baik pula. Hal ini sesuai dengan metode Regula-Falsi merupakan perbaikan dari metode Bagi-Dua, yang menggunakan garis lurus untuk pencarian akar, membuatnya lebih cepat, meskipun tidak seaman metode Bagi-Dua pada kasus tertentu [4].

Metode

Metode Newton-Raphson

Rumus metode Newton-Raphson untuk mencari akar dari fungsi $f(x)$ diberikan oleh [4]

$$x_{k+1} = x_n - \frac{f(x_k)}{f'(x_k)}, \quad (1)$$

dengan x_{n+1} adalah taksiran akar pada iterasi berikutnya (yang lebih baik), x_n adalah taksiran akar saat ini, $f(x_n)$ adalah nilai fungsi pada x_n , dan $f'(x_n)$ adalah nilai turunan pertama fungsi pada x_n . Agar metode ini dapat bekerja, fungsi $f(x)$ harus mempunyai turunan pada interval yang memuat akar dan turunan pertama pada akar tidak nol. Selain itu, untuk tebakan awal (x_0) harus cukup dekat dengan akar. Jika x_0 terlalu jauh, maka iterasi dapat divergen atau konvergen ke akar yang berbeda. Berikut adalah algoritma metode Newton-Raphson [4]:

1. Mendefinisikan fungsi f dan mencari turunan pertama f .
2. Menentukan nilai tebakan awal x_0 dan toleransi galat ϵ .
3. Melakukan iterasi untuk menghitung taksiran akar berikutnya (x_{k+1}) menggunakan Persamaan (1).
4. Iterasi dihentikan bila

$$\left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| < \epsilon. \quad (2)$$
5. Jika kondisi pada Langkah (4) belum terpenuhi, maka ulangi Langkah 3 dengan menuliskan $x_k = x_{k+1}$.

Metode Birge-Vieta

Metode Birge-Vieta adalah metode yang dirancang khusus untuk mencari akar polinomial dan secara fundamental didasarkan pada prinsip metode Newton-Raphson. Perbedaan utamanya adalah dalam menghitung nilai fungsi dan nilai turunannya secara efisien menggunakan metode Horner [5]. Metode Newton-Raphson (Persamaan 1) diterapkan pada polinomial berderajat n :

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (3)$$

menjadi

$$x_{k+1} = x_k - \frac{P_n(x_k)}{P'_n(x_k)}. \quad (4)$$

Pada polinomial berderajat tinggi, menghitung langsung nilai $P_n(x)$ dan $P'_n(x)$ akan memakan waktu

dan rentan terhadap galat pembulatan. Metode Birge-Vieta menggunakan metode Horner untuk menghitung kedua nilai ini secara simultan. Metode Horner adalah cara yang efisien untuk menghitung polinomial $P_n(x)$ pada $x = r$ dan mempunyai sifat [9]:

1. Jika $P_n(x)$ dibagi oleh $(x - r)$ menggunakan pembagian sintetik, maka sisa pembagiannya sama dengan $P_n(r)$.
2. Jika sisa pembagian dibagi lagi oleh $(x - r)$, maka sisa pembagian kedua sama dengan $P_n'(r)$.

Kedua sifat tersebut dimanfaatkan pada metode Birge-Vieta untuk menghitung $P_n(x_k)$ dan $P'(x_k)$ melalui dua kali pembagian sintetik dengan tebakan x_k . Berikut langkah metode Birge-Vieta [5, 6]:

1. Mendefinisikan polinomial P_n .
2. Menentukan tebakan awal x_0 dan toleransi galat ϵ .
3. Melakukan pembagian sintetik pertama untuk menghitung nilai $P_n(x_k)$ dan menghasilkan koefisien b_i dari polinomial tereduksi $B(x)$.

Koefisien $P_n(x)$	a_n	a_{n-1}	$\cdots$	a_1	a_0
Tebakan (x_k)		x_k	$\cdots$	x_k	x_k
Koefisien $B(x)$	b_n	b_{n-1}	$\cdots$	b_1	b_0

Koefisien B dihitung secara rekursif:

$$b_n = a_n \text{ dan}$$

$$b_i = a_i + x_k b_{i+1} \text{ untuk } i = n-1, n-2, \dots, 0, \text{ untuk memperoleh sisa pembagian (nilai } b_0) \text{ dan } P(x_k) = b_0.$$

4. Melakukan pembagian sintetik kedua untuk menghitung nilai $P'(x_k)$, dengan mengambil koefisien b_i (kecuali b_0) dan membaginya lagi dengan x_k . Ini menghasilkan koefisien c_i dari turunan polinomial $P'(x)$.

Koefisien $B(x)$	b_n	b_{n-1}	$\cdots$	b_1
Tebakan (x_k)		x_k	$\cdots$	x_k

Koefisien $C(x)$	c_n	c_{n-1}	$\cdots$	c_1
------------------	-------	-----------	----------	-------

Koefisien C dihitung secara rekursif:

$$c_n = c_n \text{ dan}$$

$$c_i = b_i + x_k c_{i+1} \text{ untuk } i = n-1, n-2, \dots, 1, \text{ untuk memperoleh sisa pembagian (nilai } c_1) \text{ dan } P'(x_k) = c_1.$$

5. Melakukan iterasi Newton Raphson. Hasil yang diperoleh pada Langkah 3 dan 4 disubstitusikan ke Persamaan (4):

$$x_{k+1} = x_k - \frac{b_0}{c_1}. \quad (5)$$

6. Iterasi dihentikan bila terpenuhi kondisi (2).
7. Jika kondisi pada Langkah 4 belum terpenuhi, maka ulangi Langkah 3 dengan menuliskan $x_k = x_{k+1}$.

Kompleksitas Waktu

Kompleksitas waktu dihitung berdasarkan jumlah operasi aritmatika dasar yang dibutuhkan per iterasi. Baik metode Newton-Raphson maupun metode Birge-Vieta mempunyai kompleksitas waktu yang sama, yaitu operasi dasar saat

- menghitung nilai polinomial $P_n(x_k)$ melibatkan n operasi penjumlahan dan n operasi perkalian;
- menghitung nilai polinomial $P_n'(x_k)$ melibatkan $n-1$ operasi penjumlahan dan $n-1$ operasi perkalian;
- iterasi Newton-Raphson melibatkan 1 operasi pengurangan dan 1 operasi pembagian.

Dengan demikian diperoleh kompleksitas totalnya adalah $2n + 2(n-1) + 2 = 4n$.

Implementasi Numerik

Tulisan ini akan menguji kedua metode dengan memilih polinomial pada pada soal latihan di [10] untuk polinomial berderajat 3, 4, 5, 6, dan 7; dan soal latihan di [11] untuk polinomial berderajat 8:

$$(a) P_3(x) = 2x^3 - 13x^2 + 32x - 13.$$

$$(b) P_4(x) = x^4 + 2,1x^3 + 2,52x^2 + 2,1x - 3,52.$$

$$(c) P_5(x) = x^5 - 156x^4 - 5x^3 + 780x^2 + 4x - 624.$$

Original Article

$$(d) P_6(x) = x^6 + 4x^5 - 8x^4 - 34x^3 + 57x^2 + 130x - 150.$$

$$(e) P_7(x) = 8x^7 + 28x^6 - 13x^4 - 124x^3 + 19x^2 + 220x - 100.$$

$$(f) P_8(x) = 5x^8 - 2,25x^6 - 120x^4 + 15x^3 - x + 90.$$

Untuk nilai tebakan awal menggunakan dua variasi nilai, yaitu $x_0 = 0,5$ dan $x_0 = 2$, sedangkan untuk batas toleransi iterasi $\epsilon = 0,000001$. Algoritma kedua metode numerik yang kami dijalankan dengan program MATLAB.

Hasil dan Diskusi

Hasil komputasi pencarian akar polinomial dengan metode Newton-Raphson dan metode Birge-Vieta untuk nilai tebakan awal $x_0 = 0,5$ dapat dilihat pada Tabel 1 berikut.

Tabel 1. Hasil komputasi menggunakan metode Newton-Raphson (NR) dan metode Birge-Vieta (BV) untuk $x_0 = 0,5$.

Contoh Polinomial	Hasil akar	Banyak iterasi	Waktu Komputasi	
	Metode NR dan Metode BV	Metode NR	Metode NR	Metode BV
P_3	0,5	1	0,0039	0,0143
P_4	1,1	8	0,0195	0,0334
P_5	1	5	0,0176	0,0322
P_6	1	4	0,0122	0,0234
P_7	0,5	1	0,0063	0,0205
P_8	0,721428	5	0,0204	0,0333

Hasil yang diperoleh menunjukkan bahwa waktu komputasi metode Newton-Raphson lebih cepat. Secara teoritis, kedua metode memiliki laju konvergensi kuadratik, namun kompleksitas operasional per iterasi metode Birge-Vieta sedikit lebih tinggi karena adanya tambahan pada pembagian sintesis.

Hal yang sama juga terjadi untuk variasi nilai tebakan awal $x_0 = 2$, yaitu waktu komputasi metode Newton-Raphson lebih cepat. Hal ini terjadi mungkin disebabkan *overhead* algoritma Birge-Vieta yang lebih besar. Hasil komputasi lengkap dapat dilihat pada Tabel 2.

Tabel 2. Hasil komputasi menggunakan metode Newton-Raphson (NR) dan metode Birge-Vieta (BV) untuk $x_0 = 2$.

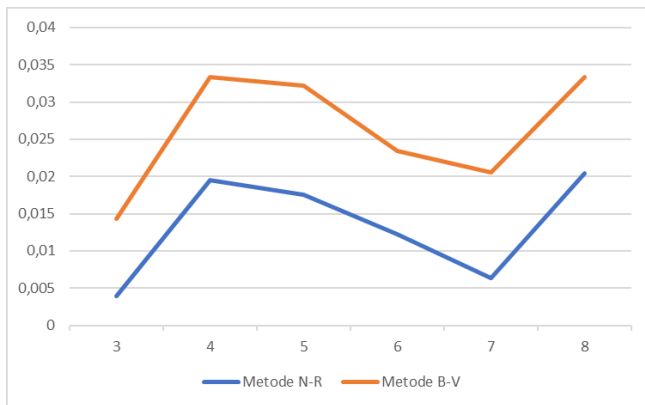
Contoh Polinomial	Hasil akar	Banyak iterasi	Waktu Komputasi	
	Metode NR dan Metode BV	Metode NR	Metode NR	Metode BV
P_3	0,5	7	0,0189	0,0347
P_4	1,1	6	0,0181	0,0324
P_5	2	1	0,0052	0,0156
P_6	1	5	0,0233	0,0334
P_7	0,5	9	0,0271	0,0363
P_8	2,320248	10	0,0269	0,0394

Pada algoritma Birge-Vieta terjadi *overhead* lebih besar, kemungkinan disebabkan oleh:

- Terdapat operasi tambahan seperti pengelolaan *array* atau variabel sementara untuk menyimpan koefisien hasil pembagian sintesis pertama.
- Pemanggilan fungsi atau *loop* bersarang yang mungkin lebih kompleks untuk menjalankan dua tahap pembagian sintesis.
- Dibandingkan dengan Newton-Raphson, yang menggunakan dua panggilan fungsi terpisah (satu untuk $P_n(x)$ dan satu untuk $P'_n(x)$), *overhead* ini mungkin memakan waktu lebih banyak pada MATLAB.

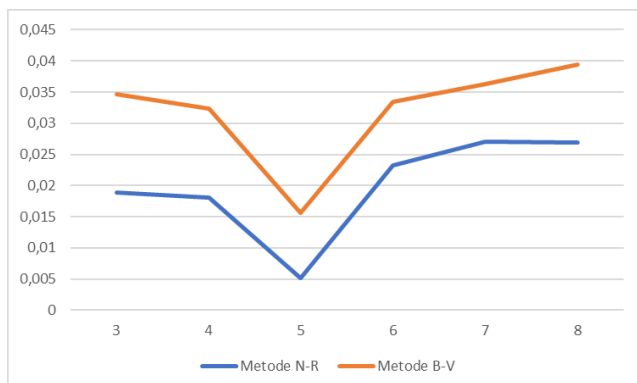
Berdasarkan Tabel 1 dan Tabel 2 juga dapat dilihat perbedaan penemuan hasil akar untuk polinomial derajat 5 dan polinomial derajat 8. Hal ini menunjukkan konvergensi hasil akar bergantung pada

kemungkinan tebakan awal sama dengan akar sebenarnya. Waktu komputasi yang paling cepat untuk nilai tebakan awal $x_0 = 0,5$ terjadi pada polinomial derajat 3 dan 7 untuk masing-masing metode (Gambar 1). Sedangkan untuk nilai tebakan awal $x_0 = 2$ terjadi pada polinomial derajat 5 (Gambar 2). Hal ini disebabkan oleh tebakan awal sama dengan hasil akar.



Gambar 1. Grafik waktu komputasi terhadap polinomial untuk nilai tebakan awal $x_0 = 0,5$.

Pada Gambar 1 dan 2 juga dapat dilihat secara kasar bahwa selisih waktu antara kedua metode semakin kecil untuk polinomial lebih tinggi. Selisih waktu terkecil terjadi pada polinomial derajat 3 dan diikuti oleh polinomial derajat 6, 8, 4, 7, dan 5 untuk nilai tebakan awal $x_0 = 0,5$ (Gambar 1). Namun untuk nilai tebakan awal $x_0 = 2$, selisih jarak terkecil terjadi pada polinomial derajat 7 diikuti polinomial derajat 6, 5, 8, 4, 3.



Gambar 2. Grafik waktu komputasi terhadap polinomial untuk nilai tebakan awal $x_0 = 2$.

Dengan mempertimbangkan pemilihan nilai tebakan awal, metode Birge-Vieta memiliki potensi lebih unggul pada polinomial berderajat yang lebih tinggi dibandingkan metode Newton-Raphson.

Kesimpulan

Meskipun secara teoritis memiliki kompleksitas yang sama, waktu komputasi nyata kedua metode bervariasi tergantung pada derajat polinomial. Untuk polinomial berderajat rendah, metode Newton-Raphson menjadi pilihan yang lebih baik karena implementasinya yang lebih sederhana dan minim *overhead*. Untuk polinomial berderajat tinggi, metode Birge-Vieta memiliki potensi keunggulan.

Oleh karena itu, dapat dilakukan penelitian lanjutan untuk menguji polinomial berderajat tinggi (misalnya, $n > 50$). Selain itu, melakukan analisa untuk kasus akar ganda atau kompleks.

Konflik Kepentingan

Tidak ada konflik kepentingan yang dinyatakan.

Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Ketua Departemen Matematika Universitas Syiah Kuala.

Referensi

- [1] S. N. Hutagalung, "Pemahaman Metode Numerik (Studi Kasus Metode New-Rhapson) Menggunakan Pemrograman Matlab", Jurnal Teknologi Informasi, Vol. 1, pp. 95-100, Juli 2017
- [2] V. B. Alekseev, *Abel's Theorem in Problems and Solutions*, New York: Kluwer Academic Publishers, 2004.
- [3] S. D. Conte and Carl de Boor, *Elementary Numerical Analysis*, New York: McGraw-Hill Book Company, 1980.

Original Article

- [4] S. C. Chapra and R. P. Canale, Numerical Methods for Engineers, New York: McGraw-Hill Education, 2015.
- [5] Deepika and P. Kumar, Solution Of Non-Linear Equations in One Variable, New Delhi: Indira Gandhi National Open University, 2021.
- [6] T. A. Nurman, "Analisis perbandingan metode Muller dan metode Birge-vieta dalam menyelesaikan persamaan polinomial", Jurnal MSA (Matematika dan Statistika serta Aplikasinya), Vol. 9, pp. 81-88, Desember 2021.
- [7] T. A. Nurman, A. Mariani, dan R. Nurhidayat, "Perbandingan Metode Muller dan Metode Newton-Raphson dengan Dekomposisi Adomian yang Dimodifikasi dalam Menyelesaikan Persamaan Polinomial", Journal of Mathematics: Theory and Applications, Vol. 6, pp. 42-54, April 2024.
- [8] N. Hafidz, M. Rofi, dan A. Priyanto, "Penggunaan Prosedur Regula-Falsi dan Interval Bagi Dua untuk Pencarian Akar pada Fungsi Polinomial Berderajat Tiga", Journal Social, Science, and Education (Soscied), Vol. 6, pp. 6-12, Juli 2023.
- [9] Tian-Xiao He and P. Shiue, "A note on Horner's method", Journal of Concrete and Applicable Mathematics, Vol. 10, 2012
- [10] J. Kiusalaas, Numerical Methods in Engineering with Python 3, New York: Cambridge University Press, 2013.
- [11] R. K. Setyansah dan D. Apriandi, Analisis Numerik: Pemograman Matlab berbasis Simulasi e-Learning, Yogyakarta: Deepublish, 2018.