# Tugas Pemrograman B KomNum

Kelompok 8 Komputasi Numerik Teknik Komputer TA 2024/2025

1st Nelson Laurensius
NPM: 2306161845

2nd Christian Hadiwijaya
NPM: 2306161952

3rd Benedict Aurelius
NPM: 2306209095

4th Jonathan Frederick Kosasih
NPM: 2306225981

***Abstract*—Studi kasus ini menyajikan implementasi Metode Newton-Raphson, sebuah metode numerik yang efisien untuk menemukan akar-akar (nol) dari suatu fungsi bernilai ril. Fokus utama adalah pada penerapan metode ini untuk mencari akar dari fungsi non linier sederhana, yaitu $f(x) = e^{-x} - x$. Program implementasi dalam bahasa C disertakan dan dianalisis. Hasil eksperimen menunjukkan bahwa Metode Newton-Raphson mampu menemukan akar-akar fungsi ini dengan cepat dan akurat, asalkan tebakan awal dipilih dengan tepat dan kondisi konvergensi terpenuhi. Laporan ini juga membahas dasar teori metode, langkah-langkah implementasi, serta analisis hasil yang diperoleh dibandingkan dengan solusi analitik.**

## I. Pendahuluan

Mencari akar dari suatu persamaan atau fungsi merupakan masalah fundamental dalam berbagai bidang ilmu pengetahuan dan teknik. Seringkali, persamaan yang dihadapi bersifat non-linear atau kompleks sehingga sulit atau bahkan tidak mungkin diselesaikan secara analitik. Dalam kasus seperti ini, metode numerik menjadi alternatif yang sangat berguna untuk mendapatkan solusi perkiraan yang akurat.

Salah satu metode numerik yang paling populer dan efisien untuk menemukan akar fungsi adalah Metode Newton-Raphson. Metode ini didasarkan pada gagasan untuk melakukan iterasi menggunakan garis singgung kurva fungsi pada suatu titik tebakan awal untuk mendekati akar yang sebenarnya.

Studi kasus ini bertujuan untuk mengimplementasikan dan menganalisis kinerja Metode Newton-Raphson dalam menemukan akar-akar dari fungsi sederhana $f(x) = e^{-x} - x$. Fungsi ini dipilih karena akarnya dapat diketahui secara analitik, sehingga hasil numerik dapat diverifikasi dengan mudah. Melalui studi kasus ini, diharapkan dapat dipahami prinsip kerja, kelebihan, dan potensi keterbatasan dari Metode Newton-Raphson.

## II. Studi Literatur

### A. Metode Newton-Raphson

Metode Newton-Raphson adalah sebuah algoritma iteratif untuk mencari akar (atau nol) dari sebuah fungsi $f(x)$ yang terdiferensiasi. Metode ini menggunakan pendekatan linear (garis singgung) terhadap fungsi pada suatu titik perkiraan untuk menemukan perkiraan akar yang lebih baik.

Secara matematis, jika $x_n$ adalah perkiraan akar pada iterasi ke-$n$, maka perkiraan akar berikutnya, $x_{n+1}$, diberikan oleh rumus iterasi:

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

dengan $f'(x_n)$ adalah turunan pertama dari fungsi $f(x)$ dievaluasi pada $x_n$. Iterasi ini diulang hingga kriteria penghentian tertentu terpenuhi, seperti nilai $|f(x_{n+1})|$ kurang dari toleransi yang ditentukan atau perubahan antara dua iterasi $|x_{n+1} - x_n|$ cukup kecil.

### B. Konvergensi Metode Newton-Raphson

Konvergensi Metode Newton-Raphson sangat bergantung pada tebakan awal dan sifat fungsi di sekitar akar. Jika tebakan awal cukup dekat dengan akar dan fungsi memenuhi kondisi tertentu (misalnya, turunan pertama tidak nol di sekitar akar), metode ini umumnya konvergen secara kuadratik, yang berarti jumlah digit signifikan yang benar berlipat ganda pada setiap iterasi. Namun, jika tebakan awal buruk atau turunan pertama mendekati nol di dekat akar, metode ini mungkin konvergen secara lambat, divergen, atau berosilasi.

## III. Penjelasan Data Yang Digunakan

Implementasi Metode Newton-Raphson dalam program C dirancang untuk menemukan akar dari suatu fungsi matematika yang telah di *hard code* di *source code*. Data utama yang digunakan dan diproses oleh program ini meliputi:

- **Fungsi Target (f(x))**: Fungsi matematis yang akarnya ingin dicari. Dalam kode *NewtonRaphson.c* fungsi yang diimplementasikan secara eksplisit adalah: $f(x) = e^{-x} - x$. Fungsi ini merupakan persamaan transendental yang tidak mudah diselesaikan secara analitik, sehingga cocok untuk dipecahkan menggunakan metode numerik seperti Newton-Raphson.
- **Turunan Fungsi (f'(x)):**: Metode Newton-Raphson memerlukan turunan pertama dari fungsi target untuk menghitung langkah iterasi berikutnya. Sesuai dengan fungsi target di atas, turunan analitiknya juga diimplementasikan dalam kode: $f'(x) = -e^{-x} - 1$.
- **Input Pengguna**: Program ini bersifat interaktif dan memerlukan beberapa parameter input dari pengguna untuk mengontrol proses pencarian akar:
  - **Tebakan Awal (Initial Guess)**: Nilai $x$ awal ($x_0$) yang digunakan untuk memulai proses iterasi. Pemilihan tebakan awal dapat mempengaruhi kecepatan konvergensi dan bahkan apakah metode akan konvergen ke akar yang diinginkan.
  - **Toleransi Error (Tolerance)**: Kriteria penghentian iterasi berdasarkan tingkat keakuratan yang diinginkan. Iterasi berhenti jika nilai absolut fungsi

$|f(x_{i+1})|$ atau perubahan relatif antara iterasi $|(x_{i+1} - x_i)/x_{i+1}|$ (atau $|x_{i+1} - x_i|$ jika $x_{i+1}$ mendekati nol) lebih kecil dari nilai toleransi yang ditentukan pengguna.
  - **Jumlah Iterasi Maksimum (Maximum Iterations)**: Batas atas jumlah iterasi yang diizinkan. Parameter ini berfungsi sebagai pengaman untuk mencegah *infinite loop* jika metode gagal konvergen karena alasan tertentu (misalnya, tebakan awal yang buruk atau perilaku fungsi yang menyulitkan).
- **Data Output**: Selama eksekusi, program menghasilkan data output yang menunjukkan proses iterasi dan hasil akhir. Ini termasuk tabel yang merinci nilai $x_i$, $f(x_i)$, $f'(x_i)$, $x_{i+1}$, dan *error* relatif perkiraan untuk setiap iterasi, serta nilai akar perkiraan akhir dan jumlah iterasi yang diperlukan untuk mencapainya.

## IV. Penjelasan Metode Yang Digunakan

Dalam program, algoritma inti diwujudkan dalam fungsi `metode_newton_raphson`, yang menerima tebakan awal, toleransi error, dan jumlah iterasi maksimum sebagai parameter, serta sebuah pointer untuk mengembalikan nilai akar yang ditemukan.

Langkah-langkah utama implementasi metode dalam kode adalah sebagai berikut:

1) **Inisialisasi:** Fungsi `metode_newton_raphson` dimulai dengan menginisialisasi variabel-variabel yang diperlukan, termasuk $x_0$ (tebakan awal), $x_1$ (tebakan berikutnya), $f(x_0)$ (nilai fungsi di $x_0$), $f'(x_0)$ (nilai turunan di $x_0$), dan penghitung iterasi.
2) **Definisi Fungsi dan Turunan:** Dua fungsi terpisah, `fungsi(double x)` dan `turunan_fungsi(double x)`, mendefinisikan fungsi target $f(x)$ dan turunan pertamanya $f'(x)$.
3) **Input Pengguna:** Fungsi `main` bertanggung jawab untuk berinteraksi dengan pengguna, meminta *input* untuk tebakan awal (`tebakan_awal_input`), toleransi error (`toleransi_input`), dan jumlah iterasi maksimum (`maks_iterasi_input`) menggunakan `scanf`.
4) **Loop Iterasi:** Inti dari metode diimplementasikan dalam `loop do-while` di dalam fungsi `metode_newton_raphson`. Loop ini terus berjalan selama kriteria konvergensi belum terpenuhi dan jumlah iterasi belum mencapai batas maksimum.
5) **Perhitungan Iterasi:** Dalam setiap iterasi:
   - Nilai fungsi $f(x_0)$ dan turunan $f'(x_0)$ dihitung.
   - Dilakukan pengecekan untuk menghindari pembagian dengan nol atau angka sangat kecil. Jika $|f'(x_0)| < 10^{-9}$, program mencetak pesan error dan keluar.
   - Nilai tebakan berikutnya $x_1$ dihitung menggunakan rumus Newton-Raphson:
   $$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$
   - Error relatif dihitung sebagai:
   $$\text{error} = \begin{cases} |x_1 - x_0| & \text{jika } |x_1| < 10^{-9} \\ \left|\frac{x_1 - x_0}{x_1}\right| & \text{lainnya} \end{cases}$$
   - Informasi iterasi (nomor, $x_i$, $f(x_i)$, $f'(x_i)$, $x_{i+1}$, dan error relatif) di *print out* ke *console*.
   - Nilai $x_0$ diperbarui menjadi $x_1$ untuk iterasi berikutnya.
   - Penghitung iterasi ditingkatkan.
6) **Pengecekan Batas Iterasi:** Jika jumlah iterasi mencapai batas maksimum, program akan mengirim peringatan dan akan menyimpan nilai terakhir sebagai jawaban.
7) **Kriteria Konvergensi:** Loop dihentikan jika:
   $$|f(x_1)| < \text{toleransi} \quad \text{dan} \quad \text{error relatif} < \text{toleransi}$$
8) **Output Hasil:** Setelah loop berhenti, fungsi `main` mencetak hasil akhir, termasuk:
   - Jumlah iterasi yang dilakukan.
   - Nilai akar perkiraan yang ditemukan.
   - Nilai fungsi pada akar tersebut.

   Jika metode gagal konvergen, program mencetak pesan kegagalan.

## V. Diskusi dan Analisa Hasil Experimen

Berdasarkan hasil eksekusi program implementasi Metode Newton-Raphson untuk mencari akar dari fungsi $f(x) = e^{-x} - x$, didapatkan data sebagai berikut:

TABLE I
Tabel Iterasi Metode Newton-Raphson

| Iterasi | $x_i$ | $f(x_i)$ | $f'(x_i)$ | $x_{i+1}$ | $\frac{\lvert x_{i+1}-x_i\rvert}{\lvert x_{i+1}\rvert}$ |
|---|---|---|---|---|---|
| 0 | 0.000000 | 1.000000 | -2.000000 | 0.500000 | 1.000000e+00 |
| 1 | 0.500000 | 0.106531 | -1.606531 | 0.566311 | 1.170929e-01 |
| 2 | 0.566311 | 0.001305 | -1.567616 | 0.567143 | 1.467287e-03 |
| 3 | 0.567143 | 0.000000 | -1.567143 | 0.567143 | 2.210639e-07 |

### A. Interpretasi Hasil Iterasi

Hasil eksperimen menunjukkan bahwa Metode Newton-Raphson berhasil menemukan akar fungsi $f(x) = e^{-x} - x$ dengan cepat dan akurat. Berdasarkan tabel iterasi yang dihasilkan oleh program, proses konvergensi dapat dianalisis sebagai berikut:

- **Iterasi ke-0**
  Tebakan awal $x_0 = 0.000000$. Nilai fungsi pada tebakan awal adalah $f(x_0) = 1.000000$, dan turunan pertama adalah $f'(x_0) = -2.000000$. Dari rumus Newton-Raphson, tebakan berikutnya dihitung menjadi $x_1 = 0.500000$. Error relatif pada iterasi ini adalah $1.000000$.
- **Iterasi ke-1**
  Tebakan baru $x_1 = 0.500000$. Nilai fungsi dan turunan pertama berturut-turut adalah $f(x_1) = 0.106531$ dan $f'(x_1) = -1.606531$. Tebakan berikutnya dihitung menjadi $x_2 = 0.563111$. Error relatif turun menjadi $0.117093$.

- **Iterasi ke-2**
  Tebakan baru $x_2 = 0.563111$. Nilai fungsi dan turunan pertama berturut-turut adalah $f(x_2) = 0.001305$ dan $f'(x_2) = -1.567616$. Tebakan berikutnya dihitung menjadi $x_3 = 0.567143$. Error relatif turun secara signifikan menjadi $0.001467$.
- **Iterasi ke-3**
  Tebakan baru $x_3 = 0.567143$. Nilai fungsi dan turunan pertama berturut-turut adalah $f(x_3) = 0.000000$ dan $f'(x_3) = -1.567143$. Pada iterasi ini, nilai fungsi hampir nol $f(x_3) \approx 0$, dan error relatif sangat kecil ($2.210639 \times 10^{-7}$). Karena nilai fungsi dan error relatif telah mencapai toleransi yang ditentukan, metode berhenti pada iterasi ini.

### B. Kecepatan Konvergensi

Metode Newton-Raphson menunjukkan konvergensi kuadratik, seperti yang diharapkan. Hal ini terlihat dari penurunan nilai error relatif yang sangat cepat pada setiap iterasi:

- **Iterasi ke-0**: Error relatif $= 1.000000$
- **Iterasi ke-1**: Error relatif $= 0.117093$
- **Iterasi ke-2**: Error relatif $= 0.001467$
- **Iterasi ke-3**: Error relatif $2.210639 \times 10^{-7}$

Penurunan ini menunjukkan bahwa jumlah digit signifikan yang benar dalam perkiraan akar berlipat ganda pada setiap iterasi, sesuai dengan karakteristik konvergensi kuadratik.

Akar yang ditemukan oleh program adalah $x \approx 0.567143$ dengan nilai fungsi pada akar tersebut $f(x) \approx 0$. Hasil ini valid karena memenuhi kriteria konvergensi.

Tebakan awal $x_0 = 0$ ternyata cukup dekat dengan akar sebenarnya, sehingga metode Newton-Raphson dapat mencapai konvergensi dengan cepat. Jika tebakan awal jauh dari akar sebenarnya, metode mungkin mengalami konvergensi lambat atau bahkan divergen. Oleh karena itu, pemilihan tebakan awal yang tepat sangat penting untuk efisiensi metode.

Meskipun Metode Newton-Raphson memiliki kecepatan konvergensi yang tinggi, ada beberapa keterbatasan yang perlu diperhatikan:

- **Kebutuhan Turunan Pertama**: Metode ini memerlukan turunan pertama fungsi yang harus diketahui secara analitik. Jika turunan sulit dihitung atau tidak tersedia, metode alternatif seperti Metode Secant mungkin lebih cocok.
- **Sensitivitas Terhadap Tebakan Awal**: Tebakan awal yang buruk dapat menyebabkan metode divergen atau konvergen ke akar yang salah.
- **Masalah dengan Turunan Nol**: Jika turunan mendekati nol di sekitar akar, metode dapat mengalami ketidakstabilan numerik atau divergensi.

## VI. Kesimpulan

Dalam studi kasus ini, Metode Newton-Raphson berhasil diimplementasikan dalam program C untuk mencari akar fungsi non-linear $f(x) = e^{-x} - x$. Hasil eksperimen menunjukkan bahwa metode ini memiliki kecepatan konvergensi yang sangat baik, dengan hanya membutuhkan 4 iterasi untuk mencapai akurasi yang diinginkan (toleransi error $= 1 \times 10^8$).

Beberapa poin penting yang dapat disimpulkan adalah:

- **Efisiensi Konvergensi**: Metode Newton-Raphson menunjukkan konvergensi kuadratik, yang ditandai dengan penurunan nilai error relatif secara eksponensial pada setiap iterasi.
- **Pengaruh Tebakan Awal**: Pemilihan tebakan awal yang tepat sangat penting untuk keberhasilan metode. Dalam kasus ini, tebakan awal $x_0 = 0$ cukup dekat dengan akar sebenarnya, sehingga metode dapat mencapai konvergensi dengan cepat.
- Validasi Hasil : Akar yang ditemukan $x \approx 0.567143$ valid, karena nilai fungsi pada akar tersebut hampir nol ($f(x) \approx 0$) dan memenuhi kriteria konvergensi.
- **Keterbatasan**: Meskipun metode ini sangat efisien, ada beberapa keterbatasan, termasuk kebutuhan turunan pertama, sensitivitas terhadap tebakan awal, dan potensi divergensi jika turunan mendekati nol.

Secara keseluruhan, implementasi Metode Newton-Raphson dalam program ini memberikan pemahaman yang komprehensif tentang cara kerja metode ini dalam menyelesaikan masalah pencarian akar fungsi non-linear. Metode ini sangat berguna dalam aplikasi praktis, tetapi pengguna harus mempertimbangkan keterbatasannya agar dapat digunakan dengan efektif.

## VII. Link Github

Kode sumber lengkap untuk implementasi Metode Newton-Raphson yang dianalisis dalam laporan ini, beserta file pendukung lainnya, tersedia di repositori GitHub berikut: https://github.com/nlsnlaurensius/TugasPemrogramanB_Kelompok_8

## References


[1] S. C. Chapra and R. P. Canale, Numerical Methods for Engineers , 6th ed., McGraw-Hill Education, 2010.

[2] W. F. Tinney and C. E. Hart, ”Power Flow Solution by Newton’s Method,” IEEE Transactions on Power Apparatus and Systems, vol. PAS-86, no. 11, pp. 1449–1460, Nov. 1967.

[3] S. W. Ng and Y. S. Lee, ”Variable dimension Newton-Raphson method,” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, no. 6, pp. 809–817, Jun. 2000.

[4] J. Verbeke and R. Cools, ”The Newton-Raphson method,” International Journal of Mathematical Education in Science and Technology, vol. 26, no. 2, pp. 177–193, Mar. 1995.

[5] V. V. Mehtre, ”Root Finding Methods: Newton Raphson Method,” International Journal for Research in Applied Science and Engineering Technology, vol. 7, no. 11, pp. 411–414, Nov. 2019.

[6] J. H. He, ”A modified Newton-Raphson method,” Communications in Numerical Methods in Engineering, vol. 20, no. 10, pp. 801–805, Jun. 2004.

[7] F. Casella and B. Bachmann, ”On the choice of initial guesses for the Newton-Raphson algorithm,” arXiv preprint arXiv:1911.12433, Nov. 2019.