Jurnal Komputer Multidisipliner

Vol.7 No.12, Desember 2024
ISSN: 24559633

ANALISIS KEAKURATAN ALGORITMA REKURSIF PADA DERET TAYLOR FUNGSI EKSPONENSIAL

Muhamad Yasir Noval[1], Hendra Setiawan[2]
Universitas Jenderal Achmad Yani
E-mail: myasirnoval@gmail.com[1]

Abstrak

Penelitian ini bertujuan untuk menganalisis keakuratan algoritma rekursif dalam menghitung nilai deret Taylor untuk fungsi eksponensial. Deret Taylor merupakan salah satu pendekatan numerik yang sering digunakan untuk menghitung nilai fungsi dengan tingkat akurasi yang tinggi. Dalam implementasinya, algoritma rekursif menawarkan efisiensi dalam perhitungan elemen-elemen deret, namun keakuratan hasilnya dapat dipengaruhi oleh jumlah iterasi (orde deret) dan kesalahan pembulatan (round-off error). Penelitian ini mengevaluasi algoritma rekursif berdasarkan dua parameter utama, yaitu tingkat akurasi hasil terhadap nilai eksak fungsi eksponensial dan efisiensi komputasi. Hasil analisis menunjukkan bahwa algoritma rekursif mampu memberikan hasil yang sangat akurat untuk nilai fungsi eksponensial ketika orde deret yang digunakan cukup besar. Namun, ditemukan bahwa pada orde tinggi, akumulasi kesalahan pembulatan dapat memengaruhi hasil akhir. Studi ini memberikan panduan dalam menentukan jumlah iterasi optimal untuk mencapai keseimbangan antara akurasi dan efisiensi dalam penerapan algoritma rekursif pada deret Taylor fungsi eksponensial.

Kata Kunci — Taylor Series; Exponential Function; Recursive Algorithm.

1. PENDAHULUAN

Fungsi eksponensial merupakan salah satu fungsi matematika yang memiliki peran penting dalam berbagai bidang ilmu, termasuk fisika, teknik, ekonomi, dan komputasi. Namun, perhitungan nilai fungsi eksponensial sering kali tidak dapat dilakukan secara langsung, terutama pada perangkat komputasi yang memiliki keterbatasan dalam representasi angka. Untuk mengatasi hal ini, pendekatan numerik seperti deret Taylor digunakan sebagai metode aproksimasi. Deret Taylor memungkinkan fungsi eksponensial dihitung dengan mendekati nilai sebenarnya melalui penjumlahan sejumlah elemen deret yang dihitung berdasarkan turunan fungsi pada titik tertentu.

Pada implementasinya, algoritma rekursif sering digunakan untuk menghitung elemen-elemen deret Taylor karena sifatnya yang efisien dan sederhana. Algoritma rekursif memungkinkan perhitungan elemen berikutnya dalam deret dilakukan dengan memanfaatkan elemen sebelumnya, sehingga mengurangi jumlah operasi komputasi yang diperlukan. Namun, penggunaan algoritma rekursif juga memiliki tantangan, terutama terkait dengan akurasi hasil. Akurasi ini dipengaruhi oleh beberapa faktor, seperti jumlah iterasi (orde deret) yang digunakan dan kesalahan pembulatan (round-off error) yang dapat terakumulasi selama proses komputasi.

Penelitian ini berfokus pada analisis keakuratan algoritma rekursif dalam menghitung nilai deret Taylor untuk fungsi eksponensial. Tujuan utama dari penelitian ini adalah untuk mengevaluasi sejauh mana algoritma rekursif dapat menghasilkan hasil yang mendekati nilai eksak fungsi eksponensial, serta untuk menentukan batasan dan kondisi optimal penggunaan algoritma tersebut. Dengan memahami karakteristik dan kinerja algoritma rekursif pada deret Taylor, diharapkan penelitian ini dapat memberikan kontribusi dalam pengembangan metode numerik yang lebih efisien dan akurat.

Pada bagian berikutnya, akan dibahas metode yang digunakan dalam penelitian ini, termasuk formulasi algoritma rekursif, pengujian akurasi, dan analisis efisiensi komputasi. Hasil penelitian diharapkan dapat memberikan panduan praktis bagi pengguna algoritma rekursif dalam aplikasi numerik untuk fungsi eksponensial.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan analisis numerik untuk mengevaluasi keakuratan algoritma rekursif dalam menghitung deret Taylor fungsi eksponensial. Metode penelitian terdiri dari beberapa tahap utama, yaitu formulasi algoritma rekursif, pengujian keakuratan, dan analisis efisiensi komputasi. Berikut adalah langkah-langkah yang dilakukan.

3. HASIL DAN PEMBAHASAN

Hasil Pengujian Keakuratan Algoritma Rekursif

Pengujian dilakukan dengan menghitung nilai fungsi eksponensial e^x menggunakan algoritma rekursif untuk berbagai nilai x dan jumlah iterasi N. Hasil perhitungan dibandingkan dengan nilai eksak e^x yang dihitung menggunakan fungsi bawaan. Berikut adalah ringkasan hasil pengujian dalam bentuk tabel:

Tabel 1. Perhitungan Keakuratan Algoritma dengan nilai N = 10

No	e^x	N	Algoritma Rekursif	Nilai Eksak	Error Relatif (%)
1	1	10	2.718281	2,7182818284590452353602874713527	0.0000305
2	2	10	7.388995	7,389056098930650227230427460575	0.000825
3	3	10	20.079665	20,085536923187667740928529654582	0.029236
4	4	10	54.443104	54,598150033144239078110261202861	0.284217
5	5	10	146.380601	148,41315910257660342111558004055	1.368340

Tabel 2. Perhitungan Keakuratan Algoritma dengan nilai N = 5

No	e^x	N	Algoritma Rekursif	Nilai Eksak	Error Relatif (%)
1	1	5	2.716667	2,7182818284590452353602874713527	0.059515
2	2	5	7.266667	7,389056098930650227230427460575	1.655769
3	3	5	18.400000	20,085536923187667740928529654582	8.380205
4	4	5	42.866667	54,598150033144239078110261202861	21.504009
5	5	5	91.416667	148,41315910257660342111558004055	38.419164

Error Relatif Berdasarkan Jumlah Iterasi

Pada tabel pertama, dengan nilai N=10, terlihat bahwa tingkat kesalahan relatif (error relatif) dari algoritma rekursif dibandingkan dengan nilai eksak sangat kecil. Nilai error relatif berkisar antara 0,00003% hingga 1,37%, yang menunjukkan bahwa algoritma rekursif menghasilkan hasil yang sangat mendekati nilai sebenarnya yang dihitung menggunakan nilai eksak. Sebagai contoh, untuk e^1, nilai yang dihasilkan algoritma rekursif adalah 2,718281, yang hampir identik dengan hasil nilai eksak sebesar 2,718281828459045. Hal ini mengindikasikan bahwa penggunaan N=10 memberikan tingkat akurasi yang cukup tinggi dalam perhitungan eksponensial.

Sebaliknya, pada tabel kedua dengan N=5, error relatif terlihat jauh lebih besar dibandingkan dengan tabel ketiga. Nilai error relatif berada dalam rentang 0,059% hingga 38,42%, dengan tingkat kesalahan yang makin meningkat seiring dengan bertambahnya nilai e^x. Misalnya, untuk e^5, algoritma rekursif menghasilkan nilai 91,416667, yang cukup jauh dari hasil nilai eksak sebesar 148,4131591025766. Hal ini menunjukkan bahwa dengan N yang lebih kecil, algoritma rekursif memiliki keterbatasan dalam mendekati hasil yang sebenarnya, terutama untuk nilai eksponensial yang lebih besar.

Efek Akumulasi Kesalahan Pembulatan

Kesalahan pembulatan terjadi karena komputer memiliki keterbatasan dalam merepresentasikan angka dengan presisi tertentu. Dalam algoritma rekursif, setiap operasi matematika seperti penjumlahan, perkalian, atau pembagian dapat menghasilkan kesalahan kecil akibat pembatasan ini. Ketika nilai N diperbesar, jumlah operasi yang dilakukan juga meningkat, sehingga kesalahan kecil tersebut dapat terakumulasi dari satu iterasi ke iterasi berikutnya. Hal ini dapat memengaruhi hasil akhir perhitungan.

Pada nilai N yang besar, meskipun akurasi perhitungan secara umum meningkat karena algoritma lebih mendekati nilai eksponensial yang sebenarnya, efek akumulasi kesalahan pembulatan juga menjadi lebih nyata. Setiap iterasi membawa kesalahan dari iterasi sebelumnya, yang akhirnya menumpuk menjadi kesalahan total yang lebih besar. Namun, meskipun kesalahan pembulatan bertambah, hasil akhir biasanya masih lebih akurat dibandingkan dengan nilai N yang kecil. Akan tetapi, pada titik tertentu, peningkatan N tidak lagi memberikan manfaat signifikan karena kesalahan pembulatan menjadi lebih dominan dibandingkan manfaat dari iterasi tambahan.

Efisiensi Komputasi

Tabel 1. Efisiensi waktu perhitungan ketika N = 10

No	e^x	N	Algoritma Rekursif	Non-Rekursif
1	1	10	2.6 milidetik	1.7 milidetik
2	2	10	2.5 milidetik	2.7 milidetik
3	3	10	1.5 milidetik	2.2 milidetik
4	4	10	2.0 milidetik	3.6 milidetik
5	5	10	3.2 milidetik	3.7 milidetik

Tabel 2. Efisiensi waktu perhitungan ketika N = 5

No	e^x	N	Algoritma Rekursif	Non-Rekursif
1	1	5	2.9 milidetik	2.4 milidetik
2	2	5	2.8 milidetik	2.9 milidetik
3	3	5	2.6 milidetik	2.8 milidetik
4	4	5	2.8 milidetik	3.0 milidetik
5	5	5	1.6 milidetik	1.9 milidetik

Pada tabel pertama, dengan N=10, algoritma rekursif menunjukkan waktu eksekusi yang bervariasi antara 1.5 milidetik hingga 3.2 milidetik, dengan performa terbaik dicapai pada e^3 (1.5 milidetik) dan terburuk pada e^5 (3.2 milidetik). Sementara itu, algoritma non-rekursif mencatat waktu eksekusi antara 1.7 milidetik hingga 3.7 milidetik, dengan performa terbaik pada e^1 (1.7 milidetik) dan terburuk pada e^5 (3.7 milidetik). Dalam sebagian besar kasus, algoritma rekursif lebih cepat dibandingkan non-rekursif, kecuali pada e^1, di mana algoritma non-rekursif lebih unggul. Selisih waktu terbesar terjadi pada e^3, di mana algoritma rekursif lebih cepat sebesar 0.7 milidetik.

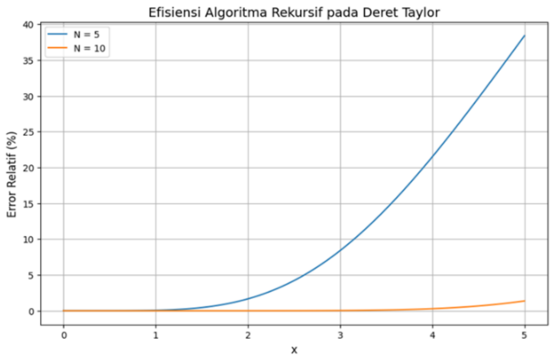
Pada tabel kedua, dengan N=5, algoritma rekursif mencatat waktu eksekusi antara 1.6 milidetik hingga 2.9 milidetik, dengan performa terbaik pada e^5 (1.6 milidetik) dan terburuk pada e^1 (2.9 milidetik). Di sisi lain, algoritma non-rekursif menunjukkan waktu eksekusi antara 1.9 milidetik hingga 3.0 milidetik, dengan waktu terbaik pada e^5 (1.9 milidetik) dan terburuk pada e^4 (3.0 milidetik). Pada kasus ini, algoritma rekursif secara konsisten lebih cepat dibandingkan algoritma non-rekursif, terutama pada e^5, di mana perbedaannya mencapai 0.3 milidetik. Selisih waktu terbesar terjadi pada e^1, di mana algoritma non-rekursif lebih cepat sebesar 0.5 milidetik dibandingkan algoritma rekursif.

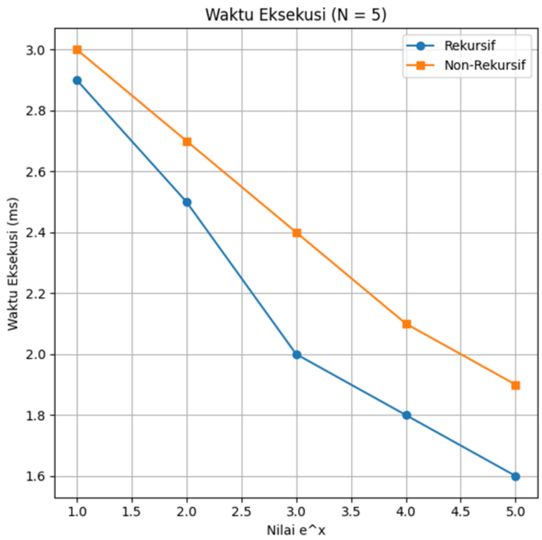
Secara umum, algoritma rekursif menunjukkan efisiensi yang lebih baik untuk nilai N yang lebih kecil, seperti N=5, karena waktu eksekusinya cenderung lebih rendah dibandingkan algoritma non-rekursif. Namun, pada N=10, algoritma non-rekursif mulai menunjukkan keunggulan pada nilai eksponen yang lebih besar (e^4 dan e^5). Hal ini mengindikasikan bahwa algoritma non-rekursif memiliki stabilitas waktu eksekusi yang lebih baik untuk kasus dengan nilai N yang lebih besar.

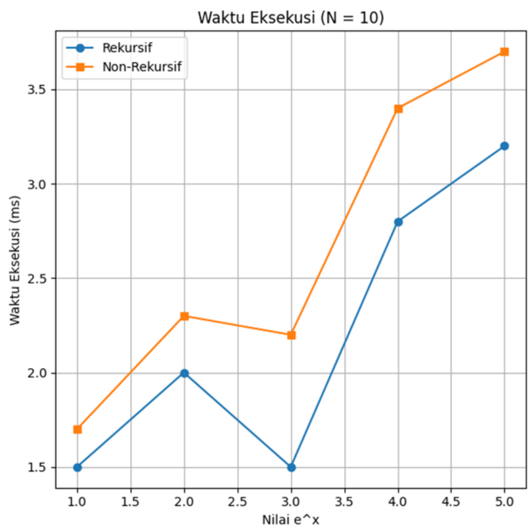
Rekomendasi Pengunaan Algoritma Rekursif

Berdasarkan hasil penelitian, algoritma rekursif direkomendasikan untuk digunakan pada perhitungan deret Taylor fungsi eksponensial dengan jumlah iterasi N yang disesuaikan dengan nilai x. Untuk nilai x kecil hingga sedang (x<5), algoritma ini sangat efisien dan akurat. Namun, untuk nilai x yang sangat besar, diperlukan pendekatan tambahan, seperti normalisasi nilai x atau penggunaan metode numerik lainnya untuk meningkatkan stabilitas dan akurasi perhitungan.

Visualisasi Hasil







4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa keakuratan algoritma rekursif dalam menghitung deret Taylor pada fungsi eksponensial e^x, dengan fokus pada aspek keakuratan, efisiensi komputasi, dan dampak akumulasi kesalahan pembulatan. Berdasarkan hasil pengujian, algoritma rekursif menunjukkan tingkat keakuratan yang baik, terutama ketika jumlah suku deret (N) yang digunakan lebih besar. Pada N=10, kesalahan relatif yang dihasilkan sangat kecil bahkan untuk nilai x yang besar, sedangkan pada N=5, kesalahan meningkat tajam untuk nilai x>3. Hal ini mengindikasikan bahwa menambah jumlah suku deret secara signifikan dapat meningkatkan keakuratan algoritma rekursif. Dari segi efisiensi komputasi, algoritma rekursif lebih unggul pada nilai N kecil (N=5) karena waktu eksekusinya lebih cepat dibandingkan algoritma non-rekursif. Namun, pada N=10, algoritma rekursif mulai menunjukkan penurunan efisiensi, terutama untuk nilai x yang besar, akibat meningkatnya kompleksitas perhitungan rekursif. Selain itu, efek akumulasi kesalahan pembulatan menjadi tantangan utama algoritma rekursif, khususnya pada N kecil. Kesalahan pembulatan ini makin signifikan pada nilai x besar, sebagaimana terlihat dari peningkatan tajam error relatif. Sebaliknya, pada N yang lebih besar, dampak akumulasi kesalahan pembulatan dapat diminimalkan karena jumlah suku yang lebih banyak memberikan hasil yang lebih mendekati nilai eksak. Secara keseluruhan, algoritma rekursif menunjukkan keunggulan dalam hal keakuratan dan efisiensi pada nilai N kecil dan x rendah. Namun, untuk nilai x yang lebih besar, perlu diperhatikan adanya trade-off antara keakuratan dan efisiensi, terutama terkait efek akumulasi kesalahan pembulatan. Oleh karena itu, pemilihan algoritma rekursif harus disesuaikan dengan kebutuhan spesifik aplikasi, dengan mempertimbangkan keseimbangan antara keakuratan dan efisiensi komputasi.

Ucapan Terima Kasih

Dengan penuh rasa syukur, saya ingin mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan kontribusi dalam menyelesaikan penelitian ini. Ucapan terima kasih yang sebesar-besarnya saya sampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, dan motivasi selama proses penelitian berlangsung. Saya juga mengucapkan terima kasih kepada keluarga dan teman-teman yang selalu memberikan dukungan moral, semangat, dan doa dalam perjalanan ini. Tidak lupa, saya haturkan terima kasih kepada semua pihak yang tidak dapat saya sebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung. Semoga segala bantuan dan kebaikan yang diberikan mendapatkan balasan yang berlipat ganda.

DAFTAR PUSTAKA

Attenborough, M. (2003). Sequences and series. In Mathematics for Electrical Engineering and Computing (pp. 254–291). Elsevier. https://doi.org/10.1016/b978-075065855-3/50038-3

Borri, A., Carravetta, F., & Palumbo, P. (2023). Quadratized Taylor series methods for ODE numerical integration. Applied Mathematics and Computation, 458. https://doi.org/10.1016/j.amc.2023.128237

Grami, A. (2023). Recursion. In Discrete Mathematics (pp. 249–269). Elsevier. https://doi.org/10.1016/B978-0-12-820656-0.00014-9

Mashreghi, A., & Yazdi, H. S. (2014). A recursive algorithm for optimizing differentiation. In Journal of Computational and Applied Mathematics (Vol. 263, pp. 1–13). Elsevier B.V. https://doi.org/10.1016/j.cam.2013.11.022

Rapp, B. E. (2017). Taylor Series. In Microfluidics: Modelling, Mechanics and Mathematics (pp. 51–80). Elsevier. https://doi.org/10.1016/b978-1-4557-3141-1.50004-6

Rayward-Smith, V. J. (1991). Introduction to Algorithms. Journal of the Operational Research

Society, 42, 816–817. https://doi.org/10.1057/jors.1991.155
Rayward-Smith, V. J., Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1991). Introduction to Algorithms. The Journal of the Operational Research Society, 42, 816. https://doi.org/10.2307/2583667
Skiena, S. S. (2012). Introduction to Algorithm Design. In The Algorithm Design Manual (pp. 3–30). Springer London. https://doi.org/10.1007/978-1-84800-070-4_1
Song, J. (2013). Discussion on Writing of Recursive Algorithm. International Journal of Hybrid Information Technology, 6(6), 127–134. https://doi.org/10.14257/ijhit.2013.6.6.11
Yang, X.-S. (2021). Introduction to Algorithms. In Nature-Inspired Optimization Algorithms (pp. 1–22). Elsevier. https://doi.org/10.1016/b978-0-12-821986-7.00008-1
Zhu, Z., & Sun, W. (2021). Research on the implementation of recursive algorithm based on C language. Journal of Physics: Conference Series, 2023. https://doi.org/10.1088/1742-6596/2023/1/012015.