Volume 9, Nomor 1, Juni 2025
Numerical: Jurnal Matematika dan Pendidikan Matematika
http://journal.iaimnumetrolampung.ac.id/index.php/numerical

Perbandingan Penggunaan *Python* dan *Excel* dalam Menyelesaikan Persamaan Tak Linier Metode *Newton Raphson*

Nurul Rohmawati[1] , Istiqomah Sarah Nur Inayah[2], Ari Wibowo[3]
[1,2] Universitas Islam Negeri Raden Mas Said Surakarta, Indonesia

Abstrak: Persamaan tak linier sering dijumpai dalam berbagai bidang, seperti fisika, teknik, dan ekonomi. Salah satu metode numerik yang sering digunakan untuk mencari akar dari persamaan tak linier adalah metode *Newton-Raphson*. Dengan kemajuan teknologi, berbagai perangkat lunak kini digunakan untuk mempermudah perhitungan numerik. Dalam konteks alat yang digunakan untuk analisis data, perbandingan antara *Python* dan *Microsoft Excel* telah menjadi bahan diskusi. Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan metode komparatif untuk menganalisis perbedaan hasil antara objek yang dibandingkan serta mengetahui mana yang lebih efektif dan efisien. Hasil penelitian menunjukkan bahwa *Python* lebih unggul dalam hal akurasi hasil, automasi iterasi, kecepatan perhitungan, fleksibilitas perubahan fungsi, reproduksibilitas, dan skalabilitas. Namun, *Python* memerlukan pengetahuan pemrograman dan kurang user-friendly bagi pemula. Sedangkan alat bantu *Microsoft Excel* lebih unggul dalam hal kemudahan penggunaan dan kompatibilitas dengan *software Microsoft*, tetapi kurang cocok untuk masalah yang kompleks atau memerlukan banyak iterasi. *Microsoft Excel* juga rentan terhadap kesalahan manual dan kurang skalabel.

Keywords: *Microsoft Excel*, *Newton Raphson, Python*

Pendahuluan

Dalam berbagai bidang ilmu, banyak permasalahan yang tidak dapat diselesaikan secara langsung menggunakan metode analitik, yaitu metode yang mengandalkan formula aljabar atau rumus matematika standar (Pandu, 2019). Seringkali, masalah matematika yang kompleks atau tingkat lanjut dianggap tidak memiliki solusi. Padahal, masalah-masalah tersebut sebenarnya dapat diselesaikan dengan pendekatan alternatif, yaitu metode numerik (Hsu, 2018; Kreyszig, 2011). Persamaan non-linier adalah persamaan yang grafik fungsinya tidak membentuk garis lurus. Terdapat beberapa jenis persamaan non-linier, yaitu: pertama, persamaan aljabar yang berbentuk polinomial dan rasional; kedua, persamaan transenden yang mencakup fungsi eksponensial, logaritma, trigonometri, dan hiperbolik (Frayudi et al., 2019). Persamaan tak linier sering dijumpai dalam berbagai bidang, seperti fisika, teknik, dan ekonomi. Salah satu metode numerik yang sering digunakan untuk mencari akar dari persamaan tak linier adalah metode *Newton-Raphson*.

Metode ini memanfaatkan turunan pertama dari fungsi untuk melakukan estimasi akar melalui proses iterasi yang berkelanjutan (Batarius, 2018). Menurut Rosidi (2019), metode *Newton-Raphson* mengadopsi pendekatan dengan satu titik awal dan mendekati akar dengan mempertimbangkan gradien di titik tersebut. Dalam konteks matematika, kemampuan memecahkan masalah adalah indikator kunci yang menunjukkan tingkat pemahaman siswa terhadap materi yang dipelajari (Siagan et al., 2019; Sumartini, 2016).

Corresponding author:
Istiqomah Sarah Nur Inayah, Universitas Islam Negeri Raden Mas Said Surakarta, Indonesia, Email: istiqomahsarah008@gmail.com

Received : 22-04-2025, Revised : 17-06-2025, Accepted : 18-06-2025. DOI: https://doi.org/10.25217/numerical.v9.i1.5798

Pemahaman konsep tidak terbatas pada sekadar memperoleh pengetahuan. Ia juga mencakup kemampuan untuk mengartikulasikan kembali konsep tersebut dengan cara yang lebih sederhana dan mudah dimengerti, serta mengaplikasikannya dalam menyelesaikan masalah (Agustina, 2018; Fajar et al., 2019). Pembelajaran matematika idealnya berpusat pada pengembangan kemampuan pemecahan masalah siswa. Salah satu faktor yang menyebabkan siswa kesulitan dalam memecahkan masalah adalah proses berpikir mereka yang perlu ditingkatkan (Taufik & Susanti, 2024).

Terdapat beragam pendekatan berpikir dalam menyelesaikan masalah matematika, salah satunya adalah *Computational Thinking* (CT). CT merupakan suatu cara berpikir dan bertindak yang diwujudkan melalui keterampilan-keterampilan spesifik, yang kemudian menjadi dasar untuk mengukur kemampuan tersebut (Saidin et al., 2021). *Computational Thinking* (CT) mengaplikasikan konsep, alat, dan teknik yang berasal dari ilmu komputer dalam konteks sains, teknologi, dan matematika (Lee et al., 2020; Swaid, 2015). *Computational Thinking* (CT) mengaplikasikan konsep, alat, dan teknik yang berasal dari ilmu komputer dalam konteks sains, teknologi, dan matematika (Angeli & Giannakos, 2020). Ketika dihadapkan pada matematika yang melibatkan rumus rumit, siswa seringkali lebih memilih menghafal dan menggunakan rumus secara langsung daripada memahami konsep dasarnya (Baiduri, 2018). Penggunaan perangkat lunak dapat memfasilitasi pemahaman konsep matematika bagi siswa dan membantu mereka menyelesaikan masalah dengan lebih akurat, terutama dalam konteks materi metode numerik (Yeh et al., 2019).

Dengan kemajuan teknologi, berbagai perangkat lunak kini digunakan untuk mempermudah perhitungan numerik. *Microsoft Excel* telah lama menjadi pilihan populer untuk analisis data dan perhitungan numerik berkat antarmukanya yang intuitif serta fitur-fitur seperti fungsi bawaan dan kemampuan makro VBA. Rozi & Rarasati (2022) mengungkapkan bahwa *Excel* memiliki banyak fitur yang sangat bermanfaat, terutama dalam pengolahan angka. Di sisi lain, *Python* telah muncul sebagai bahasa pemrograman yang kuat untuk analisis data, didukung oleh pustaka-pustaka seperti *NumPy* dan *SciPy* yang memudahkan perhitungan numerik yang kompleks. Menurut Cambridge Spark (*Python vs Excel for Data Analysis*, 2022), *Python* kini dianggap sebagai alat analisis data yang lebih unggul dibandingkan *Excel* oleh banyak pengembang dan komunitas data science.

Penelitian sebelumnya telah melakukan perbandingan terhadap metode numerik dalam menyelesaikan persamaan non-linier. Contohnya, Sunandar & Indrianto (2020) membandingkan efisiensi antara metode *Newton-Raphson* dan metode Secant dalam mencari akar persamaan non-linier dengan menggunakan bahasa pemrograman Java, dan hasilnya menunjukkan bahwa metode *Newton-Raphson* memiliki laju konvergensi yang lebih cepat dibandingkan metode Secant. Selain itu, Dwi Estuningsih et al. (2019) juga membandingkan metode *Bisection* dan metode *Newton-Raphson* dalam menyelesaikan persamaan non-linier, dan mereka menemukan bahwa metode *Newton-Raphson* lebih cepat dalam konvergensi dibandingkan metode Bisection, meskipun metode Bisection tidak memerlukan turunan pertama untuk mencari akar. Dalam konteks alat yang digunakan untuk analisis data, perbandingan antara *Python* dan *Microsoft Excel* telah menjadi bahan diskusi. Oktavian (2024) menekankan bahwa *Python* memberikan fleksibilitas dalam pemrograman dan kemampuan untuk menangani dataset yang besar, sementara *Excel* memiliki keterbatasan dalam hal tersebut.

Namun, tidak banyak literatur yang membandingkan secara langsung penggunaan *Python* dan *Microsoft Excel* dalam penerapan metode *Newton-Raphson* untuk menyelesaikan persamaan non-linier. Penelitian ini akan membandingkan kedua alat ini

berdasarkan faktor-faktor seperti kecepatan perhitungan, kemudahan penggunaan, dan akurasi hasil dalam penerapan metode *Newton-Raphson*. Peneliti memfokuskan pada persamaan non-linear jenis polinomial pangkat tiga. Hasilnya diharapkan dapat membantu praktisi dan akademisi memilih alat yang paling tepat untuk menyelesaikan persamaan non-linier sesuai dengan kebutuhan dan konteks khusus.

Metode

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan metode komparatif untuk menganalisis perbedaan hasil antara objek yang dibandingkan serta mengetahui mana yang lebih efektif dan efisien. Subjek penelitian ini yaitu persamaan nonlinear tertentu yang akan diselesaikan dengan Metode *Newton-Raphson*. Adapun instrumen penelitian ini yaitu kode *Python* untuk *Python* serta rumus dan perhitungan di *Microsoft Excel*. Dengan pendekatan ini, penelitian akan menghasilkan wawasan yang jelas mengenai kemudahan penggunaan, kelebihan dan kekurangan masing-masing aplikasi (*Python* dan *Microsoft Excel* versi 2021) dalam konteks penyelesaian persamaan nonlinear tertentu yang akan diselesaikan dengan Metode *Newton-Raphson*.

Langkah pertama penelitian 1) Menentukan persamaan nonlinear yang akan digunakan dalam penelitian. 2) Memahami teori dan langkah-langkah Metode *Newton-Raphson* untuk menyelesaikan persamaan nonlinear. 3) Menyiapkan skenario penyelesaian menggunakan dua metode, yaitu implementasi Metode *Newton-Raphson* dengan *Python* dan implementasi Metode *Newton-Raphson* di *Microsoft Excel*.

Langkah Kedua Penelitian 1) Melakukan perhitungan manual untuk mendapatkan hasil dan langkah-langkah penyelesaian. 2) Menuliskan kode *Python* di VScode untuk menyelesaikan persamaan dengan Metode *Newton-Raphson*. 3) Menyusun formula dan perhitungan di *Microsoft Excel* untuk mendapatkan hasil yang sama. 4) Menganalisis hasil dari ketiga pendekatan dan membandingkannya berdasarkan akurasi, efisiensi waktu, serta kemudahan penggunaan.

Hasil dan Pembahasan

Persamaan Non-Linear

Misalkan $f(x)$ adalah fungsi yang kontinu. Setiap nilai r dalam domain f yang memenuhi $f(r) = 0$ disebut sebagai akar dari persamaan $f(x) = 0$, atau juga dikenal sebagai pembuat nol dari fungsi $f(x)$. Secara ringkas, r disebut sebagai akar fungsi $f(x)$ (Maharani & Suprapto, 2018). Salah satu contoh dari persamaan non-linier adalah persamaan kuadrat, yang memiliki bentuk umum: $ax^2 + bx + c = 0$. Sebagai contoh, untuk menentukan akar dari persamaan non-linier $x^2 - 5x + 6 = 0$, kita mendapatkan $x_1 = 2\ dan\ x_2 = 3$ (dihitung secara analitik). Akar-akar ini memberikan nilai-nilai yang membuat persamaan tersebut sama dengan nol. Namun, untuk jenis-jenis persamaan non-linier dengan derajat lebih dari dua, sering kali terdapat kesulitan dalam menemukan akar-akarnya (Setiawan, 2007 dalam Pandia & Sitepu, 2021).

Peneliti memilih fungsi polinomial derajat tiga sebagai contoh karena jenis fungsi ini cukup sederhana untuk dianalisis, tetapi tetap mencerminkan kompleksitas yang khas dari persamaan non-linier. Selain itu, turunan dari fungsi kubik masih mudah dihitung baik secara manual maupun menggunakan program, sehingga cocok untuk dibandingkan antara *Python* dan *Excel*. Jika fungsi yang digunakan memiliki pangkat pecahan seperti $x^{\frac{1}{2}}$ atau $x^{\frac{1}{3}}$, perhitungan menjadi lebih rumit karena adanya batasan domain dan kemungkinan nilai turunan yang tidak terdefinisi pada titik tertentu. Dalam situasi ini, diperlukan validasi input dan penyesuaian skrip agar iterasi tetap berjalan dengan stabil.

Hal ini juga ditekankan oleh Kreyszig (2011), yang menyatakan bahwa untuk menerapkan metode *Newton-Raphson*, syarat-syarat seperti keberadaan turunan yang kontinu dan tidak nol harus dipenuhi agar metode dapat konvergen dengan baik.

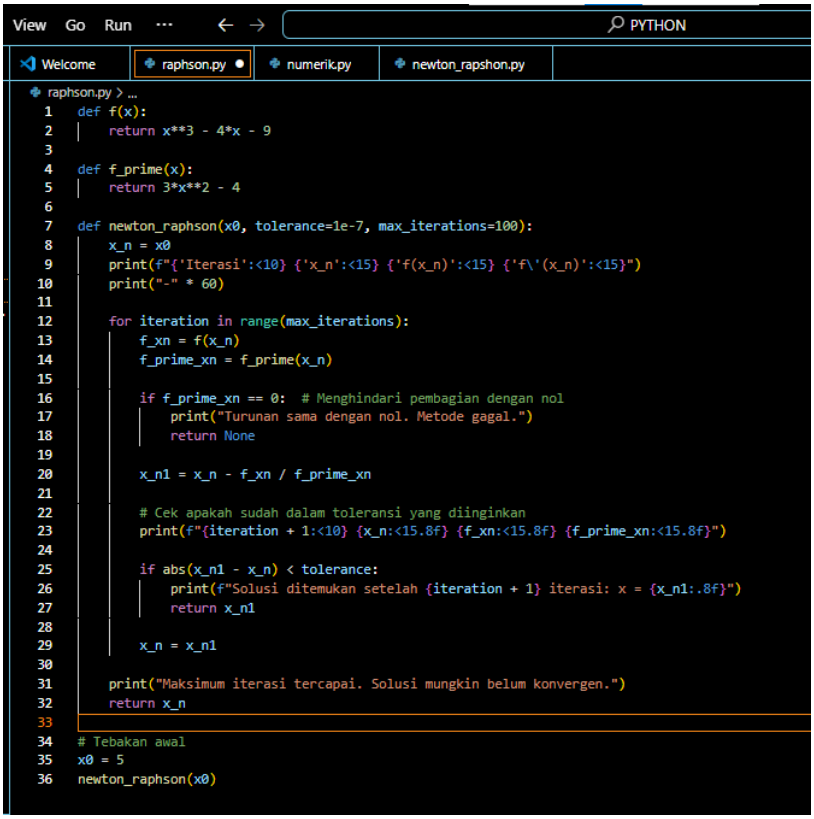
Metode *Newton Raphson*

Metode *Newton-Raphson* adalah suatu teknik pendekatan yang memanfaatkan satu titik awal dan mendekatinya dengan menggunakan gradien di titik tersebut. Proses ini dimulai dengan mencari garis singgung pada kurva di titik $(x_i, f(x_i))$ yang ditentukan. Titik perpotongan antara garis singgung dan sumbu x yaitu x_{i+1} akan menjadi nilai x baru Prosedur ini diulang terus-menerus hingga diperoleh akar dari persamaan (Dwi Estuningsih et al., 2019). Metode *Newton-Raphson* adalah metode numerik yang digunakan untuk mencari akar-akar persamaan non-linear. Metode ini efektif untuk memecahkan masalah persamaan non-linear.

Langkah-langkah metode *Newton Raphson*: 1) Pilih nilai awal x_0 sebagai tebakan awal. 2) Hitung turunan pertama $f'(x)$. 3) Gunakan rumus iterasi *Newton-Raphson*: $x_n + 1 = x_n - \frac{f(x_n)}{f\prime(x_n)}$. 4) Terapkan rumus tersebut secara berulang hingga mendapatkan nilai yang mendekati akar persamaan.

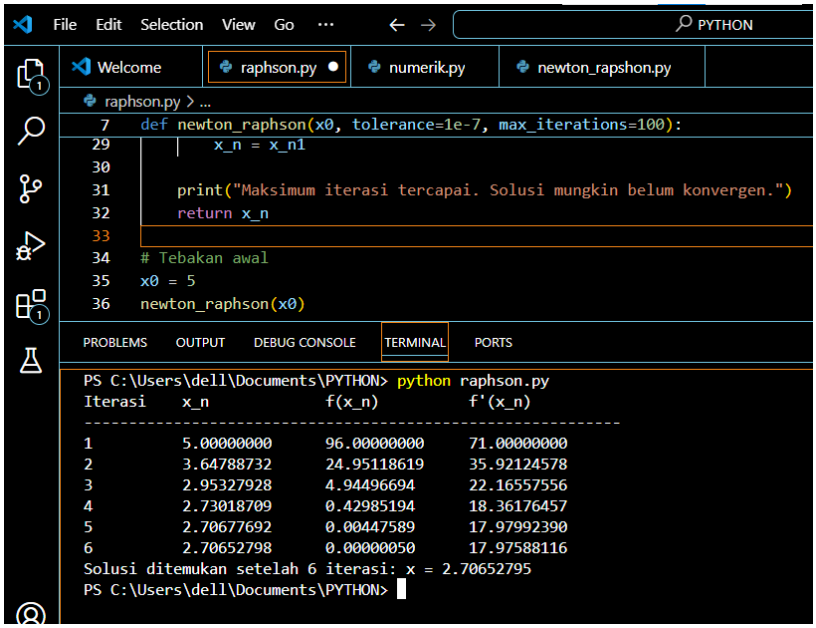
Penyelesaian Metode *Newton Raphson* Menggunakan *Python*

Metode *Newton-Raphson* digunakan untuk mencari akar persamaan non-linier dengan pendekatan iteratif. Dalam penelitian ini, metode *Newton-Raphson* diterapkan untuk menyelesaikan persamaan $f(x) = x^3 - 4x - 9 = 0$ dengan turunan pertama $f'(x) = 3x^2 - 4$. Proses iterasi dilakukan menggunakan *Python* dengan tebakan awal $x_0 = 5$. Berikut (Gambar 1) merupakan langkah-langkah menyelesaikan persamaaan non-linear metode *Newton-Raphson* menggunakan *Python*:



Gambar 1. Program Metode *Newton Rapshon* dengan *Python*

Setelah menuliskan skrip diatas dengan benar, kita dapat mengetahui hasil yang kita inginkan dengan memunculkan terminal. Klik "Terminal – New Terminal" maka akan otomatis memunculkan hasil perhitungan (Gambar 2).



Gambar 2. Hasil Perhitungan Metode *Newton Raphson* dengan Pyhton

Pada persoalan tersebut, memunculkan hasil perhitungan dengan solusi ditemukan $x_n = 2{,}70652795$ setelah 6 iterasi dengan nilai error 0,00000050.

Penyelesaian Metode *Newton Raphson* Menggunakan Micrososft *Excel*

Metode *Newton-Raphson* digunakan untuk mencari akar persamaan non-linier dengan pendekatan iteratif. Dalam penelitian ini, metode *Newton-Raphson* diterapkan untuk menyelesaikan persamaan $f(x) = x^3 - 4x - 9 = 0$ dengan turunan pertama $f'(x) = 3x^2 - 4$. Proses iterasi dilakukan menggunakan *Microsoft Excel* dengan tebakan awal $x_0 = 5$.

Langkah pertama adalah membuat tabel iterasi yang terdiri dari beberapa kolom, yaitu **Iterasi**, $f(x_n), f'(x_n), x_{n+1}$, **Error**, dan **Status**. Nilai tebakan awal $x_0 = 5$ dimasukkan ke dalam kolom x_n. Kemudian, nilai fungsi $f(x_n)$ dihitung menggunakan rumus *Excel* =B2^3 - 4*B2 - 9, sedangkan nilai turunannya$f'(x_n)$ dihitung dengan rumus =3*B2^2 - 4. Setelah itu, nilai iterasi berikutnya x_{n+1} dihitung dengan rumus *Newton-Raphson*, yaitu =B2 - C2/D2.

Pada iterasi selanjutnya, nilai x_n diperbarui dengan nilai x_{n+1} sebelumnya menggunakan rumus=E2. Perhitungan $f(x_n), f'(x_n)$, dan x_{n+1} diulang untuk iterasi berikutnya dengan menggunakan referensi sel sebelumnya. Kolom **Error** dihitung sebagai selisih nilai x_n antara iterasi sebelumnya dan iterasi saat ini dengan rumus =ABS(B3 - B2). Untuk menentukan apakah iterasi masih perlu dilanjutkan, digunakan rumus =IF(F3<0.0001, "Berhenti", "Lanjut"). Iterasi akan terus dilakukan hingga kolom **Status** menunjukkan "Berhenti", yang menandakan bahwa solusi telah mencapai tingkat akurasi yang cukup tinggi.

Setelah proses iterasi selesai, nilai x_n pada iterasi terakhir dianggap sebagai solusi dari persamaan. Jika nilai $f(x_n)$ pada iterasi terakhir mendekati nol, maka metode *Newton-*

Raphson telah berhasil menemukan akar persamaan dengan tingkat kesalahan yang sangat kecil. Dengan memanfaatkan *Microsoft Excel*, metode ini dapat diterapkan secara efisien tanpa perlu perhitungan manual yang rumit.

Iterasi	x_n	f(x_n)	f'(x_n)	x_(n+1)	Error	Lanjut/Berhenti
0	5	96	71	3,64788732	1,35211268	Lanjut
1	3,64788732	24,95118619	35,92124578	2,95327928	0,69460804	Lanjut
2	2,95327928	4,944966938	22,16557556	2,73018709	0,2230922	Lanjut
3	2,73018709	0,429851938	18,36176457	2,70677692	0,02341016	Lanjut
4	2,70677692	0,004475891	17,9799239	2,70652798	0,00024894	Lanjut
5	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
6	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
7	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
8	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
9	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
10	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
11	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
12	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
13	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti
14	2,70652798	5,03204E-07	17,97588116	2,70652795	2,7993E-08	Berhenti

Gambar 3. Hasil Perhitungan Iterasi Menggunakan *Excel*

Dalam perhitungan menggunakan metode *Newton-Raphson* dengan *Excel*, tujuan utama adalah menemukan akar dari persamaan tak linier $f(x) = x^3 - 4x - 9$ dengan turunan pertama $f'(x) = 3x^2 - 4$. Proses iterasi dimulai dengan tebakan awal $x_0 = 5$, kemudian nilai x_n diperbarui menggunakan rumus *Newton-Raphson* hingga memenuhi kriteria konvergensi, yaitu selisih antara x_{n+1} dan x_n lebih kecil dari 10^{-6}.

Berdasarkan hasil perhitungan di *Excel* (gambar 3), iterasi pertama menghasilkan nilai x_1 yang lebih dekat ke akar sejati. Proses ini berlanjut dengan iterasi berikutnya, di mana setiap langkah semakin memperkecil nilai error hingga mencapai batas toleransi yang telah ditentukan. Dari tabel iterasi yang dihasilkan, dapat dilihat bahwa metode ini memberikan konvergensi yang cepat, karena nilai x_n semakin mendekati akar persamaan dengan hanya beberapa kali iterasi. Pada persoalan tersebut, penyelesaian dihentikan pada akar hampiran sudah ditemukan, yaitu pada iterasi 4 dengan nilai error 0,00024894 dan nilai $x_n = 2{,}70677692$.

Perbandingan Penyelesaian Metode *Newton-Raphson* dengan *Python* dan *Excel*

Berdasarkan penyelesaian yang telah dilakukan, diperoleh perbandingan antara *Python* dan *Microsoft Excel* (tabel 1).

Tabel 1. Perbandingan Program

Aspek	Alat Bantu	Kelebihan	Kekurangan
Akurasi Hasil	***Python***	Tingkat akurasi tinggi dengan menghasilkan solusi yang sangat akurat, dengan nilai akar persamaan x≈2,70652795 setelah 6 iterasi dan nilai error 0,00000050.	Untuk masalah sangat sederhana, waktu kompilasi kode *Python* mungkin lebih lama dibandingkan *Excel*, meskipun akurasinya tetap unggul.

Aspek	Alat Bantu	Kelebihan	Kekurangan
	Microsoft Excel	Tingkat akurasi cukup cepat untuk masalah sederhana mencapai solusi dalam 4 iterasi dengan error 0,00024894 cukup akurat untuk kebutuhan dasar.	Meskipun mendekati akar sejati, nilai error yang dihasilkan (0,00024894) 500x lebih besar dibandingkan *Python*.
Kemudahan Penggunaan	***Python***	*Python* relatif mudah dipelajari bagi pengguna yang sudah memiliki dasar pemrograman. *Syntax*-nya sederhana dan intuitif.	Bagi pengguna yang tidak memiliki latar belakang pemrograman, *Python* mungkin terasa sulit karena memerlukan penulisan kode dari awal.
	Microsoft Excel	*Excel* sangat *user-friendly* dan tidak memerlukan pengetahuan pemrograman. Pengguna dapat menggunakan rumus dan fungsi bawaan untuk menyelesaikan persamaan nonlinear.	Untuk mengimplementasikan Metode *Newton-Raphson* secara manual, pengguna perlu membuat rumus iterasi secara manual, yang bisa rumit dan rentan kesalahan.
Automasi Iterasi	***Python***	*Python* sangat baik dalam mengotomatisasi iterasi. Pengguna dapat dengan mudah menulis loop (perulangan) untuk melakukan iterasi Metode *Newton-Raphson* hingga mencapai konvergensi.	Diperlukan penulisan kode untuk mengimplementasikan iterasi, yang mungkin memakan waktu bagi pemula.
	Microsoft Excel	*Excel* memiliki fitur **Solver** yang dapat melakukan iterasi secara otomatis, tetapi fitur ini terbatas pada masalah yang sederhana.	Untuk mengimplementasikan iterasi Metode *Newton-Raphson* secara manual, pengguna harus menyalin rumus ke banyak sel atau menggunakan IF, yang tidak efisien dan rentan kesalahan.
Kecepatan Perhitungan	***Python***	*Python*, terutama dengan library seperti **NumPy** dan **SciPy**, dapat melakukan perhitungan numerik dengan sangat cepat, bahkan untuk masalah yang kompleks.	Untuk masalah yang sangat besar, *Python* mungkin memerlukan waktu lebih lama dibandingkan bahasa pemrograman yang lebih cepat seperti C++
	Microsoft Excel	*Excel* cukup cepat untuk masalah yang sederhana dan tidak memerlukan banyak iterasi.	Untuk masalah yang kompleks atau memerlukan banyak iterasi, *Excel* cenderung lambat karena keterbatasan dalam menangani perhitungan yang rumit.
Fleksibilitas Perubahan Fungsi	***Python***	*Python* sangat fleksibel. Fungsi dan logika perhitungan dapat dimodifikasi langsung dalam script sesuai kebutuhan.	Diperlukan penulisan ulang script jika fungsi atau metode berubah, yang bisa memakan waktu.

Aspek	Alat Bantu	Kelebihan	Kekurangan
	Microsoft Excel	Pengguna dapat dengan mudah mengubah rumus atau fungsi di *Excel* dengan mengedit sel yang sesuai.	Terbatas pada fungsi bawaan seperti IF. Jika fungsi yang digunakan kompleks, perubahan rumus bisa rumit dan rentan kesalahan, terutama jika melibatkan banyak sel.
Tingkat Kesalahan Manual	***Python***	Setelah kode ditulis dengan benar, *Python* dapat menjalankan perhitungan tanpa kesalahan manual. Kesalahan biasanya terjadi pada tahap penulisan kode, bukan selama eksekusi.	Jika ada kesalahan dalam kode, debugging bisa memakan waktu, terutama bagi pemula.
	Microsoft Excel	*Excel* memiliki fitur error checking yang dapat membantu mengidentifikasi kesalahan dalam rumus.	Kesalahan manual sangat mungkin terjadi, terutama jika pengguna harus menyalin rumus ke banyak sel atau mengatur iterasi secara manual.
Reproduksibilitas dan Skalabilitas	***Python***	Kode *Python* dapat dengan mudah direproduksi dan digunakan kembali untuk masalah yang berbeda. *Python* juga sangat skalabel, sehingga cocok untuk menyelesaikan masalah yang lebih besar atau lebih kompleks.	Diperlukan pengetahuan pemrograman untuk menulis kode yang dapat digunakan kembali (*reusable*).
	Microsoft Excel	File *Excel* dapat dengan mudah dibagikan dan digunakan kembali, terutama jika masalahnya sederhana.	*Excel* kurang skalabel untuk masalah yang besar atau kompleks. Reproduksibilitas juga terbatas karena rumus dan iterasi harus diatur secara manual.
Kompatibilitas dengan *Software* Lain	***Python***	*Python* memiliki kompatibilitas yang sangat baik dengan *software* lain. Pengguna dapat mengintegrasikan *Python* dengan database, tools visualisasi, atau bahkan *software* lain seperti MATLAB.	Diperlukan pengetahuan tambahan untuk mengintegrasikan *Python* dengan *software* lain.
	Microsoft Excel	*Excel* dapat dengan mudah diintegrasikan dengan *software Microsoft* lainnya seperti Word atau PowerPoint. *Excel* juga mendukung impor/ekspor data dari berbagai format file.	Kompatibilitas dengan *software non-Microsoft* atau tools pemrograman lainnya terbatas.

Berdasarkan tabel di atas menunjukkan bahwasannya *Python* memiliki keunggulan dalam hal akurasi hasil, automasi iterasi, kecepatan perhitungan, dan fleksibilitas. Berbantuan library seperti NumPy dan SciPy, *Python* lebih unggul dalam permasalahan yang membutuhkan presisi tinggi, mampu melakukan perhitungan numerik secara cepat dan akurat, bahkan untuk masalah yang kompleks. Selaras dengan penelitian oleh Nadiyah et al. (2024) yang menyatakan bahwa *Python* menggunakan cara pemrograman berorientasi objek dan mudah dipahami karena bahasanya fleksibel, sehingga bisa menjalankan banyak perintah sekaligus dengan cepat.

Selain itu, kode *Python* dapat dengan mudah dimodifikasi dan digunakan kembali untuk persamaan lain serta menjadikannya sangat skalabel. Hal tersebut sejalan dengan penelitian oleh Khairudin (2024) yang mengemukakan bahwasannya *Python* dianggap sebagai bahasa pemrograman yang memiliki kode-kode pemrograman yang sangat jelas dan mudah untuk dipahami. Namun, kelemahan utama *Python* terletak pada kebutuhan akan pengetahuan pemrograman, yang mungkin menjadi hambatan bagi pengguna pemula.

Di sisi lain, *Microsoft Excel* lebih unggul dalam kemudahan penggunaan karena antarmukanya yang intuitif dan fitur rumus bawaan. Pengguna tanpa latar belakang pemrograman dapat dengan mudah memasukkan persamaan dan melakukan iterasi secara manual. Namun, *Excel* memiliki keterbatasan dalam menangani masalah yang memerlukan banyak iterasi atau dataset besar. Proses iterasi yang dilakukan secara manual juga rentan terhadap kesalahan dan kurang efisien dibandingkan automasi yang ditawarkan oleh *Python*. *Excel* cocok untuk perhitungan sederhana dengan toleransi error lebih rendah.

Ditinjau dari segi hasil, *Python* memberikan solusi yang mendekati akar sejati, yaitu $x \approx 2{,}70652795$. Sedangkan *Excel* memberikan Solusi mendekati akar sejati yaitu $x \approx 2{,}70677692$. *Python* lebih konsisten karena minim kesalahan manual, sedangkan *Excel* memerlukan ketelitian dalam pengaturan rumus dan sel. Kecepatan perhitungan *Python* juga lebih unggul, terutama untuk iterasi dalam jumlah besar. Pemilihan antara *Python* dan *Excel* tergantung pada kebutuhan pengguna. *Python* lebih cocok untuk masalah kompleks dan pengguna yang familiar dengan pemrograman, sementara *Excel* menjadi pilihan praktis untuk perhitungan sederhana dan pengguna awam.

Kesimpulan

Penyelesaian persamaan nonlinear tertentu yang diselesaikan dengan Metode *Newton-Raphson* dengan alat bantu *Python* lebih unggul dalam hal akurasi hasil, automasi iterasi, kecepatan perhitungan, fleksibilitas perubahan fungsi, reproduksibilitas, dan skalabilitas. Namun, *Python* memerlukan pengetahuan pemrograman dan kurang *user-friendly* bagi pemula. Sedangkan alat bantu *Microsoft Excel* lebih unggul dalam hal kemudahan penggunaan dan kompatibilitas dengan software Microsoft, tetapi kurang cocok untuk masalah yang kompleks atau memerlukan banyak iterasi. *Microsoft Excel* juga rentan terhadap kesalahan manual dan kurang skalabel. Pilihan antara *Python* dan *Microsoft Excel* tergantung pada kompleksitas masalah, latar belakang pengguna, dan kebutuhan akan fleksibilitas serta otomatisasi. Penelitian ini memberikan wawasan berharga bagi akademisi dan praktisi dalam memilih alat yang tepat untuk menyelesaikan persamaan non-linear sesuai dengan konteks dan kemampuan mereka.

Ucapan Terimakasih

Ucapan terimakasih disampaikan kepada dosen mata kuliah yang telah memberikan dukungan dan dorongan dalam proses penyelesaian artikel ini.

Kontribusi Penulis

NR dan IS berkontribusi dalam pengumpulan data hinggs penyusunan aritkel ini. Sedangkan AR berkontribusi dalam memberikan bimbingan pengolahan data dan keperluan revisi artikel ini.

Daftar Pustaka

Agustina, N. (2018). Kemampuan Pemahaman Konsep Siswa Smp Pada Materi Persamaan Garis Lurus Dalam Pembelajaran Berbasis Apos. *HISTOGRAM: Jurnal Pendidikan Matematika*, *2*(1), 12. https://doi.org/10.31100/histogram.v2i1.34

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*(January). https://doi.org/10.1016/j.chb.2019.106185

Baiduri, B. (2018). Some Methods Used by Mathematics Teachers in Solving Equations. *Journal of Education and Learning (EduLearn*, *12*(3), 340–349. https://doi.org/10.11591/edulearn.v12i3.6605

Batarius, P. (2018). Nilai Awal Pada Metode *Newton-Raphson* Yang Dimodifikasi Dalam Penentuan Akar Persamaan. *Pi: Mathematics Education Journal*, *1*(3), 108-115. https://doi.org/10.21067/pmej.v1i3.2784

Dwi Estuningsih, R., Rosita Program Studi Analisis Kimia, T., No, A. K. A. B. J. P. S., P, B., T, U., B, B., K, & Barat, J. (2019). Perbandingan Metode Biseksi Dan Metode Newton Raphson Dalam Penyelesaian Persamaan Non Linear. *Jurnal Warta Akab*, *43*(2), 21–23. https://jurnal.aka.ac.id/index.php/warta_akab/article/view/125/93

Fajar, A. P., Kodirun, K., Suhar, S., & Arapu, L. (2019). Analisis Kemampuan Pemahaman Konsep Matematis Siswa Kelas VIII SMP Negeri 17 Kendari. *Jurnal Pendidikan Matematika*, *9*(2), 229. https://doi.org/10.36709/jpm.v9i2.5872

Frayudi, F., Bahri, S., & Bakar, N. N. (2019). Menentukan Akar Persamaan Nonlinier Dengan Metode Approksimasi Lingkaran. *Jurnal Matematika UNAND*, *4*(2), 38. https://doi.org/10.25077/jmu.4.2.38-45.2015

Hsu, T. R. (2018). Applied Engineering Analysis. https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=CELEX:32016R0679&from=PT%0Ahttp://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52012PC0011:pt:NOT

Khairudin. (2024). Pengenalan Bahasa Pemrograman Pyhton Untuk Meningkatkan Kompetensi Yatim Piatu RW 01 Kelurahan Panunggan Kec. *Pinang. Praxis: Jurnal Pengabdian Kepada Masyarakat*, *4*(3), 36–43. Retrieved from https://pijarpemikiran.com/index.php/praxis/article/view/716

Kreyszig, E. (2011). Advanced Engineering Mathematics. Laurie Rosatone. www.ieee.org.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology. *Engineering, and Mathematics Education. Journal of Science Education and Technology*, *29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

Maharani, S., & Suprapto, E. (2018). Analisis Numerik Berbasis Grup Investigation Untuk Meningkatkan Kemampuan Berpikir Kritis. In *CV. Ae Media Grafika* (1st ed.). CV. Ae Media Grafika. https://doi.org/10.2307/3718634

Nadiyah, A., I, N. H., & Karim, A. (2024). Penerapan Algoritma K-Means Untuk Clustering Penilaian Layanan Berdasarkan Indeks Kepuasan Mahasiswa Universitas Nurul

Jadid. *Jurnal Advanced Research Informatika*, *2*(2), 23–30. https://doi.org/10.24929/jars.v2i2.3431

Oktavian, R. (2024). *Python* vs Excel untuk Analisis Data. https://mindstem.id/2024/02/06/*Python*-vs-excel-untuk-analisis-data/

Pandia, W., & Sitepu, I. (2021). Penentuan Akar Persamaan Non Linier Dengan Metode Numerik. *Jurnal Mutiara Pendidikan Indonesia*, *6*(2), 122–129. https://doi.org/10.51544/mutiarapendidik.v6i2.2326

Pandu, Y. K. (2019). Penerapan Integral Numerik Dalam Menghitung Luas Daerah Tidak Beraturan. *Asimtot : Jurnal Kependidikan Matematika*, *1*(2), 127–132. https://doi.org/10.30822/asimtot.v1i2.278

Rosidi, M. (2019). Metode Numerik Menggunakan R Untuk Teknik Lingkungan. https://bookdown.org/moh_rosidi2610/Metode_Numerik/

Rozi, S., & Rarasati, N. (2022). Template Metode Numerik Pada Excel Untuk Menemukan Solusi Dari Persamaan Nonlinier. *AXIOM : Jurnal Pendidikan Dan Matematika*, *11*(1), 33. https://doi.org/10.30821/axiom.v11i1.11254

Saidin, N. D., Khalid, F., Martin, R., Kuppusamy, Y., & Munusamy, N. A. P. (2021). Benefits and challenges of applying computational thinking in education. *International Journal of Information and Education Technology*, *11*(5), 248–254. https://doi.org/10.18178/ijiet.2021.11.5.1519

Setiawan, A. (2007). *Pengantar Metode Numerik*. CV. Andi Offset.

Siagan, M. V., Saragih, S., & Sinaga, B. (2019). Development of Learning Materials Oriented on Problem-Based Learning Model to Improve Students' Mathematical Problem Solving Ability and Metacognition Ability. *International Electronic Journal of Mathematics Education*, *14*(2), 331–340. https://doi.org/10.12973/EU-JER.11.4.1947

Sumartini, T. S. (2016). Kemampuan Pemecahan Masalah Matematika dalam Menyelesaikan Soal Cerita Berdasarkan Langkah Polya. *Jurnal Pendidikan Matematika STKIP Garut*, *5*(2), 1–7. https://scholar.googleusercontent.com/scholar?q=cache:jfDgJQUQWmcJ:scholar.google.com/+Peningkatan+Kemampuan+Pemecahan+Masalah+Matematis+Siswa+melalui+Pembelajaran+Berbasis+Masalah&hl=id&as_sdt=0,5

Sunandar, E., & Indrianto, I. (2020). Perbandingan Metode *Newton-Raphson* & Metode Secant Untuk Mencari Akar Persamaan Dalam Sistem Persamaan Non-Linier. *Petir*, *13*(1), 72–79. https://doi.org/10.33322/petir.v13i1.893

Swaid, S. I. (2015). Bringing Computational Thinking to STEM Education. *Procedia Manufacturing*, *3*(July 2015), 3657–3662. https://doi.org/10.1016/j.promfg.2015.07.761

Taufik, M., & Susanti, R. D. (2024). Solving numerical method problems with mathematical software: Identifying computational thinking. *Pedagogical Research*, *9*(3), 0209. https://doi.org/10.29333/pr/14583

Yeh, C. Y. C., Cheng, H. N. H., Chen, Z.-H., Liao, C. C. Y., & Chan, T.-W. (2019). Enhancing achievement and interest in mathematics learning through Math-Island. *Research and Practice in Technology Enhanced Learning*, *14*(1). https://doi.org/10.1186/s41039-019-0100-9