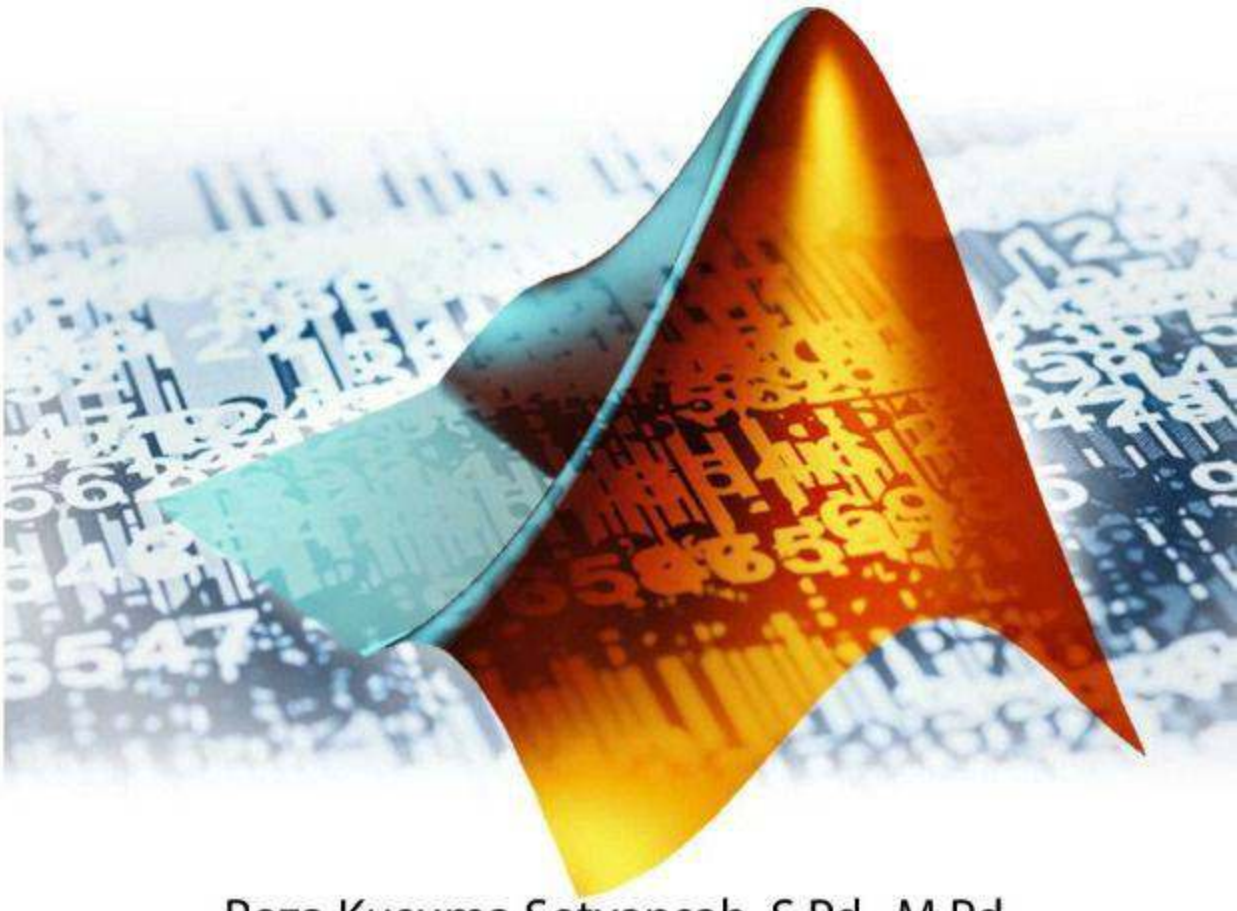
Analisis Numerik Pemograman

MATLAB

berbasis Simulasi e-Learning



Reza Kusuma Setyansah, S.Pd., M.Pd.
Davi Apriandi, S.Pd.Si., M.Pd.

LINGKUP PENDIDIKAN MATEMATIKA

ANALISIS NUMERIK

Pemograman MATLAB Berbasis Simulasi E-Learning

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

ANALISIS NUMERIK

Pemograman MATLAB Berbasis Simulasi E-Learning

Reza Kusuma Setyansah, S.Pd., M.Pd.
Davi Apriandi, S.Pd.Si., M.Pd.

deepublish
glorify and develop the intellectual of human's life

ANALISIS NUMERIK
PEMOGRAMAN MATLAB BERBASIS SIMULASI E-LEARNING

Reza Kusuma Setyansah
Davi Apriandi

Desain Cover : Nama
Tata Letak Isi : Nurul Farma Subekti
Sumber Gambar: Sumber

Cetakan Pertama: Juli 2018

PENERBIT DEEPUBLISH
(Grup Penerbitan CV BUDI UTAMA)
Anggota IKAPI (076/DIY/2012)

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id

Katalog Dalam Terbitan (KDT)

SETYANSAH, Reza Kusuma
Analisis Numerik: Pemograman Matlab Berbasis Simulasi E-Learning/oleh Reza Kusuma Setyansah & Davi Apriandi.--Ed.1, Cet. 1--Yogyakarta: Deepublish, Juli 2018.

x, 202 hlm.; Uk:14x20 cm

ISBN **978-Nomor ISBN**

1. Pemograman I. Judul
005.1

Kata Pengantar

Alhamdulillah telah diberikan kesempatan kali ini, seiring dengan penulisan pengantar yang sederhana ini kami telah menyelesaikan penyusunan buku Analisis Numerik Pemrograman MATLAB berbasis Simulasi E-Learning. Puji dan syukur kita selalu panjatkan ke hadirat Allah SWT karena atas berkat rahmat dan hidayah-Nya kami bisa menyelesaikan penyusunan buku ini.

Penyusunan buku ini terinspirasi dari penelitian yang telah dilaksanakan kami dalam penelitian sebelumnya. Hal ini dilatarbelakangi dari kurangnya pemahaman konseptual dan kemandirian belajar dari peserta didik sehingga sifat dasar pembelajaran dalam komputasi matematika dengan penggunaan MATLAB dalam pembelajaran analisis numerik memerlukan pendampingan berupa tutorial materi. Dalam hal ini, kami mengembangkan buku yang didampingi dengan simulasi e-learning yang diletakkan dalam aplikasi android berisikan tutorial penggunaan pemrograman MATLAB untuk penyelesaian analisis numerik.

Analisis Numerik Pemrograman MATLAB berbasis Simulasi E-Learning merupakan buku pembelajaran yang dirancang untuk membantu menerapkan konsep matematika dan menerapkan konsep komputasi matematis program MATLAB. MATLAB adalah kependekan dari matrix laboratory,

dimana MATLAB merupakan perangkat lunak berbayar untuk komputasi teknis dan saintifik. MATLAB merupakan integrasi komputasi, visualisasi, dan pemrograman yang mudah digunakan. Sehingga MATLAB dapat bertindak sebagai kalkulator dan bahasa pemrograman.

Terima kasih kami ucapkan khususnya kepada DIKTI yang telah memberikan dana hibah sehingga terwujudnya buku ini dan seluruh pihak dari Universitas PGRI Madiun yang telah membantu dalam penyusunan buku ini. Semoga dengan kehadiran buku Analisis Numerik Pemrograman MATLAB berbasis Simulasi E-Learning ini, mampu meningkatkan pemahaman konseptual dan kemandirian belajar komputasi MATLAB dalam peserta didik pada umumnya dan khususnya untuk kami sendiri.

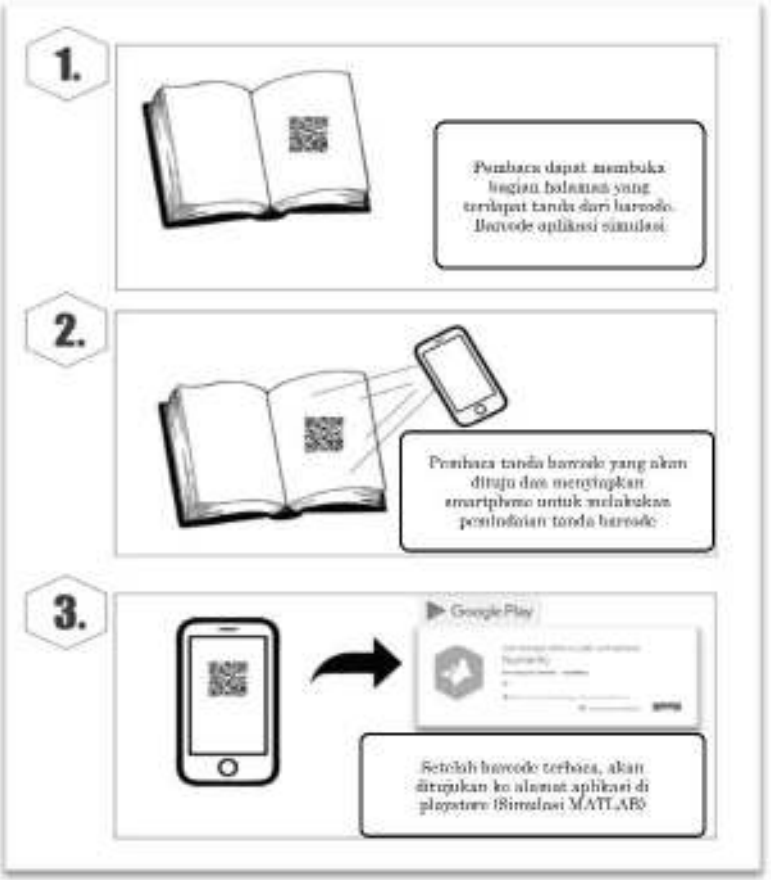
Daftar Isi

Daftar Program Simulasi

- ✓ Pengantar Simulasi MATLAB.
- ✓ Program Simulasi MATLAB Perhitungan Analitik dan Numerik.
- ✓ Program Simulasi MATLAB Galat Pemotongan absolut dan relatif.
- ✓ Program Simulasi MATLAB Galat Pemotongan deret Taylor.
- ✓ Program Simulasi MATLAB Metode Tabel.
- ✓ Program Simulasi MATLAB Metode Biseksi.
- ✓ Program Simulasi MATLAB Metode Interpolasi Linier.
- ✓ Program Simulasi MATLAB Metode Newton Rhapson.
- ✓ Program Simulasi MATLAB Metode Eliminasi Gauss.
- ✓ Program Simulasi MATLAB Metode Iterasi Jacobi.
- ✓ Program Simulasi MATLAB Metode Iterasi Gauss Seidel.
- ✓ Program Simulasi MATLAB Pendekatan Interpolasi Linier.
- ✓ Program Simulasi MATLAB Pendekatan Interpolasi Lagrange.
- ✓ Program Simulasi MATLAB Regresi Linier.
- ✓ Program Simulasi MATLAB Regresi Polinomial.
- ✓ Program Simulasi MATLAB Metode Integrasi Trapezium.
- ✓ Program Simulasi MATLAB Metode Integrasi Simpson.

Mekanisme Simulasi E-Learning

Perhatikan langkah dari mekanisme penggunaan untuk memperoleh aplikasi simulasi e-learning di playstore.



Aplikasi untuk melakukan pemindaian barcode dapat dicari melalui laman

https://play.google.com/store/search? Dengan menggunakan kata pencarian **qr code reader**. Pembaca dapat memilih aplikasi gratis atau berbayar sesuai dengan kebutuhan yang diperlukan, seperti yang tertampil pada gambar di bawah ini.

Link url menuju Aplikasi di Playstore: https://play.google.com/store/apps/details?id=com.wHome_7439063 atau https://goo.gl/NXDQvD

Pindai Barcode Link menuju Aplikasi Simulasi MATLAN (Analisis Numerik) di Playstore:

SIMULASI 1: PENDAHULUAN

1.1. Pengantar MATLAB

Bahasa pemrograman sebagai perangkat untuk berinteraksi antara manusia dengan komputer era dewasa ini dibuat agar semakin mudah dan cepat. Sebagai contoh, dapat dilihat dari perkembangan bahasa pemrograman Pascal, yang terus bermunculan varian baru hingga akhirnya menjadi Delphi, demikian pula dengan basic dengan Visual basic-nya serta C dengan C++ Builder-nya. Pada akhirnya semua bahasa pemrograman akan semakin memanjakan pemakainya dengan penambahan fungsi-fungsi barunya yang sangat mudah dan sederhana digunakan bahkan oleh tingkat pemula.

MATLAB merupakan perangkat lunak yang dipergunakan untuk melakukan pemrograman analisis, serta komputasi teknis dan matematis berbasis matriks. MATLAB adalah singkatan dari *Matrix Laboratory* karena memiliki kemampuan menyelesaikan permasalahan perhitungan dalam bentuk matriks. MATLAB versi pertama dirilis pada tahun 1970 oleh Cleve Moler. Pada awalnya, MATLAB didesain untuk menyelesaikan masalah-masalah persamaan aljabar linier. Seiring perkembangan waktu, program ini terus mengalami kemajuan dan peningkatan dari segi fungsi dan performa komputasi.

MATLAB dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan akses data matrik pada proyek LINPACK dan EISPACK. Selanjutnya menjadi sebuah aplikasi untuk menjalankan komputasi matriks. Dari sejak penggunaan awalnya, MATLAB memperoleh masukkan dari ribuan pemakai melalui laman https://www.mathworks.com. Dalam lingkungan dunia pendidikan ilmiah menjadi alat pemrograman standar bidang Matematika, Rekayasa dan Keilmuan terkait. Bahkan, perkembangan saat ini lingkungan dunia industri dapat menjadi pilihan paling produktif untuk perkembangan riset, pengembangan dan analisis.

MATLAB merupakan aplikasi program komputer yang mampu membantu memecahkan berbagai masalah matematis yang kerap kita temui dalam bidang perhitungan dan teknis. Kita bisa memanfaatkan kemampuan MATLAB untuk menemukan solusi dari berbagai masalah numerik. Secara singkatnya Pemahaman terhadap MATLAB, mulai hal yang paling dasar, misalkan sistem dua persamaan dengan dua variabel:

$x - 2y = 32$

$12x + 5y = 12$

hingga yang kompleks, seperti mencari akar-akar polinomial, interpolasi dari sejumlah data, perhitungan dengan matriks, pengolahan sinyal, dan metode numerik.

Salah satu aspek yang sangat berguna dari MATLAB ialah kemampuannya untuk menggambarkan berbagai jenis grafik, sehingga kita bisa mem-visualisasikan data dan fungsi yang kompleks.

Dalam buku ini kita akan mempelajari MATLAB setahap demi setahap langkah dan teori perhitungan analisis numerik, mulai dari hal yang sederhana hingga yang cukup kompleks. Hal yang harus dipersiapkan untuk belajar perhitungan analisis numerik dengan pemrograman MATLAB ialah seperangkat komputer yang sudah terinstal program MATLAB di dalamnya. Kita bisa gunakan MATLAB versi 5, 6 ataupun 7 untuk versi lama, untuk versi terbaru dapat diunduh pada laman https://www.mathworks.com untuk MATLAB R2017b. Bentuk untuk mempraktekkan berbagai contoh yang ada di buku ini. Di dalam buku ini kita akan mempelajari 'teori' penggunaan MATLAB, namun dalam materi ini menerapkan pemahaman dalam perhitungan Analisis Numerik.

Analisis Numerik adalah teknik-teknik yang dipergunakan untuk menyelesaikan masalah-masalah matematika sehingga masalah matematika tersebut dapat diselesaikan oleh pengoperasian aritmetika. Tujuan dari penyusunan buku ini adalah memudahkan pembaca memahami dan mempelajari langkah-langkah perhitungan analisis numerik melalui cakupan melihat praktik penggunaan simulasi MATLAB melalui media e-learning dan memahaminya dengan membaca buku ini.

Untuk lebih memahami dasar-dasar dari penggunaan pemrograman MATLAB, maka dalam penyusunan buku ini membahas praktik-praktik simulasi MATLAB. Bentuk praktik-praktik simulasi yang mencakup materi analisis numerik yang berisikan tentang penyelesaian Galat, Akar Persamaan Tak Linier, Sistem Persamaan Linier, Diferensial Numerik, Integrasi Numerik dan Pencocokan Kurva. Adapun untuk memahami simulasi dalam perhitungan MATLAB, penulis memberikan pengantar pendahuluan MATLAB secara ringkas dalam bagian Simulasi Pendahuluan yang diantara berisikan Penerapan Fungsi Utama Perangkat dalam MATLAB, Operator Dasar MATLAB dan Variabel, Operator dan Fungsi Dasar MATLAB.

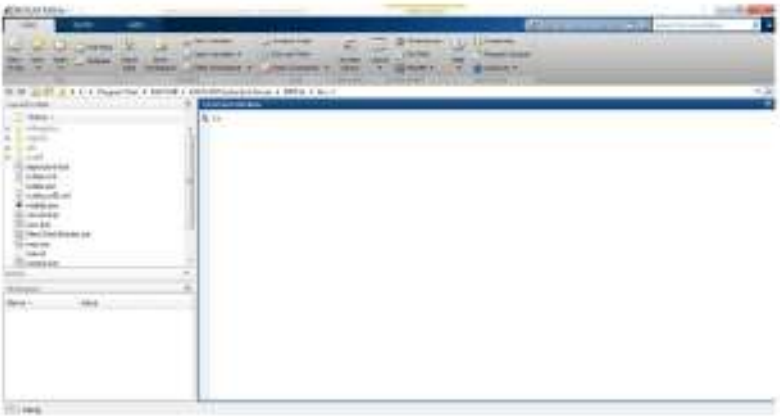
1.2. Penerapan Fungsi Utama Perangkat MATLAB

a. Lingkungan Kerja MATLAB

Dalam bahasa pemrograman lainnya, MATLAB juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Pada setiap update versi terbaru MATLAB, bentuk lingkungan yang dimiliki semakin dilengkapi. Lingkungan terpadu ini terdiri dari *form*/*window* yang memiliki fungsi masing-masing. Untuk memulai penggunaan MATLAB, pengguna perlu melakukan klik icon MATLAB pada dekstop window, atau bisa juga dengan menu dalam start seperti pada aplikasi-aplikasi lainnya.

Pertama kali membuka aplikasi MATLAB, anda akan memperoleh beberapa bagian dari *form*/*window*, yang sebenarnya menurut pengguna hanya membuat dekstop terlihat penuh. Anda hanya menutup semua *window* tersebut kecuali dari *command window* yang merupakan *window* utama dari MATLAB. MATLAB menyimpan mode/setting terakhir dari lingkungan kerja yang anda gunakan sebagai mode/setting lingkungan anda saat bekerja pada MATLAB di waktu berikut. Adapun dalam penggunaan pemrograman MATLAB dalam buku ini menggunakan versi R2015a atau disebutkan dalam versi terdahulunya dengan versi 8.5.

Window Utama MATLAB



Gambar. 1.1. Tampilan Window Utama MATLAB.

Window ini adalah window induk dari pemrograman MATLAB. Pada versi terdahulunya,

window ini secara khususnya belum ada, dan hanya ada *command window*. Tidak ada fungsi utama yang ditawarkan oleh *window* ini selain sebagai tempat docking bagi form yang lain.

Menu MATLAB

Gambar. 1.2. Tampilan Menu MATLAB.

Model menu ini, mulai diperkenalkan sejak MATLAB versi R2013 berfungsi sebagai *shortcut* bagi pengguna program untuk memanfaatkan perintah-perintahnya secara umum dalam program MATLAB. Seperti membuat kode program atau file-M baru (*New Script*), menjalankan dan menghitung waktu proses (*Run and Time*), mengatur tata letak form (*Layout*), mengatur konfigurasi umum (*Preferences*) dan mengatur pencarian direktori (*Set Path*)

Menu PLOTS

Gambar. 1.3. Tampilan Menu PLOTS MATLAB.

Menu plots ini disediakan oleh MATLAB untuk memberikan kebutuhan dan kemudahan bagi pengguna

dalam menampilkan dan memvisualisasikan data tanpa perlu menggunakan cara script. Pada simulasi ini kita cukup berkenalan dengan menunya saja, karena dalam pembahasan simulasi menu plot akan dipergunakan pada simulasi yang menampilkan grafik.

Menu APPS

Gambar. 1.4. Tampilan Menu APPS MATLAB.

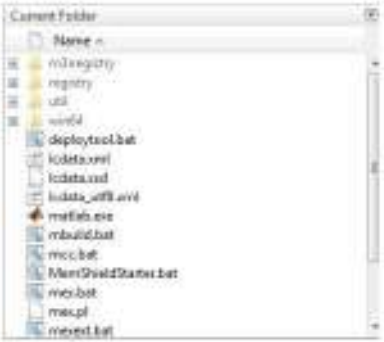
Menu APPS ini adalah bentuk dari kumpulan ikon shortcut aplikasi-aplikasi yang telah anda install pada MATLAB untuk memberikan kemudahan. Pada versi pendahulunya aplikasi ini kita sebut toolbox. Aplikasi yang telah disediakan oleh MATLAB sangatlah beragam, dan pengguna memilihnya saat instalasi MATLAB dilakukan. Perlu diperhatikan, bahwa semakin banyak aplikasi yang dipilih/diinstal maka akan semakin mempengaruhi waktu loading MATLAB anda.

Workspace Window

Gambar. 1.5. Tampilan Workspace Window MATLAB.

Window ini diperkenalkan pada saat versi 6, yang mana berfungsi sebagai navigator bagi pemakai dalam menyediakan informasi terkait dengan variabel yang sedang aktif dalam proses workspace pada saat pemakaian. Workspace adalah ruang lingkup abstrak yang menyimpan seluruh variabel perintah yang dipergunakan selama MATLAB berlangsung.

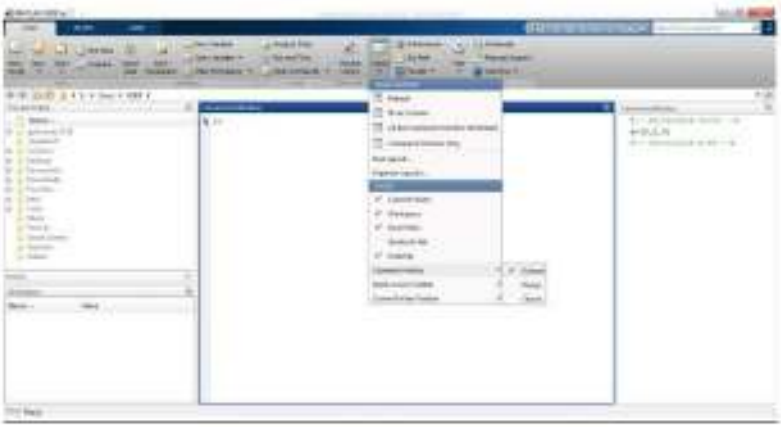
Current Directory Windows



Gambar. 1.6. Tampilan Current Directory MATLAB

Window ini juga fasilitas yang diperkenalkan pada versi 6, berfungsi sebagai browser direktori aktif, yang memiliki fungsi hampir sama dengan window eksplore.

Command History Window.



Gambar. 1.7. Tampilan Command History Window MATLAB

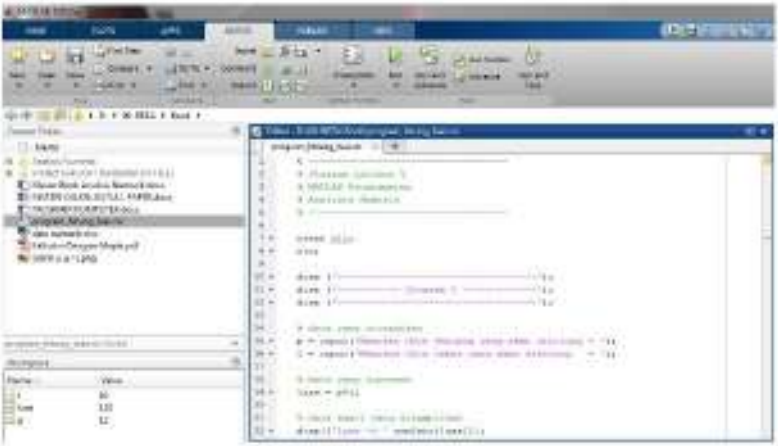
Window ini juga fasilitas yang diperkenalkan pada versi 6, berfungsi sebagai penyimpanan perintah-perintah yang pernah dikerjakan pada suatu workspace. Pada MATLAB Versi 6 ke atas. Dalam versi R2015a apabila memerlukan tampilan command history dalam MATLAB, maka klik tombol layout kemudian pilihlah command history, pilihlah docked agar ditampilkan pada menu utama.

Command Window

Gambar. 1.6. Tampilan Command Window MATLAB.

Window ini juga fasilitas yang diperkenalkan pada versi 6, berfungsi sebagai penerima perintah dari pemakai untuk menjalankan seluruh fungsi-fungsi yang disediakan oleh MATLAB. Pada dasarnya window inilah inti dari pemrograman MATLAB yang menjadi media utama satu-satunya bagi kita untuk berinteraksi dengan MATLAB.

MATLAB Editor



Gambar. 1.7. Tampilan MATLAB Editor.

Window ini berfungsi untuk membuat skrip program MATLAB. Adapun skrip program yang dapat dibuat melalui program editor seperti notepad, wordpad, word, dan lain-lain. Namun sangat dianjurkan untuk menggunakan MATLAB editor ini karena kemampuannya dalam mendeteksi kesalahan pengetikan sintaks oleh programer.

b. Cara Bekerja dengan MATLAB

Dalam melakukan pekerjaan pemrograman menggunakan bahasa MATLAB, pengguna dapat melakukan salah satu cara di bawah ini.

Penggunaan cara pertama: Perintah langsung Command Window

Pada tahapan cara pertama ini, adalah penggunaan yang paling sering dipergunakan oleh pemula, namun akan sulit bagi anda yang mengevaluasi perintah secara keseluruhan karena biasanya perintah hanya dilakukan baris per baris.

Adapun bentuk dalam penggunaan perintah langsung command window, yaitu menggunakan perintah prompt:

```
1.  >> panjang=12;
```

Kemudian tekan tombol enter, lalu ketikan:

```
1.  >> lebar=10;
```

Kemudian tekan tombol enter, lalu ketikan:

```
1.  >> luas=panjang*lebar
```

Untuk skrip terakhir sengaja tidak diberikan tanda (;) titik koma sehingga pengguna dapat langsung melihat hasil akhir dari perhitungan pada command window.

Hasil akhir yaitu;

```
1.  >> luas=120
```

Sebagai catatan, dalam penggunaan dalam nilai salah satu variabel yang dipergunakan atau lebih dari satu, misalkan pengguna mau mengganti variabel maka tinggal menekan tombol (↑) dan (↓) untuk mengganti variabel nilai yang diinginkan. Sebagai contoh: panjang = 15, maka pengguna cukup menekan tombol(↑) cari perintah panjang = 12 kemudian ganti angka yang diinginkan sehingga diperoleh tampilan sebagai berikut.

```
1.  >> panjang=15;
```

Penggunaan cara kedua: Menggunakan File-M

Pada tahapan cara kedua ini, biasanya dipilih untuk dipergunakan para programmer yang lebih mahir (dengan memahami buku dan simulasi ini, anda akan menjadi programmer pula). Kelebihan penggunaan ini adalah kemudahan untuk mengevaluasi perintah secara keseluruhan, terutama untuk program yang membutuhkan waktu pengerjaan yang cukup lama serta skrip yang cukup panjang. Dalam simulasi buku ini lebih menerapkan penggunaan file-M dalam proses analisisnya.

Berikut bentuk tampilan dengan menggunakan file-M, dengan beberapa tahapan perintah. Pertama kali, pengguna memanggil nama program dari direktori penyimpanan yang telah disiapkan. Tahapan kedua, menginputkan pemanggilan data dari inputan yang diperlukan. Agar lebih memahami dengan penggunaan file-M, perhatikan urutan perintah berikut ini.

```
2.  >> edit
```

Dengan memanggil perintah edit, maka akan memunculkan skrip yang akan dipergunakan untuk membuat program yang diperlukan. Adapun bentuk tampilan editor skrip yang kosong. Kemudian isikan skrip dengan contoh program latihan dengan tahapan sebagai berikut.

```
1.  % ----------------------------------
2.  % Program Latihan 1
3.  % MATLAB Programming
4.  % Analisis Numerik
5.  % ----------------------------------
```

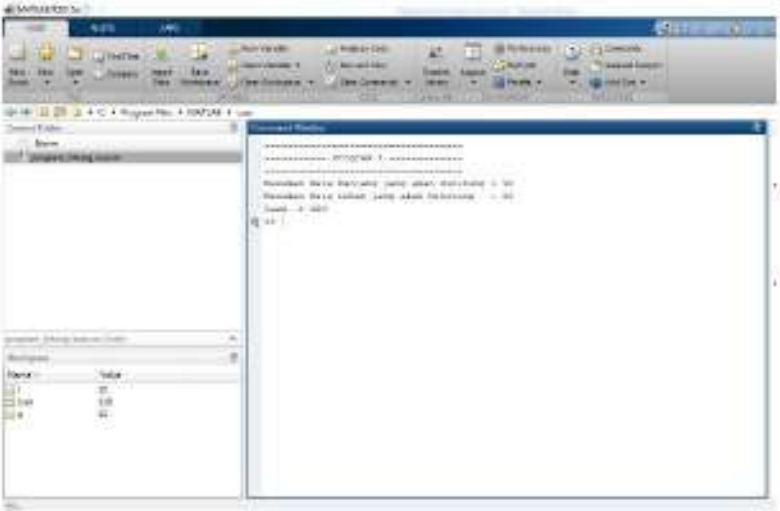
```
6.
7.  clear all;
8.  clc;
9.
10. disp ('----------------------------------');
11. disp ('----------- Program 1 -----------');
12. disp ('----------------------------------');
13.
14. % data yang diinputkan
15. p = input('Masukan Data Panjang yang akan
    dihitung = ');
16. l = input('Masukan Data Lebar yang akan
    dihitung = ');
17.
18. % data yang diproses
19. luas = p*l;
20.
21. % data hasil yang ditampilkan
22. disp(['Luas -> ' num2str(luas)]);
```

Setelah selesai mengetikkan tahapan pada program di atas, anda dapat melakukan penyimpanan pada direktori c:/Program Files/MATLAB/user, dengan nama latihan program_hitung_luas.m

Untuk mengetahui letak pada program penyimpanan di MATLAB, maka perlu diposisikan dahulu pada penyimpanan pada direktori pada bagian current folder di program MATLAB. Kemudian cek isi direktori dengan perintah di bawah ini.

```
1.  >> dir
2.  ..
3.    program_hitung_luas.m
```

Kemudian lakukan perintah pemanggilan dengan nama yang sesuai dengan penyimpanan 'program_hitung_luas' pada menu command window pada program MATLAB. Adapun bentuk tampilan hasil perintah dengan file-M akan memiliki hasil tampilan pada command window sebagai berikut.



Gambar. 1.8 Tampilan Hasil dengan Program file-M.

c. Manajemen File dan Direktori

MATLAB menggunakan metode path searching (pencarian direktori) untuk menemukan file dengan ekstensi M yang mengandung skrip dan fungsi. File M dalam MATLAB terorganisir dengan rapi pada beberapa folder/direktori. Urutan pencarian MATLAB dalam menjalankan perintah pada command window

secara bertahap adalah sebagai berikut, misalnya ketika diberikan perintah 'kubus':

1) MATLAB mencoba untuk mengenali apakah 'kubus' adalah variabel, jika ya, selesai. Jika tidak, maka MATLAB berasumsi bahwa 'kubus' adalah sebuah nama file dengan ekstensi M, lanjutkan ke tahap berikutnya.
2) MATLAB, mencoba untuk mengenali apakah 'kubus' merupakan fungsi bawaan standar, jika iya eksekusi. Jika tidak, lanjutkan ke tahap berikutnya.
3) MATLAB akan mencari file-M yang bernama kubus.m pada direktori aktif (current direktori) jika ditemukan, eksekusi. Jika tidak, lanjutkan ke tahap berikutnya.
4) MATLAB akan mencari file-M yang bernama kubus.m di sebelumnya direktori yang didaftarkan pada daftar pencariannya, jika ditemukan eksekusi. Jika tidak MATLAB akan menyampaikan pesan sebagaimana berikut ini.

```
>> kubus
??? Undefined function or variable
'kubus'.
```

Jika pesan di atas muncul, maka ada dua kesimpulan yaitu:

- Anda salah dalam menuliskan nama file, atau
- File anda tidak berada dalam direktori yang diketahui oleh MATLAB.

1.3. Operator Dasar MATLAB

Dalam MATLAB, operatornya diklasifikasikan ke dalam tiga bagian, yaitu Operator Aritmetika, Operator Relasional dan Operator Logika. Adapun penjelasan dari masing-masing operator kami berikan dalam lampiran Variabel dan Operator Dasar MATLAB.

1.4. Variabel, Operator dan Fungsi dasar MATLAB

a. Variabel

Untuk mendefinisikan suatu variabel pada MATLAB, misalkan variabel x = 8, y = 1, dan z = 3, pada Command Window, ketik:

```
>> x=8
x =
8

>> y=1
y =
1

>> z=3
z =
3
```

Jika kita ingin mengetahui variabel-variabel apa saja yang sudah kita definisikan pada MATLAB agar suatu saat kita tidak menggunakan variabel yang sama, maka ketik who di Command Window.

```
>> who
Your variabels are:
xyz
```

Perintah who di atas digunakan untuk menampilkan semua variabel yang telah terdefinisi di dalam MATLAB yang juga tercantum dalam workspace. Atau ketik whos di Command Window.

```
>> whos
NameSizeBytesClass Attributes

x 1x1 8double
y 1x1 8double
z 1x1 8double
```

Sedangkan perintah whos digunakan untuk menampilkan daftar semua variabel dalam workspace beserta ukurannya.

Jika ingin menghapus variabel x, y atau z, gunakan clear diikuti nama variabel yang akan dihapus. Misalkan kita ingin menghapus variabel x dan y, maka ketik:

```
>> clear x y
```

Untuk mengecek variabel yang tersisa ketik:

```
>> who
Your variabels are:
z
```

Sedangkan untuk menghapus semua variabel, ketik clear all.

Untuk mendefinisikan variabel secara simbolik, gunakan perintah syms dan kemudian kita juga bisa melakukan operasi aljabar pada variabel tersebut, misal:

```
>> syms x y
>> x+2*x+3*y
```

```
ans =
3*x + 3*y
```

Keterangan: yang kita lakukan pada perintah di atas adalah melakukan operasi aljabar $x + 2x + 3y$ dan setelah dieksekusi mendapatkan hasil akhir $3x + 3y$.

Contoh kedua, kita akan membuat variabel simbolik dan kita operasikan secara matriks.

```
>> syms a b c d e f g h
>> X=[a b;c d]
X =
[ a, b]
[ c, d]

>> Y=[e f;g h]
Y =
[ e, f]
[ g, h]

>> X*Y

ans =
[ a*e + b*g, a*f + b*h]
[ c*e + d*g, c*f + d*h]
```

Keterangan: yang kita lakukan pada perintah di atas adalah melakukan perkalian antara matriks X ordo 2×2 dengan matriks Y ordo 2×2

Jika kita ingin menulis suatu kata atau kalimat yang memiliki format string, kata atau kalimat tersebut harus diberi tanda (').

```
>> teks_kalimat='contoh kalimat dengan form
at string'

```

```
3.  teks_kalimat =
4.
5.  contoh kalimat dengan format string
```

b. Variabel Operator dan Fungsi Dasar Matematika

Dalam MATLAB terdapat beberapa variabel yang sudah tersedia sehingga kita tidak perlu mendefinisikan variabel tersebut. Variabel yang dimaksud tercantum di dalam tabel di bawah ini.

Tabel. 1.1. Fungsi Variabel pada MATLAB

Variabel	Keterangan
Ans	"answer", digunakan untuk menyimpan hasil perhitungan yang terakhir.
Eps	Bilangan sangat kecil mendekati nol yang merupakan batas akurasi perhitungan di MATLAB.
i, j	Unit imajiner, √-1, untuk menyatakan bilangan kompleks.
Inf	"infinity", bilangan positif tak berhingga.
NaN	"not a number", untuk menyatakan hasil perhitungan yang tak terdefinisi, misalkan 0/0 dan inf/inf.
Pi	Konstanta π, 3,141592653589793.

Operator Aritmetika digunakan untuk mengerjakan komputasi numerik dalam pemrograman MATLAB. Operator-operator aritmetika adalah

Tabel. 1.2. Fungsi Operator Aritmetika pada MATLAB

Operator	Keterangan
(+)	Berfungsi untuk penjumlahan
(-)	Berfungsi untuk pengurangan
(*)	Berfungsi untuk perkalian (aturan matriks)
(.*)	Berfungsi untuk perkalian masing-masing elemen yang bersesuaian (aturan array)
(/)	Berfungsi untuk pembagian kanan matriks
(./)	Berfungsi untuk pembagian kanan array
(\)	Berfungsi untuk pembagian kiri matriks
(.\)	Berfungsi untuk pembagian kiri array
(^)	Untuk pangkat matriks
(.^)	Untuk pangkat array

Operator Relasional digunakan untuk membandingkan operator-operator secara kualitatif. Berikut yang termasuk operator relasional.

Tabel. 1.3. Fungsi Operator Relasional pada MATLAB

Operator	Keterangan
(=)	Berfungsi sebagai tanda sama dengan
(~=)	Berfungsi sebagai tanda Tidak sama dengan
(<)	Berfungsi sebagai tanda kurang dari
(>)	Berfungsi sebagai tanda lebih dari
(<=)	Kurang dari sama dengan
(>=)	Lebih dari sama dengan

Operator	Keterangan
(&)	Akan menghasilkan nilai 1 jika kedua elemen yang bersesuaian memiliki nilai true dan 0 untuk lainnya.
(|)	Akan bernilai 1 jika salah satu elemennya true
(~)	Komplemen dari elemen yang diinputkan
(xor)	Akan bernilai 1 jika salah satu dari kedua elemen memilki nilai berbeda dan bernilai 0 jika sama

c. Fungsi bagian bantuan dari MATLAB (help)

Selain "help" yang bersifat umum seperti pada pelajaran sebelumnya, terdapat juga fasilitas help untuk beberapa fungsi matematika dan operator yang sudah tersedia dalam MATLAB. Coba ketik salah satu fungsi yang terdapat pada tabel di bawah ini, maka akan muncul fungsi-fungsi atau operator yang berkaitan dengan topik tersebut.

Tabel. 1.4. Fungsi Help pada MATLAB

Fungsi	Keterangan
help general	Perintah untuk tujuan umum.
help ops	Karakter khusus dan operator.
help lang	Perintah bahasa pemrograman.
help elmat	Matriks dasar dan manipulasi matriks.
help elfun	Fungsi matematika dasar.
help specfun	Fungsi matematika khusus.
help matfun	Fungsi matriks-aljabar linear

	matriks.

Sebagai contoh, jika kita ketik help elfun pada Command Window maka akan muncul fungsi-fungsi matematika yang dibagi menjadi beberapa kategori, yaitu fungsi trigonometri, eksponensial, bilangan kompleks, pembulatan, dan sisa.

```
>> help elfun
  Elementary math functions.

  Trigonometric.
    sin         - Sine.
    sind        - Sine of argument in degrees.
    sinh        - Hyperbolic sine.
    asin        - Inverse sine.
    asind       - Inverse sine, result in degrees.
    asinh       - Inverse hyperbolic sine.
    cos         - Cosine.
    cosd        - Cosine of argument in degrees.
    cosh        - Hyperbolic cosine.
    acos        - Inverse cosine.
    acosd       - Inverse cosine, result in degrees.
    acosh       - Inverse hyperbolic cosine.
    tan         - Tangent.
    tand        - Tangent of argument in degrees.
    tanh        - Hyperbolic tangent.
    atan        - Inverse tangent.
    atand       - Inverse tangent, result in degrees.
    atan2       - Four quadrant inverse tangent.
    atanh       - Inverse hyperbolic tangent.
    sec         - Secant.
    secd        - Secant of argument in degrees.
    sech        - Hyperbolic secant.
    asec        - Inverse secant.
    asecd       - Inverse secant, result in degrees.
    asech       - Inverse hyperbolic secant.
    csc         - Cosecant.
    cscd        - Cosecant of argument in degrees.
    csch        - Hyperbolic cosecant.
    acsc        - Inverse cosecant.
    acscd       - Inverse cosecant, result in degrees.
    acsch       - Inverse hyperbolic cosecant.
    cot         - Cotangent.
    cotd        - Cotangent of argument in degrees.
    coth        - Hyperbolic cotangent.
    acot        - Inverse cotangent.
    acotd       - Inverse cotangent, result in degrees.
    acoth       - Inverse hyperbolic cotangent.
    hypot       - Square root of sum of squares.

```

```
Exponential.
  exp           - Exponential.
  expm1         - Compute exp(x)-1 accurately.
  log           - Natural logarithm.
  log1p         - Compute log(1+x) accurately.
  log10         - Common (base 10) logarithm.
  log2          - Base 2 logarithm and dissect floating point number.
  pow2          - Base 2 power and scale floating point number.
  realpow       - Power that will error out on complex result.
  reallog       - Natural logarithm of real number.
  realsqrt      - Square root of number greater than or equal to zero.
  sqrt          - Square root.
  nthroot       - Real n-th root of real numbers.
  nextpow2      - Next higher power of 2.

Complex.
  abs           - Absolute value.
  angle         - Phase angle.
  complex       - Construct complex data from real and imaginary parts.
  conj          - Complex conjugate.
  imag          - Complex imaginary part.
  real          - Complex real part.
  unwrap        - Unwrap phase angle.
  isreal        - True for real array.
  cplxpair      - Sort numbers into complex conjugate pairs.

Rounding and remainder.
  fix           - Round towards zero.
  floor         - Round towards minus infinity.
  ceil          - Round towards plus infinity.
  round         - Round towards nearest integer.
  mod           - Modulus (signed remainder after division).
  rem           - Remainder after division.
  sign          - Signum.
```

1.5. Pendahuluan Analisis Numerik

Persoalan yang melibatkan model matematika yang memunculkan dalam berbagai disiplin ilmu pengetahuan, seperti dalam bidang fisika, kimia, ekonomi, atau pada persoalan rekayasa (engineering), seperti Teknik Sipil, Teknik Mesin, Elektro dan sebagainya. Seringkali model matematika tersebut muncul dalam bentuk yang tidak ideal atau disebut rumit. Model matematika yang rumit ini adakalanya tidak dapat diselesaikan dengan metode analitik yang sudah umum untuk mendapatkan solusi sejatinya (exact solution). Metode analitik adalah metode

penyelesaian model matematika dengan rumus-rumus aljabar yang selakunya sudah baku (lazim). Sebagai contoh perhatikan sekumpulan persoalan matematik dibawah ini.

a. Tentukan akar-akar persamaan polinom.
$5x^8 - 2.25x^6 - 120x^2 + 15x^3 - 120x^2 - x + 90 = 0$

b. Selesaikan sistem persamaan linier
$1.2a - 3b - 12c + 12d + 4.8e - 5.5f + 100g = 18$
$0.9a + 3b - c + 16d + 8e - 5f - 10g = 17$
$4.6a + 3b - 6c - 2d + 4e + 6.5f - 13g = 19$
$3.7a - 3b + 8c - 7d + 14e + 8.4f + 16g = 6$
$2.2a + 3b + 17c + 6d + 12e - 7.5f + 18g = 9$
$5.9a + 3b + 11c + 9d - 5e - 25f + 10g = 0$
$1.6a + 3b + 1.8c + 12d - 7e + 2.5f + g = -5$

c. Tentukan nilai maksimum fungsi tiga dimensi
$$F(x,y) = \cos\frac{x - \sqrt{\sin 2x} + 3}{4 + (xy)^2}$$

d. Bila diperoleh kumpulan data titik-titik (x,y) sebagai berikut. (yang dalam hal ini rumus fungsi y=f(x) tidak diketahui secara eksplisit.

X	2.4	3	3.5	4.5	6.5
Y=f(x)	1.4253	1.7562	2.0005	2.8975	3.8795

Melihat dari permasalahan matematika, terdapat benak pemikiran "bagaimana cara menyelesaikannya?". Untuk menghadapi permasalahan tersebut mungkin sebagian besar tidak mampu atau bahkan menyerah untuk menyelesaikan permasalahan tersebut dengan metode yang dikenal. Perhatikan permasalahan pertama, untuk polinom derajat 2 masih mampu

diselesaikan dengan rumus abc yaitu $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ atau dengan cara metode yang lain. Namun, untuk polinom derajat lebih dari dua, rumus abc tidak dapat dipergunakan untuk menyelesaikan permasalahan pertama. Solusi lain untuk menyelesaikan, mungkin harus memanipulasi polinom, misalnya dengan memfaktorkan atau menguraikan polinom menjadi perkalian beberapa suku. Akan tetapi, langkah tersebut memiliki kelemahan yaitu apabila derajat polinom sangat tinggi, maka akan sangat sulit melakukan manipulasi polinomnya.

Melihat dari permasalahan kedua, juga tidak ada rumus yang baku untuk menentukan solusi Sistem Persamaan Linier, apabila persamaannya hanya berupa dua garis lurus dengan dua peubah masih dapat diselesaikan dengan menggunakan metode eliminasi dan substitusi, menggambarkan grafik untuk mengetahui titik potongnya dan beberapa metode lainnya. Untuk sistem persamaan linier dengan tiga buah persamaan linier aturan-aturan tersebut masih dapat dipergunakan untuk menyelesaikan permasalahan sistem persamaan linier dengan persamaan dan peubah lebih besar dari empat, aturan-aturan tersebut akan mengalami kesulitan dalam menentukan titik potong garis. Namun, aturan eliminasi, substitusi, menggambarkan grafik belum efektif dan efisiensi juga dipergunakan untuk menyelesaikan persamaan linier dengan peubah lebih dari empat. Jika permasalahan kedua dipaksakan dengan metode-metode ini akan

semakin panjang dan sulit untuk menyelesaikan permasalahan sistem persamaan linier ini.

Melihat dari permasalahan ketiga, memiliki kendala dalam mencari titik optimum fungsi yang memiliki banyak peubah. Untuk menentukan titik ekstrem fungsi, pertama-tama harus menentukan turunan fungsi, dengan menjadikan ruas kanannya sama dengan nol, lalu memeriksa jenis dari titik ekstremnya. Apabila fungsinya cukup rumit dan disusun oleh banyak peubah, menghitung turunan fungsi menjadi pekerjaan yang cukup sulit atau bahkan tidak mungkin diselesaikan.

Melihat dari permasalahan keempat, pertanyaan yang mungkin pertama kali muncul "bagaimana menghitung nilai sebuah fungsi dengan rumus fungsinya yang tidak diketahui?". Kita mungkin memahami nilai suatu fungsi diperoleh dengan cara mensubstitusikan suatu harga dari peubahnya ke dalam persamaan fungsi. Berdasarkan pengamatan tidak diketahuinya relasi yang menghubungkan parameter-parameter itu, sehingga tidak satu pun metode yang tersedia untuk menyelesaikan permasalahan-permasalahan jenis ini.

1.6. Analisis Numerik dan Analisis Analitik

Analisis Analitik

Solusi berupa fungsi matematik yang selanjutnya fungsi matematik tersebut dapat dievaluasi untuk menghasilkan nilai dalam bentuk angka, Solusi

yang dihasilkan solusi exact. Metode dengan analisis analitik hanya unggul untuk sejumlah permasalahan-permasalahan yang terbatas, yaitu permasalahan yang memiliki tafsiran geometri sederhana. Padahal persoalan yang muncul dalam dunia nyata seringkali permasalahan yang melibatkan bentuk dan proses yang rumit. Apabila analisis analitik tidak dapat lagi diterapkan, maka solusi persoalan sebenarnya masih dapat dicari dengan menggunakan metode numerik.

Analisis Numerik

Solusi selalu berbentuk angka Solusi yang dihasilkan solusi pendekatan/hampiran (approxomation), solusi hampiran jelas tidak tepat sama dengan solusi sejati, sehingga ada selisih antara keduanya yang disebut galat (error). Analisis numerik merupakan proses rekayasa dan penelitian, analisis diharapkan dapat menghasilkan bilangan yang diperlukan dalam perencanaan ataupun penghayatan masalah.

Perbedaan utama antara analisis numerik dan analisis analitik terletak pada dua hal. Pertama, solusi dengan menggunakan analisis numerik selalu berbentuk angka. Bandingkan dengan analisis analitik yang biasanya dalam bentuk fungsi matematik yang selanjutnya bentuk fungsi matematik tersebut dievaluasi untuk menghasilkan nilai dalam bentuk angka. Kedua, dengan analisis numerik kita hanya memperoleh solusi yang menghampiri atau mendekati

solusi sejati sehingga solusi numerik dinamakan juga solusi hampiran atau solusi pendekatan. Solusi hampiran jelas tidak tepat sama dengan solusi sejati sehingga ada selisih diantara keduanya. Selisih inilah yang disebut galat (error).

Dalam analisis analitik selalu digunakan cara yang sudah baku atau dengan aturan-aturan kalkulus, Hasil perhitungan berupa suatu fungsi/relasi, Nilai perhitungan adalah nilai sejati atau exact (tepat), tidak selalu mudah memperoleh solusi. Sedangkan, analisis numerik dalam perhitungan menggunakan aritmetika seperti tanda +,-,* dan /, hasilnya berupa angka, nilai perhitungan adalah hampiran adalah hampiran atau tidak exact, solusi selalu dapat diperoleh dengan bantuan komputer.

Sebagai contoh ilustrasi penyelesaian dengan analisis numerik, lihatlah sebuah persoalan $I = \int_{-1}^{1}(9 + x^2)dx$. Perhitungan analisis analitik, kita dapat menemukan solusi sejatinya dengan mudah. Pada materi kalkulus integral tentu dapat diketahui teknik pengintegralan untuk fungsi sederhana seperti $\int a_n = \frac{1}{n+1}ax^{n+1} + C$. Berdasarkan hal tersebut maka dapat diartikan pengintegralan suku-suku dari fungsi integralnya lalu menghitung nilai integral tentunya.

Adapun pembahasan penyelesaian dengan analisis analitik, dengan kalkulus integral dibahas sebagaimana berikut ini.

$$\int_{-1}^{1} (9 - x^2)\, dx$$

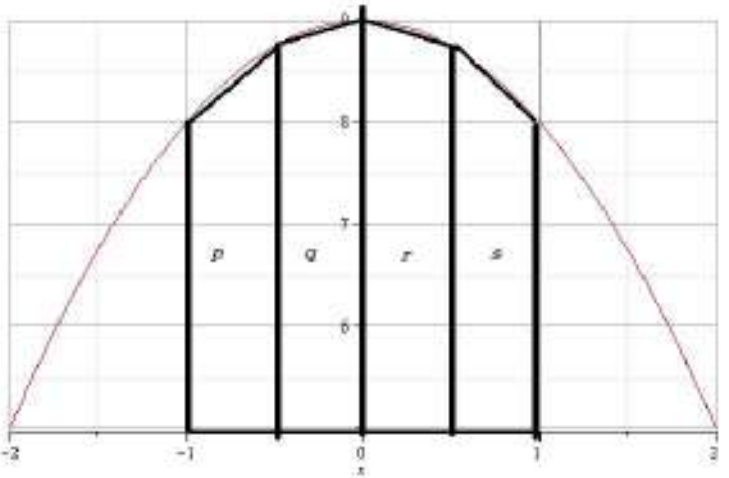
$$= \int_{-1}^{1} 9\, dx + \int_{-1}^{1} -x^2\, dx$$

$$= 18 + \int_{-1}^{1} -x^2\, dx$$

$$= 18 - \int_{-1}^{1} x^2\, dx$$

$$= \frac{52}{3}$$

Bandingkan hasil akhir dari perhitungan dengan analisis analitik sebesar $\frac{52}{3}$ memperoleh perhitungan dalam bentuk desimal memperoleh 17.33333333. Maka penyelesaian analisis analitik dengan persoalan integrasi tersebut diselesaikan dengan analisis numerik. Dalam penyelesaian kalkulus integral kita tentu masih ingat bahwa interprestasi geometri integral f(x) dari x = a sampai x = b, adalah luas daerah yang dibatasi oleh kurva f(x), sumbu-x, dan garis x = a dan x = b. Luas daerah tersebut dapat dihampiri dengan cara sebagai berikut. Bagilah daerah integrasi [-1, 1] atas sejumlah trapesium dengan luasan satuan 0.5 perhatikan Gambar. 1.8. Maka, luas daerah integrasi dihampiri dengan luasan keempat trapesium.



Gambar. 1.8. Grafik Trapesiumy = $9 + x^2$

Adapun hasil perhitungan yang mengacu bentuk dari luas daerah integrasi dari luasan satu hingga sampai luasan keempat trapesium, dengan perhitungan sebagai berikut.

$I \approx p + q + r + s$

$\approx \{[f(-1) + f(-1/2)] \times 0.5/2\} + \{[f(-1/2) + f(0)] \times 0.5/2\}$
$+ \{[f(0) + f(1/2)] \times 0.5/2\} + \{[f(1/2) + f(1)] \times 0.5/2\}$

$\approx 0.5/2 \{f(-1) + 2.f(-1/2) + 2.f(0) + 2.f(1/2) + f(1)\}$

$\approx 0.5/2 \{10 + 2.(9.25) + 2.(9) + 2.(9.25) + 10\}$

$\approx 0.5/2 \{70\}$

≈ 17.25

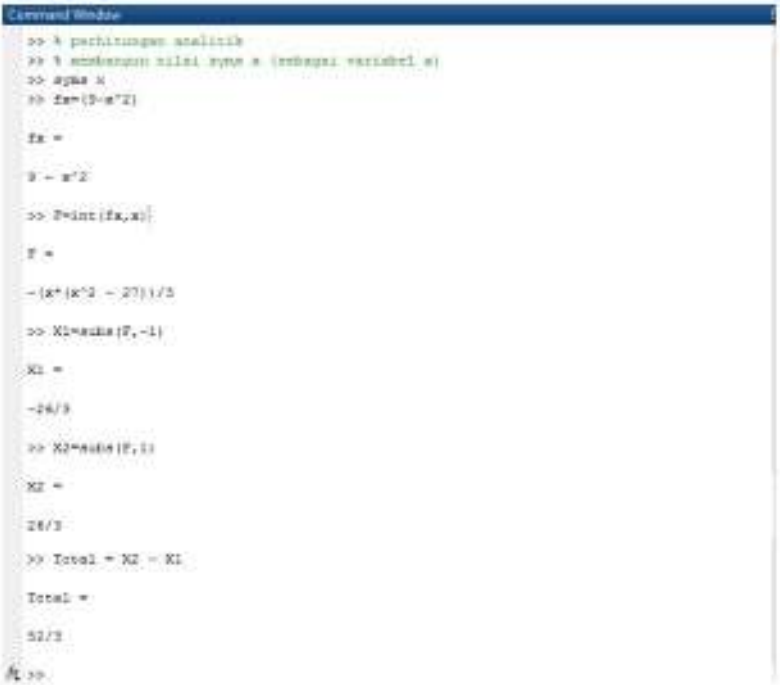
Memperhatikan hasil dari perbandingan, kita dapat melihat untuk dapat memperkecil galat tersebut dengan membuat lebar trapesium yang lebih kecil (yang berarti jumlah pada trapesium semakin banyak, yang

berarti jumlah komputasi semakin banyak). Contoh ini juga memperlihatkan bahwa meskipun solusi dengan analisis numerik merupakan hampiran, tetapi hasilnya dapat dilakukan dengan hasil yang seteliti mungkin dengan cara mengubah parameter komputasi (pada contoh perhitungan integral di atas, lebar trapesium yang dikurangi).

Maka hasil dari galat yang diperoleh solusi hampiran (tanda "≃" artinya kira-kira) terhadap solusi sejati (52/3). Galat solusi hampiran terhadap solusi sejati adalah galat = | 17.25 - 17.3333 | = 0.0833

1.7. Program MATLAB

Sebagaimana dalam bahasa pemrograman pada umumnya, MATLAB mengembangkan metode perhitungan dalam sintaksnya memperhitungkan hasil dari perbandingan dari perhitungan analisis analitik dan analisis numerik. Perhitungan perbandingan dengan menggunakan program MATLAB menggunakan command window dalam memperhitungkan analisis analitik, dengan kalkulus integral dibahas sebagaimana berikut ini.



Gambar. 1.9. Tampilan Perhitungan Analisis Analitik $y=9+x^2$ Command Window dengan MATLAB

Bentuk hasil perhitungan melalui command window terhadap permasalahan $y = 9 + x^2$ memperoleh hasil $\frac{52}{3}$, agar perhitungan secara rasional dari analisis analitik dapat dibandingkan dengan hasil dari analisis numerik, maka perlu dilakukan perhitungan secara hasil desimal dengan langkah yang tertampil pada gambar berikut ini.

Gambar. 1.10. Tampilan Perhitungan format short dengan MATLAB

Dalam perhitungan analisis analitik maka format short diperoleh hasil nilai desimal sebesar 17.3333 dari perhitungan $\frac{52}{3}$, adapun pembandingan pembahasan dengan menggunakan program MATLAB terhadap analisis numerik dengan perhitungan dengan luas daerah integrasi dari luasan satu hingga sampai luasan keempat trapesium, dengan perhitungan sebagai berikut.

Gambar. 1.11. Tampilan Perhitungan Analisis Numeriky $=9+x^2$ Command Window dengan MATLAB

Diperoleh hasil perhitungan analitik (nilai sejati) dan perhitungan numerik (nilai hampiran) yang dibandingkan dan dihitung galat solusi hampiran dengan menggunakan program MATLAB, sehingga diperoleh langkah sebagai berikut.

Gambar. 1.12. Tampilan Perhitungan Galat terhadap hasily =9+x² Command Window dengan MATLAB

SIMULASI 2: GALAT

2.1. Pengantar Galat

Metode numerik merupakan suatu metode untuk menyelesaikan masalah-masalah matematika dengan menggunakan sekumpulan aritmetika sederhana dan operasi logika pada sekumpulan bilangan atau data numerik yang diberikan. Metode komputasi yang digunakan disebut algoritma. Proses penyelesaiannya mungkin memerlukan puluhan bahkan sampai jutaan operasi, tergantung pada kompleksitas masalah yang harus diselesaikan, tingkat keakuratan yang diinginkan dan seterusnya.

Setiap penyelesaian akhir yang diperlukan, misalnya set dari tabulasi data yang diberikan dan berbentuk numerik. Tujuan dari analisis numerik memberikan bentuk dari metode-metode yang efisien untuk memperoleh jawaban numerik dari berbagai permasalahan. Metode numerik dalam permasalahan yang ada di berbagai bidang belum meluas. Hal ini disebabkan karena pada masa tersebut alat bantu hitungan belum meluas dan harganya pun sangat mahal. Saat ini metode numerik telah berkembang pesatnya, dalam beberapa tahun terakhir ini perkembangan komputer sangat pesatnya dan harganya sudah sangat terjangkau. Metode numerik mampu menyelesaikan suatu sistem persamaan yang besar,

tidak linier dan sangat kompleks yang tidak mungkin diselesaikan secara analitik.

Perlunya metode numerik banyak dikembangkan oleh para ahli matematika, tetapi ilmu tersebut tidak hanya diperuntukkan milik mereka. keilmuan metode numerik adalah milik semua kalangan berbagai ahli bidang, seperti bidang teknik (sipil, mesin, kimia, elektri, dan sebagainya), kedokteran, ekonomi, sosial dan bidang ilmu lainnya. Teknik/metode penyelesaian permasalahan yang diformulasikan secara sistematis dengan cara operasi hitung (aritmatik). pendekatan penyelesaian dengan metode ini dilakukan apabila penggunaan penyelesaian dengan metode ini dilakukan apabila penyelesaian secara umum (analitis) sulit dilakukan. Hal-hal khusus yang dimiliki metode ini adalah memiliki bentuk sistem perhitungan yang berulang-ulang (iteratif) yang memerlukan alat bantu proses otomatisasi dari iterasi tersebut yaitu (program) komputer.

Dengan menggunakan metode pendekatan semacam ini, penyelesaian secara numerik dari suatu bentuk persamaan matematik, hanya memberikan nilai perkiraan yang mendekati nilai eksak (nilai solusi sejati) dari penyelesaian analitik. Berarti dalam suatu penyelesaian analisis numerik tersebut terdapat kesalahan terhadap nilai sejatinya. Hal ini dapat dilihat pada bab sebelumnya pada simulasi pertama yang membahas perbandingan nilai solusi sejati dan nilai solusi hampiran. Tentunya, dalam setiap nilai hasil

perhitungan akan memiliki galat (error) atau nilai kesalahan. Kesalahan ini memiliki peran sentral, artinya karena kesalahannya besar, tentunya ini tidak diharapkan. Sehingga dalam pendekatan analisis numerik membahas tingkat kesalahan dan tingkat kecepatan dalam proses yang akan terjadi.

2.2. Galat (Kesalahan/Kekeliruan)

Masalah-masalah matematika yang sering kita hadapi merupakan masalah matematika yang diselesaikan dengan analisis analitik atau nilai sejati, yaitu suatu metode yang memberikan solusi sejati atau solusi yang sesungguhnya, karena memiliki galat (error) yang bernilai nol. Tetapi penyelesaian dengan menggunakan analisis analitik hanya terbatas pada masalah tertentu saja. Bila analisis analitik tidak dapat lagi diterapkan, maka solusinya masih dapat dicari yaitu dengan menggunakan metode numerik. Pada analisis numerik solusinya merupakan hampiran (pendekatan) terhadap solusi sejati.

Penyelesaian secara numeris dari suatu persamaan matematika hanya memberikan nilai perkiraan yang mendekati nilai sebenarnya. Berarti dalam penyelesaian numerik terdapat galat atau kesalahan terhadap nilai sejati atau penyelesaian yang sebenarnya. Galat berasosiasi dengan seberapa dekat solusi hampiran dengan solusi sejatinya. Semakin kecil galat, maka semakin teliti solusi numerik yang diperoleh. Ada empat macam kesalahan, yaitu

kesalahan bawaan, kesalahan pembulatan, kesalahan kompresi dan kesalahan pemotongan.

Kesalahan bawaan (inheren) adalah kesalahan (galat) dalam nilai data, yang disebabkan ketidak-pastian dalam pengukuran, kekeliruan atau oleh perlunya pendekatan dan kesalahan ini bila terjadi karena kekeliruan dalam penyalinan data, kesalahan dalam membaca skala atau kesalahan karena kurangnya pengertian mengenai hukum-hukum fisik dari data yang diukur.'

Perhitungan dalam analisis numerik, diantaranya terdapat sebagian yang tidak eksak, hal ini disebabkan karena adanya data yang diperoleh berupa data aproksimasi, keterbatasan dari pemrograman komputasi seperti dalam buku ini menerapkan pemrograman matematika menggunakan MATLAB, walaupun ada beberapa program lain seperti SCILAB, excell dan lain-lain. Karena, keterbatasan tersebut hasil yang diperoleh berupa data yang dibulatkan, sehingga dapat kita ketahui apa yang disebut dengan kesalahan pembulatan.

Kesalahan pembulatan, terjadi dikarenakan tidak diperhitungkan beberapa angka terakhir dari suatu bilangan. kesalahan ini terjadi apabila terjadi perkiraan digunakan untuk menggantikan bilangan eksak. suatu bilangan eksak dibulatkan pada posisi ke n dengan membuat semua angka disebelah kanan dari posisi tersebut nol, sedangkan angka pada posisi ke n tersebut tidak berubah atau dinaikkan satu digit yang

tergantung apakah bilangan tersebut lebih kecil atau lebih besar dari setengah dari angka posisi ke n. Pembulatan maksudnya mengurangi cacah digit pada suatu nilai hampiran dengan cara membuang beberapa digit terakhir. Cara melakukan pembulatan sudah dijelaskan di depan. Galat pembulatan terjadi disebabkan karena adanya pembulatan dalam komputasi numerik, sehingga pengulangan pembulatan tidak disarankan dalam komputasi numerik karena hanya akan memperbesar galat.

Sebagai contoh, nilai:

8632574	dapat dilakukan pembulatan menjadi	8633000
3,1415926	dapat dilakukan pembulatan menjadi	3,14

Kesalahan kompresi, merupakan bentuk kesalahan yang tidak dapat dihindarkan sehingga memang terkadang dilakukan secara sengaja. Kesalahan tahap ini diakibatkan oleh pengguna rumus atau formula aproksimasi dalam perhitungan atau disebut kesalahan mutlak.

Kesalahan pemotongan terjadi karena tidak dilakukan perhitungan sesuai dengan prosedur matematika yang benar. Kesalahan pemotongan merujuk pada galat yang disebabkan oleh penggantian ekspresi matematika yang rumit dengan rumus yang lebih sederhana. Penghentian suatu deret yang tak berhingga menjadi suatu deret yang berhingga itulah sebenarnya yang menyebabkan galat pemotongan. Sebagai contoh, suatu proses tak terhingga diganti dengan proses berhingga. Di dalam matematika, suatu

fungsi dipresentasikan dalam bentuk deret tak terhingga, misalkan:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

Nilai eksak dari e^x diperoleh hasil apabila semua suku dari deret tersebut diperhitungkan. Namun, dalam prakteknya sulit untuk dilakukan perhitungan semua suku dmapai tak terhingga. Apabila hanya diperhitungkan beberapa suku pertama disebut sebagai kesalahan pemotongan. Untuk lebih memahami kesalahan pemotongan ini akan dijelaskan lebih dalam materi simulasi 2.4 Sumber Utama Galat.

2.3. Galat Absolut dan Relatif

Relasi keterikatan antara nilai sejati (eksak), nilai hampiran (perkiraan) dan kesalahan (galat) berasosiasi dengan seberapa dekat solusi hampiran dengan solusi sejatinya. Semakin kecil galat, maka semakin teliti solusi numerik yang diperoleh, Berarti dalam penyelesaian numerik terdapat galat atau kesalahan terhadap nilai sejati atau penyelesaian yang sebenarnya. hal ini dapat dinyatakan dalam bentuk berikut ini.

$$a = \hat{a} + \varepsilon$$

Misalkan â adalah nilai hampiran terhadap nilai sejati a, maka bentuk dari selisih ε = a – â disebut galat. Nilai galat positif ataupun negatif tidak berpengaruh, sehingga perhitungannya digunakan tanda mutlak, sehingga didefinisikan sebagai berikut.

$$|\varepsilon| = |a - \hat{a}|$$

ukuran galat ε kurang bermakna karena tidak menjelaskan seberapa besar galat tersebut dengan nilai sejatinya. Sebagai contoh misalkan panjang seutas tali panjangnya 99 cm, padahal panjang sebenarnya 100 cm, sehingga galatnya 100 – 99 = 1 cm. Kemudian sebatang pensil panjangnya 9 cm padahal panjang sebenarnya 10 cm, sehingga galatnya juga 1 cm, namun galat 1 cm pada pengukuran sebatang pensil lebih berarti dari pada galat 1 cm pada pengukuran panjang tali, mengapa? Jika tidak ada keterangan panjang sesungguhnya kita menganggap kedua galat itu sama, sehingga untuk mengintepretasi kedua galat ini harus dinormalkan terhadap nilai sejatinya, sehingga melahirkan istilah galat relatif (ε_R). Galat relatif didefinisikan sebagai berikut:

$$\varepsilon_R = \frac{\varepsilon}{a} \times 100\ \%$$

Karena galat dinormalkan terhadap nilai sejati, maka galat relatif tersebut dinamakan juga galat relatif sejati. Sehingga pengukuran pengukuran panjang tali mempunyai galat relatif sejati = 1/100 = 0,01, sedang pengukuran panjang pensil mempunyai galat relatif sejati = 1/10 = 0,1. Adapun bentuk Galat relatif hampiran didefinisikan sebagai berikut:

$$\varepsilon_{RA} = \frac{\varepsilon}{\hat{a}} \times 100\ \%$$

2.4. Sumber Utama Galat

Secara umum ada dua sumber penyebab galat dalam perhitungan numerik,
Yaitu Galat Pemotongan dan Galat Pembulatan.

a. Galat Pemotongan

Pengertian galat pemotongan merujuk pada galat yang disebabkan oleh penggantian ekspresi matematika yang rumit dengan rumus yang lebih sederhana. Penghentian suatu deret yang tak berhingga menjadi suatu deret yang berhingga itulah sebenarnya yang menyebabkan galat pemotongan.

Sebagai contoh, perhatikan hitungan kesalahan yang terjadi pada nilai e^x apabila hanya diperhitungan beberapa suku pertama saja. Nilai eksak dari $e^{0.5}$ = 1,648721271.

Untuk menunjukkan pengaruh yang diperhitungkan dengan beberapa suku pertama dari deret terhadap besarnya kesalahan pemotongan, maka hitungan yang dilakukan dengan beberapa kondisi. Kondisi pertama memperhatikan perhitungan pada suku pertama, keadaan kondisi kedua hanya pada dua suku pertama, dan seterusnya sampai memperhitungkan 6 suku pertama. Nilai e^x dapat dihitung berdasarkan deret berikut ini.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

1) Tahap ini memperhitungkan satu suku pertama:

$e^{0.5} \approx 1$

Kesalahan relatif terhadap nilai eksak diperhitungkan dengan menggunakan persamaan

formula galat relatif, dengan langkah sebagai berikut.

$$\varepsilon_R = \frac{\varepsilon}{a} \times 100\ \% = \frac{1{,}648721271 - 1}{1{,}648721271} \times 100\ \% = 39{,}35\ \%$$

2) Tahap berikutnya memperhitungkan dua suku pertama:

$e^x = 1 + x$

Untuk x = 0,5 maka:

$e^x = 1 + 0{,}5 = 1{,}5$

Kesalahan relatif terhadap nilai eksak adalah

$$\varepsilon_R = \frac{\varepsilon}{a} \times 100\ \% = \frac{1{,}648721271 - 1{,}5}{1{,}648721271} \times 100\ \% = 9{,}02\ \%$$

Kesalahan berdasarkan perkiraan terbaik dihitung dengan persamaan galat relatif hampiran:

$$\varepsilon_{RA} = \frac{\varepsilon}{\hat{a}} \times 100\ \% = \frac{|1 - 1{,}5|}{1{,}5} \times 100\ \% = 33{,}33\%$$

3) Tahapan dalam memperhitungkan tiga suku pertama:

$$e^x = 1 + x + \frac{x^2}{2!} = 1 + 0{,}5 + \frac{0{,}5^2}{2!} = 1{,}625$$

Kesalahan relatif terhadap nilai eksak adalah

$$\varepsilon_R = \frac{1{,}648721271 - 1{,}625}{1{,}648721271} \times 100\ \% = 1{,}44\ \%$$

Kesalahan berdasarkan perkiraan ternbaik dihitung dengan persamaan galat relatif hampiran:

$$\varepsilon_{RA} = \frac{1{,}625 - 1{,}5}{1{,}625} \times 100\ \% = 7{,}69\ \%$$

Adapun, perhitungan dilanjutkan dengan sampai 6 suku pertama, dan hasilnya ditampilkan dalam tabel berikut ini.

Tabel. 2.1. Hasil data pada enam suku pertama dari nilai e^x

Suku	Hasil	ε_R (%)	ε_{RA} (%)
1	1	39,3	-
2	1,5	9,02	33,3
3	1,625	1,44	7,69
4	1,645833333	0,175	1,27
5	1,648437500	0,0172	0,158
6	1,648697917	0,00142	0,0158

Berpandangan pada permasalahan dengan perkiraan deret taylor memiliki dasar untuk menyelesaikan perhitungan dalam masalah analisis numerik, terutama untuk penyelesaian persamaan diferensial. Jika suatu fungsi f(x) diketahui titik x_i dan semua turunan dari f terhadap x diketahui pada titik tersebut, maka dengan deret taylor yang dapat dinyatakan nilai f pada titik x_{i+1} yang terletak pada jarak Δx dari titik x_i. Adapun bentuk dari deret taylor sebagai berikut.

$$f(x_{i+1}) = f(x_i) + f'(x_i)\frac{\Delta x}{1!} + f''(x_i)\frac{\Delta x^2}{2!} + f'''(x_i)\frac{\Delta x^3}{3!} + \ldots + f^n(x_i)\frac{\Delta x^n}{n!} + R_n$$

Keterangan:

$f(x_i)$ = fungsi di titik x

$f(x_{i+1})$ = fungsi di titik x i + 1

$f', f'', \ldots f^n$ = turunan pertama, kedua, ... ke n dari fungsi

Δx = jarak antara xi dan xi + 1

R_n = kesalahan pemotongan

! = operator faktorial, misal 2! = 1 x 2

Dalam persamaan deret taylor kesalahan pemotongan R_n diberikan bentuk berikut ini.

$$R_n = f^{n+1}(x_i)\frac{\Delta x^{n+1}}{(n+1)!} + f^{n+2}(x_i)\frac{\Delta x^{n+2}}{(n+2)!} + ...$$

Persamaan deret taylor yang mempunyai suku banyak tak terhingga akan memberikan perkiraan nilai suatu fungsi sesuai dengan penyelesaian eksaknya. Dalam menggunakan perhitungan semua suku tersebut dan biasanya hanya diperhitungkan dengan beberapa suku pertama saja.

1) Memperhitungkan satu suku pertama (Order Nol).

   $f(x_{i+1}) \approx f(x_i)$

   Perkiraan akan benar bila fungsi yg diperkirakan adalah konstan

2) Memperhitungkan dua suku pertama (Order Satu)

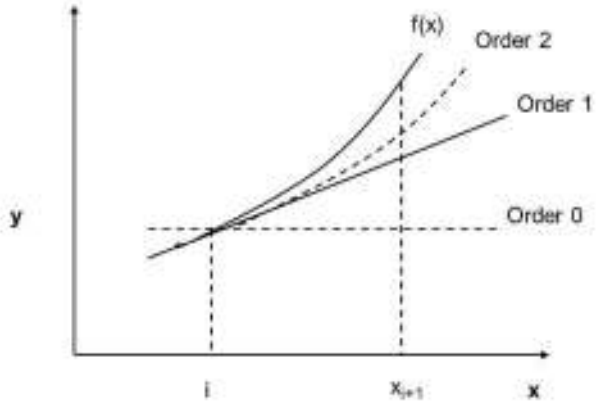
   $$f(x_{i+1}) = f(x_i) + f'(x_i)\frac{\Delta x}{1!}$$

   Berupa garis lurus (naik/turun)

3) Memperhitungkan tiga suku pertama (Order Dua)

   $$f(x_{i+1}) = f(x_i) + f'(x_i)\frac{\Delta x}{1!} + f''(x_i)\frac{\Delta x^2}{2!}$$

Perkiraan dari suatu fungsi dengan deret taylor secara grafis ditunjukkan sebagai berikut.



Gambar. 2.1. Tampilan Perkiraan dari Suatu Fungsi dengan Deret Taylor

Melihat permasalahan dari hampiran deret taylor f(x) = cos (x) di sekitar x = 0, deret Taylor tersebut dipotong sampai suku orde ke-6. kita lihat deret Taylor tersebut dipotong sampai suku orde ke-6 tersebut tidak memberikan suku yang tepat. Galat pada nilai hampiran diakibatkan oleh pemotongan suku-suku deret. Jumlah suku-suku setelah pemotongan deret dinamakan galat pemotongan untuk cos (x). Tetapi kita tidak dapat menghitung galat pemotongan tersebut karena jumlahnya tak mungkin bisa dihitung. Namun, kita dapat menghampiri galat pemotongan ini dengan rumus suku sisa: perhatikan bentuk dari deret taylor f(x) = cos (x).

$$R_n(x) = \sum_{k=1}^{n} \frac{(x - x_0)^{(n+1)}}{(n+1)!} f^{(n+1)}(c), \quad x_0 < c < x.$$

Pada contoh cos (x) diatas

$$R_6(x) = \frac{x^7}{7!}\cos(c).$$

Untuk nilai c pada batasan selang tertentu, maka nilai maksimum yang mungkin dari R_n untuk c dalam selang tersebut:

$$|R_n(x)| < Maks_{x_0 < c < x} \left| f^{(n+1)}(c) \right| \times \frac{(x - x_0)^{n+1}}{n+1}$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots \pm \frac{x^{(2n)}}{(2n)!} + R_n(x)$$

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + R_5(x)$$

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + R_7(x)$$

$$= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + R_8(x)$$

Perhatikan kasus contoh permasalahan berikut.

Diketahui suatu fungsi $f(x) = 0{,}25x^3 + 0{,}5x^2 + 0{,}25x + 0{,}5$. Perkirakan fungsi tersebut dengan menggunakan deret Taylor order nol, satu, dua dan tiga pada titik $x_{i+1}=1$, berdasar nilai fungsi pada titik $x_i = 0$. Titik x_{i+1} berada pada jarak $\Delta x = 1$ dari titik $x_i = 0$.

Adapun bentuk penyelesaian, karena bentuk fungsi sudah diketahui, maka dapat dihitung nilai f(x) antara 0 dan 1.

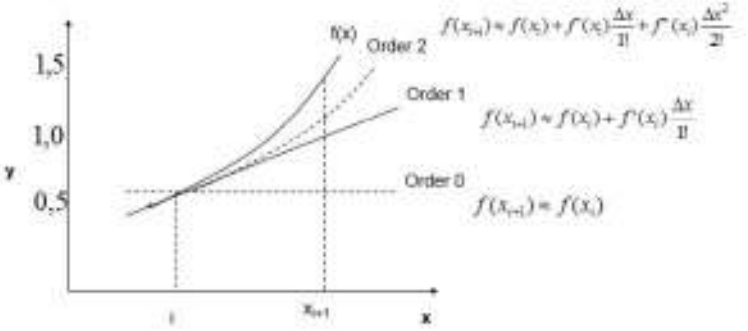
$$Untuk\ x_i = 0, maka\ f(x = 0) = 0{,}25(0)^3 + 0.5(0)^2 + 0{,}25(0) + 0{,}5 = 0{,}5$$

$Untuk\ x_{i+1} = 1, maka\ f(x_{i+1} = 1) = 0{,}25(1)^3 + 0{,}5(1)^2 + 0{,}25(1) + 0{,}5 = 1{,}5$

Jadi nilai eksak untuk f(x=1) adalah 1,5. apabila digunakan deret taylor order nol, maka persamaan didapat:

$$f(x_{i+1} = 1) \approx f(x_i = 0) \approx 0{,}5$$

Bentuk dalam implementasi persamaan dalam perkiraan orde nol, nilai f pada titik x_{i+1} sama dengan nilai pada x_i. Perkiraan tersebut adalah benar jika fungsi yang diperkirakan adalah suatu konstan. Jika fungsi tidak konstan, maka harus diperhitungkan suku-suku berikutnya dari deret Taylor.



Gambar. 2.2. Tampilan Perkiraan dari Setiap Order Fungsi Dengan Deret Taylor

Melihat dari grafik diatas, maka nilai eksak untuk f(x=1) adalah 1,5. Apabila digunakan deret Taylor order nol, maka berdasarkan deret taylor didapat:

$$f(x_{i+1} = 1) \approx f(x_i = 0) \approx 0{,}5$$

Seperti tampak pada grafik deret taylor, perkiraan order nol adalah konstan, dan kesalahan pemotongannya adalah:

$$\varepsilon_m = a - \hat{a} = 1{,}5 - 0{,}5 = 1{,}0$$

Apabila digunakan deret taylor order satu, nilai $f(x_{i+1} = 1)$ dapat dihitung dengan menggunakan persamaan order satu. Pertama kali dihitung turunan fungsi di titik $x_i = 0$, melihat dari fungsi berikut.

$f(x) = 0{,}25x^3 + 0{,}5x^2 + 0{,}25x + 0{,}5$

$$f'(x_i = 0) = 0{,}75x^2 + x + 0{,}25 = 0{,}75(0)^2 + 0 + 0{,}25 = 0{,}25$$

Sehingga diperoleh,

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{\Delta x}{1!} = 0{,}5 + 0{,}25 \times \frac{1}{1} = 0{,}75$$

Perkiraan order satu adalah garis lurus, Kesalahan pemotongan:

$$\varepsilon_a = a - \hat{a} = 1{,}5 - 0{,}75 = 0{,}75$$

Apabila digunakan deret taylor order dua, nilai $f(x_{i+1} = 1)$ dapat dihitung dengan menggunakan persamaan galat pemotongan, dihitung turunan kedua fungsi di titik $x_i = 0$, maka diperoleh perhitungan dari fungsi berikut.

$$f'(x_i = 0) = 0{,}75x^2 + x + 0{,}25$$

$$f''(x_i = 0) = 1{,}5x + 1 = 1{,}5(0) + 1 = 1{,}0$$

Sehingga diperoleh,

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{\Delta x}{1!} + f''(x_i)\frac{\Delta x^2}{2!}$$

$$\approx 0{,}5 + 0{,}25(1) + 1{,}0\left(\frac{1}{1\times 2}\right) = 1{,}25$$

Perkiraan order dua adalah garis lengkung, Kesalahan pemotongan:

$\varepsilon_a = a - \hat{a} = 1{,}5 - 1{,}25 = 0{,}25$

Apabila digunakan deret taylor order ketiga, nilai $f(x_{i+1} - 1)$ dapat dihitung dengan menggunakan persamaan galat pemotongan, dihitung turunan kedua fungsi di titik $x_i = 0$, maka diperoleh perhitungan dari fungsi berikut.

$f'(x_i = 0) = 0{,}75x^2 + x + 0{,}25$

$f''(x_i = 0) = 1{,}5x + 1 = 1{,}5(0) + 1 = 1{,}0$

$f'''(x_i = 0) = 1{,}5 = 1{,}5$

Sehingga diperoleh,

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{\Delta x}{1!} + f''(x_i)\frac{\Delta x^2}{2!} + f'''(x_i)\frac{\Delta x^3}{3!}$$

$$\approx 0{,}5 + 0{,}25(1) + 1{,}0\left(\frac{1}{1\times 2}\right) + 1{,}5\left(\frac{1}{1\times 2\times 3}\right) = 1{,}5$$

Sehingga kesalahan pemotongan diperoleh,

$\varepsilon_a = a - \hat{a} = 1{,}5 - 1{,}5 = 0$

Terlihat bahwa dengan menggunakan deret taylor order tiga, hasil penyesuaian numerik sama dengan penyelesaian eksak.

b. Galat Pembulatan

Pembulatan maksudnya mengurangi cacah digit pada suatu nilai hampiran dengan cara membuang beberapa digit terakhir. Cara melakukan pembulatan sudah dijelaskan di depan. Galat pembulatan terjadi disebabkan karena adanya pembulatan dalam komputasi numerik, sehingga pengulangan pembulatan tidak disarankan dalam komputasi numerik karena hanya akan memperbesar galat.

2.5. Program MATLAB

Adapun pembahasan MATLAB mengembangkan metode perhitungan dalam sintaksnya dibahas sebagaimana berikut ini. Bentuk pembahasan dalam bentuk perhitungan galat pemotongan dengan perkirakan suatu fungsi tertentu dengan menggunakan program MATLAB menggunakan *command window* galat absolut relatif dan galat pemotongan deret taylor.

a. Galat Pemotongan absolut dan relatif

Tahap simulasi dengan menggunakan program MATLAB, mengacu dari penyelesaian hitungan kesalahan yang terjadi pada nilai e^x apabila hanya diperhitungkan beberapa suku pertama saja. Nilai eksak dari $e^{0,5} = 1{,}648721271$.

Nilai e^x dapat dihitung berdasarkan deret berikut ini.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

Kesalahan relatif terhadap nilai eksak diperhitungkan dengan menggunakan formula:

$$\varepsilon_R = \frac{\varepsilon}{a} \times 100\ \%$$

Dengan menggunakan program MATLAB, maka memperhitungkan satu suku pertama dengan langkah sebagai berikut.

Gambar. 2.3. Hasil penerapan MATLAB dengan fungsi nilai e^x pada suku pertama

Pada tahapan dua suku pertama,$e^x = 1 + x$, maka untuk nilai x = 0,5 sehingga diperoleh hasil$e^x = 1 + 0{,}5 = 1{,}5$

Dalam tahapan ini memperhitungkan hasil dengan melihat hasil dari kesalahan relatif dan galat relatif hampiran.

$\varepsilon_R = \frac{\varepsilon}{a} \times 100\ \%$ dan $\varepsilon_{RA} = \frac{\varepsilon}{\hat{a}} \times 100\ \%$

Dengan menggunakan program MATLAB, maka memperhitungkan dua suku pertama dengan langkah sebagai berikut.

Gambar. 2.4. Hasil Penerapan MATLAB dengan Fungsi Nilai e^x pada Dua Suku Pertama

Gambar. 2.5. Hasil Penerapan MATLAB dengan Fungsi Nilai e^x pada Dua Suku Pertama

Pada tahapan tiga suku pertama, $e^x = 1 + x + \frac{x^2}{2!}$, maka untuk nilai x = 0,5 sehingga diperoleh hasil $e^x = 1 + 0{,}5 + \frac{0.5^2}{2!} = 1{,}625$

Dalam tahapan ini memperhitungkan hasil dengan melihat hasil dari kesalahan relatif dan galat relatif hampiran. Adapun penggunaan program MATLAB, maka memperhitungkan dua suku pertama dengan langkah sebagai berikut.

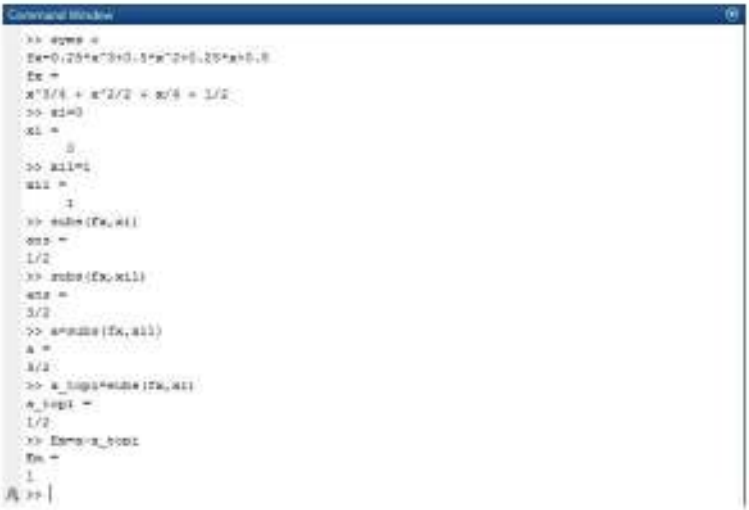
Gambar. 2.6. Hasil Penerapan MATLAB dengan Fungsi Nilai e^x pada Satu Suku Kedua

Gambar. 2.7. Hasil Penerapan MATLAB dengan Fungsi Nilai e^x pada Satu Suku Kedua

b. Galat Pemotongan Deret Taylor

Tahap simulasi dengan menggunakan program MATLAB, mengacu dari penyelesaian hitungan kesalahan yang terjadi pada nilai suatu fungsi f(x) = $0{,}25x^3 + 0{,}5x^2 + 0{,}25x + 0{,}5$. Perkirakan fungsi tersebut dengan menggunakan deret taylor order nol, satu, dua dan tiga pada titik $x_{i+1}=1$, berdasar nilai fungsi pada titik $x_i = 0$. Titikx_{i+1} berada pada jarak $\Delta_x = 1$ dari titik $x_i=0$.

Adapun bentuk penyelesaian dengan bentuk simulasi dengan program MATLAB, sesuai dengan bentuk fungsi sudah diketahui, maka dapat dihitung nilai f(x) antara 0 dan 1. Hasil dari perhitungan dengan program MATLAB sebagai berikut.



Gambar. 2.8. Hasil penerapan MATLAB dengan fungsi deret taylor pada order nol

Seperti tampak pada gambar deret taylor, perkiraan order nol adalah konstan, dan kesalahan pemotongannya adalah:

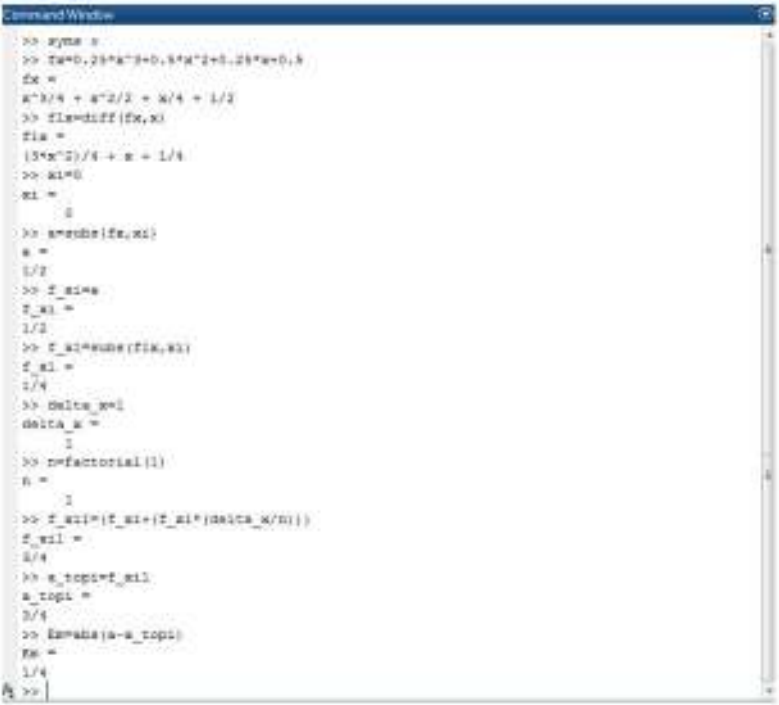
$$\varepsilon_m = a - \hat{a} = 1{,}5 - 0{,}5 = 1{,}0$$

Adapun perhitungan simulasi MATLAB pada deret taylor order satu, nilai $f(x_{i+1} = 1)$ dapat dihitung dengan menggunakan persamaan order satu. Pertama kali dihitung turunan fungsi di titik $x_i = 0$, Hasil dari perhitungan dengan program MATLAB sebagai berikut.

Adapun terlebih dahulu dalam penggunaan pemrograman MATLAB, pengguna terlebih dahulu membuat pemisalan variabel. Dalam simulasi ini, pemisalan dilakukan dengan tabel variabel sebagai berikut.

Tabel. 2.2. Tampilan variabel dalam MATLAB

Dalam analitik	Dalam MATLAB
f(x)	fx
f ' (x)	f1x
$(x_{i=0})$	xi=0
$f(x_{i+1})$	f_xi1
Δx	delta_x
$n!$	factorial(n)
$\hat{a}$	a_topi



Gambar. 2.9. Hasil Penerapan MATLAB dengan Fungsi Deret Taylor pada Order Satu

Seperti tampak pada gambar deret taylor, perkiraan order dua adalah Perkiraan order satu adalah garis lurus, Kesalahan pemotongan:

$$\varepsilon_a = a - \hat{a} = 1{,}5 - 0{,}75 = 0{,}75$$

Adapun perhitungan simulasi MATLAB pada deret taylor order dua, nilai $f_{(xi+1\ =\ 1)}$ dapat dihitung dengan menggunakan persamaan order satu. Pertama

kali dihitung turunan fungsi di titik $x_i = 0$, Hasil dari perhitungan dengan program MATLAB sebagai berikut.

Kedua fungsi di titik $x_i = 0$, maka diperoleh perhitungan dari fungsi berikut.

$$f'(x_i = 0) = 0{,}75x^2 + x + 0{,}25$$

$$f''(x_i = 0) = 1{,}5x + 1 = 1{,}5(0) + 1 = 1{,}0$$

Sehingga diperoleh,

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)\frac{\Delta x}{1!} + f''(x_i)\frac{\Delta x^2}{2!}$$

$$\approx 0{,}5 + 0{,}25(1) + 1{,}0\left(\frac{1}{1\times 2}\right) = 1{,}25$$

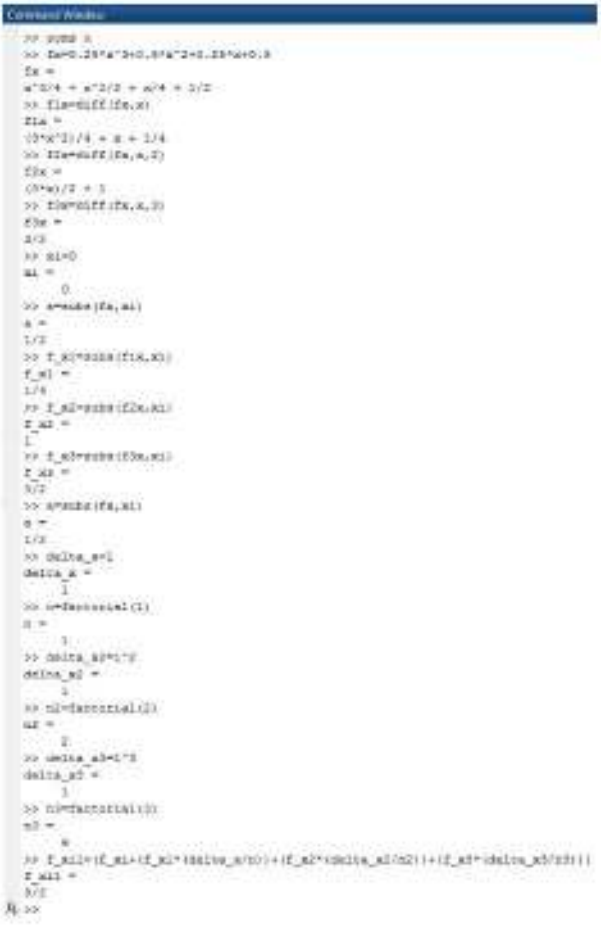
```
Command Window
>> syms x
>> fx=0.25*x^3+0.5*x^2+0.25*x+0.5
fx =
x^3/4 + x^2/2 + x/4 + 1/2
>> f1x=diff(fx,x)
f1x =
(3*x^2)/4 + x + 1/4
>> f2x=diff(fx,x,2)
f2x =
(3*x)/2 + 1
>> x1=0
x1 =
     0
>> a=subs(fx,x1)
a =
1/2
>> f_x1=subs(f1x,x1)
f_x1 =
1/4
>> delta_x=1
delta_x =
     1
>> n=factorial(1)
n =
     1
>> delta_x2=1^2
delta_x2 =
     1
>> n2=factorial(2)
n2 =
     2
>> f_x2=subs(f2x,x1)
f_x2 =
1
>> f_x1=(f_x1+(f_x1*(delta_x/n))+(f_x2*(delta_x2/n2)))
f_x1 =
5/4
>> a_topi=f_x1
a_topi =
5/4
>> Em=abs(a-a_topi)
Em =
3/4
fx >>
```

Gambar. 2.10. Hasil Penerapan MATLAB dengan Fungsi Deret Taylor pada Order Dua

Perkiraan galat order dua dalam pemrograman MATLAB, diperoleh:

$$\varepsilon_m = a - \hat{a} = 1{,}5 - 1{,}25 = 0{,}25$$

Adapun perhitungan simulasi MATLAB pada deret taylor order tiga dengan program MATLAB sebagai berikut.



Gambar. 2.11. Hasil Penerapan MATLAB dengan Fungsi Deret Taylor pada Order Tiga

Sehingga kesalahan pemotongan diperoleh,

$$\varepsilon_m = a - \hat{a} = 1{,}5 - 1{,}5 = 0$$

Terlihat bahwa dengan menggunakan deret taylor order tiga, hasil penyesuaian numerik sama dengan penyelesaian eksak.

SIMULASI 3: AKAR PERSAMAAN TAK LINIER

3.1. Pendahuluan

Salah satu masalah yang paling umum ditemui dalam matematika adalah mencari akar suatu persamaan. Jika diketahui fungsi f(x), akan dicari nilai-nilai x yang memenuhi f(x) = 0. Termasuk dalam masalah menentukan titik potong dua buah kurva. Apabila kurva-kurva tersebut dinyatakan oleh fungsi f(x) dan g(x), maka absis titik potong kedua kurva tersebut merupakan akar-akar persamaan f(x) – g(x) = 0

Adapun dalam mempelajari bab ini dengan menggunakan beberapa metode untuk mencari akar-akar suatu persamaan. Untuk persamaan polinomial derajat dua, persamaan dapat diselesaikan dengan rumus persamaan kuadrat yang sangat sederhana. Misalnya bentuk persamaan $ax^2+bx+c = 0$, dapat dicari akar-akarnya secara analitis dengan rumus sebagai berikut:

$$x_{12} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Tetapi apabila persamaan polinomial derajat tertinggi ≥ 3 atau berbentuk transeden seperti 1 + cos(x), x tan (x) – cosh (x), e^x-sin(x) dan lainnya. Untuk menentukan akar-akar persamaan tersebut belum tentu ada persamaan (rumus) eksak untuk menyelesaikannya. Tetapi dengan analisis numerik kita

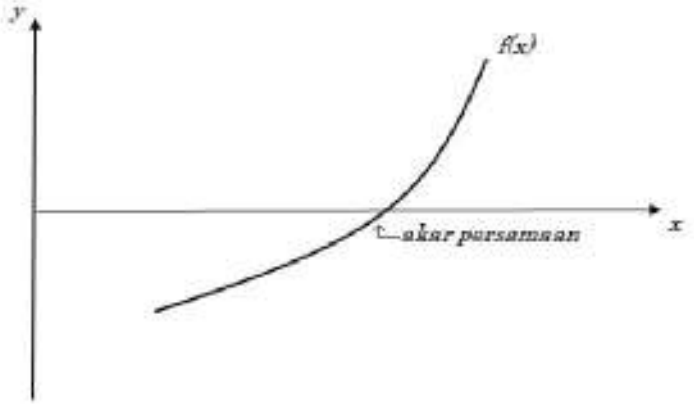
dapat menentukan akar-akar persamaan polinomial dengan cara pendekatan.

Karena metode yang dipergunakan merupakan metode pendekatan, maka akar persamaan polinom juga merupakan pendekatan. Sebelum menentukan menggunakan metode atau pendekatan apa yang digunakan untuk menyelesaikan persamaan polinom. Lebih baik diselidiki terlebih dahulu apa persamaan polinom tersebut benar-benar memiliki akar real. Dalam buku ini akan membahas penggunaan simulasi program MATLAB terhadap penyelesaian akar persamaan tak linier, yang akan dibahas pada materi ini.

Adapun bentuk penyelesaian numerik mengacu pada perkiraan yang berurutan (iterasi), sedemikian sehingga setiap hasil merupakan hasil yang lebih teliti dari perkiraan sebelumnya. Dengan melakukan sejumlah prosedur iterasi yang dianggap cukup, akhirnya didapat hasil perkiraan yang mendekati hasil eksak (hasil yang sejati/benar) dengan tingkat toleransi kesalahan yang diizinkan.

Salah satu cara yang paling sederhana untuk mendapatkan penyelesaian perkiraan adalah melalui cara menggambarkan fungsi tersebut dan kemudian dicari titik potongnya dengan sumbu x yang menunjukkan akar dari persamaan tersebut. Hal ini dapat memperhatikan Gambar 3.1 Menentukan akar persamaan secara grafis. Namun, langkah tahap ini hanya menunjukkan hasil sekilas dan kurang rinci. Hal

ini, dikarenakan sulit untuk menetapkan nilai sampai beberapa digit dibelakang koma hanya dengan memperkirakan dari membaca gambar. Metode lain untuk menyelesaikan persamaan tersebut adalah dengan langkah membandingkan, yaitu dengan mencoba nilai x sembarang kemudian dilakukan evaluasi apakah nilai f(x) = 0. Jika nilai x tidak sama dengan nol, maka kemudian dicoba nilai x yang lain. Prosedur ini dilakukan ulang secara terus sampai dengan nol kemudian dicoba nilai x yang lain. Prosedur ini dilakukan ulang secara terus sampai akhirnya didapatkan nilai f(x) = 0, untuk suatu nilai x tertentu, yang merupakan akar dari persamaan yang diselesaikan.



Gambar. 3.1. Menentukan Akar Persamaan Secara Grafis

Kedua cara tersebut merupakan langkah yang tidak efisien dan tidak sistematis. Ada beberapa metode

yang merupakan penyelesaian perkiraan, tetapi lebih sistematis untuk menghitung akar-akar persamaan. Metode-metode tersebut akan dipelajari pada simulasi materi ini.

3.2. Metode Tabel

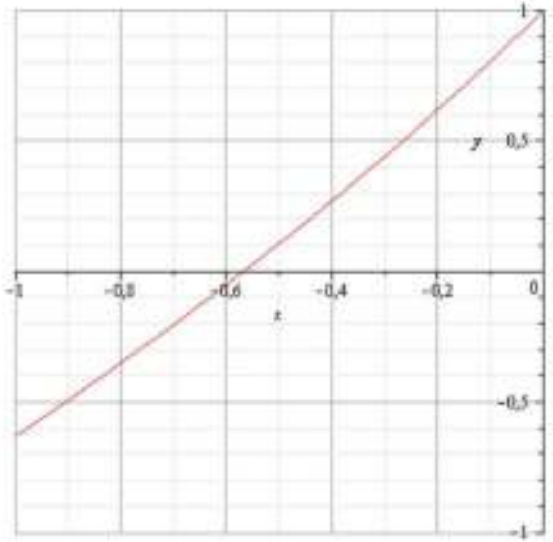
Metode pertama ini akan dibahas berdasarkan atas teknis penyelesaian suatu range x=[a,b] mempunyai akar bila f(a) dan f(b) berlawanan tanda atau memenuhi f(a),f(b) < 0. Metode tabel merupakan bentuk sederhana diantara beberapa metode yang akan dipelajari Secara sederhana, untuk menyelesaikan persamaan non-linier dapat dilakukan dengan menggunakan metode tabel atau pembagian area, dimana untuk x=[a,b] atau x di antara a dan b dibagi sebanyak N bagian dan pada masing-masing bagian dihitung nilai f(x) sehingga diperoleh tabel berikut.

Tabel. 3.1. Tampilan Metode Tabel

x	f(x)
x_0=a	f(a)
x_1=a	f(x_1)
x_2=a	f(x_2)
...	...
x_n=a	f(b)

Apabila ditemukan f(x_k)=0 atau mendekati nol maka dikatakan adalah penyelesaian. Bila tidak ada f(x_k)=0maka dicari nilai f(x_k) dan f(x_{k+1}) yang berlawanan tanda Bila tidak ditemukan, dikatakan tidak memiliki akar untuk x=[a,b].

Perhatikan contoh permasalahan berikut ini. Selesaikan persamaan$x + e^x = 0$ dengan x = [-1,0] dengan menggambarkan grafik persamaan $x + e^x = 0$ diperoleh hasil sebagai berikut.



Gambar. 3.2. Hasil Grafik dengan Persamaan $x + e^x = 0$

Adapun bentuk penjajaran perhitungan dalam persamaan $x + e^x = 0$, diperoleh dalam bentuk tabel sebagai berikut dengan rentangan internal $\Delta x = 0.1$ (diperoleh dari pembagian 10 bagian)

Tabel. 3.2. Hasil Metode Tabel dengan 10 Bagian

X	f(x)
-1,0	-0,63212
-0,9	-0,49343
-0,8	-0,35067
-0,7	-0,20341
-0,6	-0,05119
-0,5	0,10653
-0,4	0,27032
-0,3	0,44082
-0,2	0,61873
-0,1	0,80484

10 Bagian

Dari tabel 3.2. diperoleh penyelesaian di antara -0.6 dan -0.5 dengan nilai masing-masing -0,05119 dan 0,10653 sehingga dapat diambil keputusan penyelesaian di x=-0.6 sd x=-0.5 Perolehan hasil sebagai berikut pada tabel 3.3.

Tabel. 3.3. Pengambilan Keputusan x=[-1,0] dengan 10 Bagian

X	f(x)
-1,0	-0,63212
-0,9	-0,49343
-0,8	-0,35067
-0,7	-0,20341
-0,6	-0,05119
-0,5	0,10653
-0,4	0,27032
-0,3	0,44082
-0,2	0,61873
-0,1	0,80484

10 Bagian

Adapun x=[-0.6,-0.5] dibagi dengan banyaknya 10 bagian sehingga diperoleh tabel 3.4. berikut.

Tabel 3.3. Pengambilan Keputusan x=[-0.6,-0.5] dengan 10 Bagian

X	f(x)
-0,60	-0,05119
-0,59	-0,03567
-0,58	-0,02010
-0,57	-0,00447
-0,56	0,01121
-0,55	0,02695
-0,54	0,04275
-0,53	0,05860
-0,52	0,07452
-0,51	0,09050

10 Bagian

keputusan penyelesaian pada range x di bagi dengan banyaknya bagian diperoleh f(x) terdekat dengan nol pada x = -0.57 dengan nilai f(x) = -0.00447.

Sehingga melihat dari contoh permasalahan diatas, kita dapat menyelesaikan persamaan akar persamaan non linier dengan menggunakan analisis numerik dengan pendekatan metode tabel. Adapun bentuk algoritma metode tabel, ditunjukkan sebagai berikut.

Algoritma Metode Tabel.

Adapun tahapan dalam penyelesaian dengan metode tabel, dengan aturan algoritma sebagai berikut.

1. Definisikan fungsi f(x).
2. Menentukan range untuk x yang merupakan batas bawah (x_{bawah}) dan batas atas (x_{atas}).
3. Tentukan jumlah pembagian N.
4. Hitung langkah pembagi h. → $H = \frac{x_{atas} - x_{bawah}}{N}$
5. Untuk i=0 sampai dengan N, hitung →
6. Untuk I=0 sampai dengan N, dicari k dimana

$$x_i = x_{bawah} + ih$$
$$y_i = f(x_i)$$

Catatan:

Bila $f(x_k) = 0$, maka x_k adalah solusi penyelesaian, namun apabila $f(x_k)$. $f(x_{k+1}) < 0$, maka penyelesaian tergantung dari:

Bila $| f(x_k)| < | f(x_{k+1})| = 0$ maka x_k adalah solusi penyelesaian.

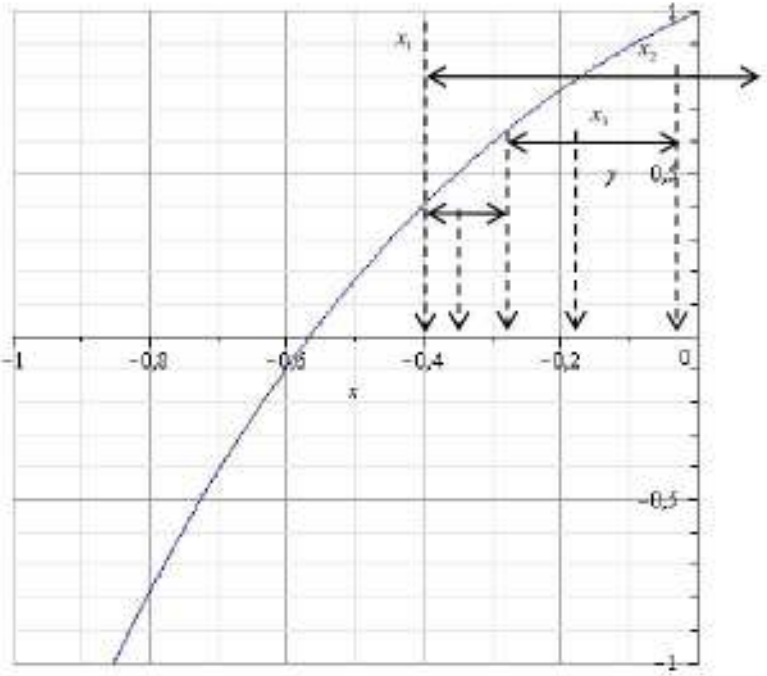
Apabila tidak, (x_{k+1}) adalah penyelesaian atau dapat dikatakan penyelesaian berada di (x_k) dan (x_{k+1}).

3.3. Metode Biseksi (*Bisection Method*)

Metode berikut ini akan dibahas berdasarkan atas teknis metode yang dinamakan metode setengah interval atau metode belah dua. Metode biseksi merupakan bentuk sederhana diantara beberapa metode yang akan dipelajari.

Penggunaan metode tabel, dimana areanya dibagi menjadi N bagian. Metode ini membagi menjadi 2 bagian, sehingga dari dua bagian yang terpilih tidak mengandung akar yang dibuang. Hal ini dilakukan berulang hingga diperoleh akar persamaan. Terlebih dahulu menentukan batas bawah (a) dan batas atas (b). Kemudian menentukan nilai tengah.

Adapun bentuk grafis menentukan akar persamaan dengan metode biseksi akan digambarkan pada Gambar 3.2, yang mana menentukan prosedur perhitungan secara grafis.



Gambar. 3.3. Grafik Metode Biseksi

Berdasarkan pada grafis terlihat bahwa pada Teorema Nilai Rata-rata pada kalkulus, misalkan f(x) suatu fungsi kontinu pada interval tertutup [a,b] sedemikian sehingga f(a) dan f(b) berlawanan tanda (f(a).f(b) < 0), maka terdapat suatu akar persamaan f(x)

= 0 pada interval (a,b). Pada setiap kali iterasi, selang [a,b] kita bagi dua di x = c, sehingga terdapat dua upselang yang berukuran sama yaitu [a,c] dan [b,c]. Selang yang diambil untuk iterasi berikutnya adalah upselang yang memuat akar, bergantung apakah f(a).f(b)<0 atau f(c).f(b)<0. Selang yang baru dibagi dua lagi dengan cara yang sama, begitu seterusnya sampai ukuran selang yang baru sudah sangat kecil.

Dari nilai x tersebut, perlu dilakukan pengecekan keberadaan akar. Secara matematis, di dalam suatu range terdapat akar persamaan bila f(a) dan f(b) berlawanan tanda atau dituliskan:

$$f(a).f(b) < 0$$

Setelah diketahui di bagian mana terdapat akar, batas bawah dan batas atas diperbarui sesuai dengan range bagian yang mempunyai akar.

Lebih memahami Perhatikan contoh permasalahan berikut, yang mengacu dari metode tabel. Selesaikan persamaan$xe^{-x} + 1 = 0$ denganx=[-1,0].

Adapun hasil perhitungan dengan metode biseksi, dengan memperhatikan tanda pada tabel perhitungan pada tabel. 3.4. sebagai berikut.

Tabel 3.4. Perhitungan Metode Biseksi.

iterasi	a	b	x	f(x)	f(a)	Keterangan
1	-1	0	-0.5	0.175639	-1.718282	*berlawanan tanda*
2	-1	-0.5	-0.75	-0.587750	-1.718282	
3	-0.75	-0.5	-0.625	-0.167654	-0.587750	
4	-0.625	-0.5	-0.5625	0.012782	-0.167654	*berlawanan tanda*
5	-0.625	-0.5625	-0.59375	-0.075142	-0.167654	

iterasi	a	b	x	f(x)	f(a)	Keterangan
6	-0.598876	-0.5625	-0.579313	-0.030619	-0.075142	
7	-0.579313	-0.5625	-0.570531	-0.008780	-0.030619	
8	-0.570531	-0.5625	-0.56641	0.002035	+0.008780	*berlawanan tanda*
9	-0.57031	-0.56641	-0.56836	-0.003364	-0.008789	
10	-0.56836	-0.56641	-0.56738	-0.000662	-0.003364	Iterasi berhenti

Sehingga melihat dari contoh permasalahan diatas, kita dapat menyelesaikan persamaan akar persamaan non linier dengan menggunakan analisis numerik dengan pendekatan metode biseksi. Adapun bentuk algoritma metode biseksi, ditunjukkan sebagai berikut.

Algoritma Metode Biseksi (Setengah Interval)

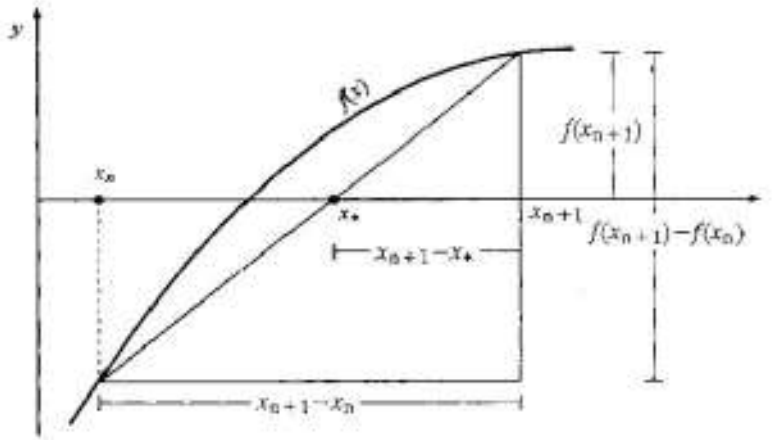
1. Hitung fungsi interval yang sama dari x sampai pada perubahan tanda dari fungsi $f(x_i)$ dan $f(x_{i+1})$, yaitu apabila $f(x_i) \times f(x_{i+1}) < 0$.
2. Estimasi pertama dari akar x_t dihitung dengan $x_t = \{ x_i + x_{i+1} \}$.
3. Buat evaluasi berikut untuk menentukan di dalam sub interval mana akar persamaan berada:
   a. Jika $f(x_i) \times f(x_t) < 0$, akar persamaan berada pada sub interval pertama, kemudian tetapkan $x_{i+1} = x_t$ dan lanjutkan pada langkah ke-4.

b. Jika $f(x_i) \times f(x_t) > 0$, akar persamaan berada pada sub interval kedua, kemudian tetapkan $x_i = x_t$ dan lanjutkan pada langkah ke-4.
c. Jika $f(x_i) \times f(x_t) = 0$, akar persamaan adalah x_t dan hitungan selesai.

4. Hitung perkiraan baru dari akar dengan persamaan.
5. Apabila perkiraan baru sudah cukup kecil (sesuai dengan batasan yang ditentukan), maka hitungan selesai, dan x_t adalah akar persamaan yang dicari. jika belum, maka hitungan kembali ke langkah 3.

3.4. Metode Interpolasi Linier (*False Position Method*)

Metode berikut ini akan dibahas berdasarkan atas teknis metode yang dinamakan metode *false position* atau posisi salah/palsu. Dalam permasalahan sebelumnya melihat hasil perhitungan dengan menggunakan metode setengah interval atau metode biseksi merupakan suatu teknik yang sempurna dan berlaku secara sempurna untuk menentukan akar-akar pendekatan, tetapi akar-akar pendekatannya kurang efisien jika metode biseksi digunakan. Cari nilai fungsi untuk setiap interval Δx (menghitung nilai f (x) pada interval antara dua titik sedemikian sehingga nilai f (x)pada kedua titik tersebut berlawanan tanda).



Gambar. 3.3. Grafik Metode Interpolasi Linier

Kedua nilai fungsi tersebut ditarik garis lurus hingga terbentuk suatu segitiga, dengan menggunakan sifat segitiga sebangun didapat persamaan berikut.

$$\frac{x_{i+1}-x_*}{x_{i+1}-x_i}=\frac{f(x_{i+1})}{f(x_{i+1})-f(x_i)}$$

$$x_*=x_{i+1}-\frac{f(x_{i+1})}{f(x_{i+1})-f(x_i)}(x_{i+1}-x_i)$$

Gunakan lagi untuk interpolasi linier dengan nilai f (x_i) atau f (x_{i+1}) sedemikian sehingga kedua fungsi mempunyai tanda berbeda, Langkah di atas diulang sampai tingkat ketelitian yang diinginkan.

Lebih memahami Perhatikan contoh permasalahan berikut, yang mengacu dari metode interpolasi linier. Perhatikan permasalahan berikut ini.

Hitung salah satu akar dari persamaan pangkat tiga pada interval antara dua titik $x_1 = 1$ dan $x_2 = 2$berikut ini pada: $f(x) = x^3 + x^2 - 3x - 3$. dengan tingkat ketelitian 0,0001.

Penyelesaian terhadap fungsi dari $f(x) = x^3 + x^2 - 3x - 3$. dengan tingkat ketelitian 0,0001 maka dimulai dari hitungan nilai f(x) pada interval antara dua titik $x_1 = 1$ dan $x_2 = 2$.

Untuk x_1 :

$$f(x_1) = (1)3 + (1)2 - 3(1) - 3 = -4$$

Untuk x_2 :

$$f(x_2) = (2)3 + (2)2 - 3(2) - 3 = 3$$

Dengan menggunakan persamaan segitiga di atas maka, didapat:

$$x_* = x_{i+1} - \frac{f(x_{i+1})}{f(x_{i+1}) - f(x_i)}(x_{i+1} - x_i) = 2 - \frac{3}{3+4}(2-1) = 1{,}57142$$

$f(x_*) = (1{,}57142)3 + (1{,}57142)2 - 3(1{,}57142) - 3 = -1{,}36449.$

Karena $f(x_*)$ bertanda negatif maka akar terletak antara $x = 1{,}57142$ dan $x = 2$, selanjutnya dihitung nilai x_*:

$x_* = 1{,}70540$

$f(x_*) = (1{,}70540)3 + (1{,}70540)2 - 3(1{,}70540) - 3 = -0{,}24787.$

Untuk memudahkan perhitungan maka menggunakan pemrograman komputer, hasil hitungan tersebut diatas ada pada Tabel berikut dan didapat pada iterasi ke 7,yaitu $x_* = 1{,}73205$.

Tabel 3.5. Perhitungan Metode Interpolasi Linier.

I	x_i	x_{i+1}	x_*	$f(x_i)$	$f(x_{i+1})$	$f(x_*)$
1	1,0000	2,0000	1,5714	-4,0000	3,0000	-1,3644
2	1,5714	2,0000	1,7054	-1,3644	3,0000	-0,2477
3	1,7054	2,0000	1,7279	-0,2477	3,0000	-0,0393
4	1,7279	2,0000	1,7314	-0,0393	3,0000	-0,0061
5	1,7314	2,0000	1,7320	-0,0061	3,0000	-0,0009
6	1,7320	2,0000	1,7320	-0,0009	3,0000	-0,0001
7	**1,7320**	2,0000	1,7320	-0,0001	3,0000	0,0000

Melihat hasil perhitungan dari tabel 3.5 maka diperoleh pada iterasi ke-7 perolehan $f(x_*) = 0.0000$ maka berhenti dikarenakan akar persamaan telah didapatkan. Melihat dari hubungan titik dari $(x_0, f(x_0))$ dan $(x_2, f(x_2))$ terlihat laju kekonvergenan metode ini lebih cepat dibandingkan metode biseksi.

Algoritma Interpolasi Linier

1. Hitung fungsi interval Δx yang sama sehingga didapat dari fungsi $f(xn)$ dan $f(x_{n+1})$, dengan tanda berbeda.
2. Hitung x^* dan $f(x^*)$.
3. Apakah $f(x^*)$ dan $f(xn)$ bertanda sama.
   3.1. Jika sama maka $x_{n+1} = x^*$ memiliki persamaan dengan $f(x_{n+1}) = f(x_n)$.
   3.2. Jika tidak sama $x^* = x_n$ memiliki nilai persamaan dengan $f(x_n) = f(x^*)$.
4. Apakah nilai $f(x^*)$ kecil (mendekati nol), jika iya perhitungan selesai.

5. Apabila perkiraan nilai f(x*) tidak lebih kecil, ulangi perhitungan langkah ke 2.

3.5. Metode Newton Rhapson (*Tangent Method*)

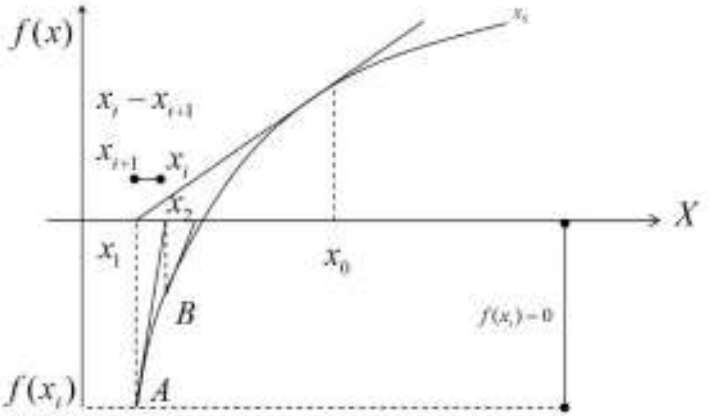
Metode berikut ini akan dibahas berdasarkan atas teknis metode yang dinamakan metode *tangent*. Dalam permasalahan sebelumnya melihat hasil perhitungan dengan menggunakan metode tabel, metode biseksi dan metode interpolasi merupakan suatu teknik yang sempurna dan berlaku secara sempurna untuk menentukan akar-akar pendekatan, tetapi akar-akar pendekatannya kurang efisien ataupun bentuk metode kurang sederhana. Metode ini akan menghasilkan akar pendekatan yang lebih baik dibandingkan metode sebelumnya dan memberikan iterasi lebih sederhana. Metode ini memiliki kinerja relatif jauh lebih cepat dalam pencapaiannya konvergensi. Hal ini dikarenakan memiliki laju konvergensi kuadrat. Adapun metode Newton Rhapson dapat bekerja dengan optimal, harus memperhatikan beberapa prasyarat sebagai berikut:

1. Diperlukan satu nilai awal (dapat berupa tebakan), dan tebakan nilai awal tersebut tidak menyebabkan nilai fungsi menjadi tak terhingga ∞.
2. Persamaan $y = f(y)$ mempunyai turunan yang dapat disebut sebagai $y' = f'(y)$ dan harus kontinu di daerah domain jawab.
3. Turunan fungsi tersebut tidak bernilai nol, $y' \neq 0$, pada harga x^k (pada iterasi ke-*k* yang diinginkan.

Metode newton-raphson adalah metode pendekatan yang menggunakan satu titik awal dan mendekatinya dengan memperhatikan *slope* atau *gradien* pada titik tersebut. Titik pendekatan i+1 dituliskan dengan:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Adapun bentuk grafis menentukan akar persamaan dengan metode newton rhapson akan digambarkan pada Gambar 3.4, yang mana menentukan prosedur perhitungan secara grafis.



Gambar. 3.4. Grafik Metode Newton Rhapson

Lebih memahami Perhatikan contoh permasalahan berikut, yang mengacu dari metode interpolasi linier. Perhatikan permasalahan berikut ini. Hitung salah satu akar dari persamaan pangkat tiga

pada interval antara dua titik $x_1 = 1$ dan $x_2 = 2$berikut ini pada: $f(x) = x^3 + x^2 - 3x - 3$. dengan tingkat ketelitian 0,0001 dengan menggunakan metode Newton Rhapson.

Langkah penyelesaian dengan metode newton rhapson, sebagai berikut.

Cara Penyelesaiannya:

1. Menentukan turunan pertama dari fungsi :
   $f'(x) = 3x^2 + 2x - 3$
2. Menghitung nilai x_{i+1}
   $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$
3. Sebelum menghitung x_{i+1} maka tentukan dulu nilai fungsi dan turunannya dengan nilai xi sembarang, misalnya $x_1 = 1$, maka
   $f(1) = (1)3 + (1)2 - 3(1) - 3 = -4$
   $f'(1) = 3(1)2 + 2(1) - 3 = 2$
4. Melanjutkan ke langkah b, yaitu menghitung fungsi dan turunannya dengan nilai $x_1 = x_{i+1}$
   $f(3) = (3)3 + (3)2 - 3(3) - 3 = 24.$
   $f'(3) = 3(3)2 + 2(3) - 3 = 30.$
5. Hitunglah sampai mendapatkan fungsi bernilai 0, untuk memudahkan gunakan program komputer

Adapun bentuk penyelesaian dengan permasalahan polinom diatas menggunakan metode Newton Rhapson dengan hasil perhitungan tabel berikut.

Tabel 3.6. Perhitungan Metode Newton-Raphson.

I	x_i	x_{i+1}	$f(x_i)=x^3+x^2-3x-3$	$f'(x_i)=3x^2+2x-3$	$f(x_{i+1})$
1	1,0000	3,0000	-4,0000	2,0000	24,0000
2	3,0000	2,2000	24,0000	30,0000	5,8888
3	2,2000	1,8302	5,8888	15,9200	0,9899
4	1,8302	1,7378	0,9989	10,7090	0,0555
5	1,7378	1,7321	0,0555	9,5350	0,0000
6	1,7321	1,7321	0,0000	9,4640	0,0000

Melihat hasil perhitungan dari tabel 3.6 maka diperoleh pada iterasi ke-6 perolehan f(x*) = 0.0000 maka berhenti dikarenakan akar persamaan telah didapatkan. Maka hasil pada iterasi ke-6 terdapat nilai fungsi bernilai 0 maka akar persamaan tersebut x = 1,7321

Algoritma Newton Raphson

1. Menentukan turunan pertama dari fungsi f(x).
2. Hitunglah x_{i+1} dengan menggunakan rumus ekivalen kemiringan.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

3. Tentukan nilai x_i sembarang hitung fungsi dan turunan pertama.
4. Lakukanlah langkah 2 sampai menghasilkan $f(x_{i+1})$ bernilai 0, akar persamaan adalah nilai xiyang terakhir diperoleh.

3.6. Program MATLAB

Adapun pembahasan MATLAB mengembangkan metode perhitungan dalam sintaksnya dibahas

sebagaimana berikut ini. Bentuk pembahasan dalam perhitungan akar persamaan perkirakan dengan menggunakan Metode Tabel, Metode Biseksi, Metode Interpolasi Linier dan Metode Newton Rhapson memanfaatkan program MATLAB menggunakan *command window* dan *File-M*. Adapun simulasi dengan menggunakan program MATLAB terbagi menjadi beberapa tahap berikut ini.

1. Metode Tabel

Adapun perhitungan simulasi MATLAB pada metode tabel, selesaikan persamaan $x+e^x=0$ dengan interval x = [-1,0] Pertama kali dihitung turunan fungsi di titik xi =0, Hasil dari perhitungan dengan program MATLAB sebagai berikut.

Adapun bentuk tampilan dalam penerapan File-M menggunakan penulisan skrip sebagai berikut.

```
clc
clear
syms x

disp('---- Metode Tabel ----')

% Masukkan Bentuk Fungsi f(x)
xa=input('Masukkan batas bawah x = ');
xb=input('Masukkan batas atas x = ');
n=input('Masukkan N pembagi = ');

% proses dalam pembagian interval nilai sebanyak N
disp('-----Rentangan nilai x dengan N pembagi-----')
x=linspace(xa,xb,n);
fprintf('%3f \n',x);
disp('Pilihlah nilai x pada baris pertama sebagai
acuan untuk memasukkan nilai x di bawah ini!')

disp('-----------Rentangan hasil nilai f (x) dengan N
pembagi-----------')
```

```
19. fx=input('Masukkan Fungsi (x) = ');
20. fprintf('%3f \n',fx);
21. disp('--------Simulasi Metode Tabel--------')
22. disp('Perhatikan dua baris yang hasil f(x) nya
    berlawanan!!!')
23. disp('Pilihlah nilai x pada baris pertama sebagai
    acuan untuk memasukkan nilai x di bawah ini!')
```

Adapun hasil yang telah diterapkan dalam penggunaan file-m pada gambar sebagai berikut.

Gambar. 3.5. Tampilan Hasil File-M Matlab Metode Tabel

Adapun bentuk dalam aplikasi luaran command window, tertampil sebagai berikut.

Gambar. 3.6. Tampilan Hasil Command Window Matlab Metode Tabel

2. Metode Biseksi

Adapun perhitungan simulasi MATLAB pada metode biseksi, menyelesaikan persamaan $xe^{-x} + 1 = 0$ dengan x=[-1,0]. Hasil dari perhitungan dengan program MATLAB sebagai berikut.

Adapun bentuk tampilan dalam penerapan File-M menggunakan penulisan skrip yang diadopsi dari Aamir Alaud Din dari situs milik mathwoks yaitu bisection.m yang mana dalam laman berikut https://www.mathworks.com/matlabcentral/fileexchange

/43535-bisection-m dan disesuaikan kembali dengan versi simulasi buku ini sebagai berikut.

```
function root = metode_biseksi(f,a,b,epsilon)
% Nama Program : metode_biseksi.m
%
% Tujuan Program : mencari akar persamaan dengan menggunakan metode biseksi
% Pengembang asli :Aamir Alaud Din
% Tanggal :17.09.2013
%
% Edit Ulang :Reza K. Setyansah
% Tanggal :27.03.2018
%
% Sintaks :akar = metode_biseksi(f,a,b)
%          dimana terlebih dahulu membuat persamaan fungsi untuk f(x)
%          contoh perhatikan : f = inline('x+exp(x)','x');
%          a dan b merupakan bentuk interval
%
% Contoh :akar = bisection(f,-1,0)

if (nargin < 3)
    error('Wrong number of input arguments.');
elseif (nargin == 3)
    if (f(a)*f(b) >= 0)
        disp('The function has positive value at the end points of interval.');
        disp('The root can''t be found using bisection method, use some other method.');
        root = 'Root can''t be found using bisection method';
        return;
    else
        fprintf('Iteration\t\t\ta\t\t\tb\t\t\tm\t\t\tf(a)\t\t\tf(b)\t\t\tf(m)\n');
        fprintf('=====\t\t=====\t\t=====\t\t=====\t\t======\t\t=====\t\t=====\n');
        m = (a + b)/2;
        if (abs(f(m)) <= 1e-6)
            root = m;
            return;
```

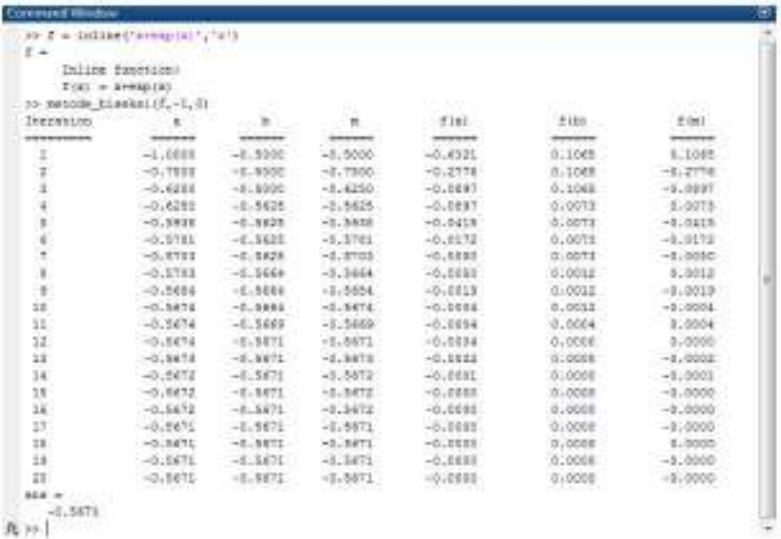
```
32.         else
33.             Iter = 0;
34.             while (abs(f(m)) >= 1e-6)
35.                 Iter = Iter + 1;
36.                 m = (a + b)/2;
37.                 if(f(a)*f(m) > 0)
38.                 a = m;
39.                 else
40.                 b = m;
41.                 end
42.                 fprintf('%3d',Iter);
43.                 fprintf('%20.4f',a);
44.                 fprintf('%12.4f',b);
45.                 fprintf('%12.4f',m);
46.                 fprintf('%14.4f',f(a));
47.                 fprintf('%16.4f',f(b));
48.                 fprintf('%16.4f',f(m));
49.                 fprintf('\n');
50.             end
51.             root = m;
52.         end
53.     end
54. elseif (nargin == 4)
55.     if (f(a)*f(b) >= 0)
56.         disp('The function has positive value at the
        end points of interval.');
57.         disp('The root can''t be found using bisection
        method, use some other method.');
58.         root = 'Root can''t be found using bisection
        method';
59.         return;
60.     else
61.         fprintf('Iteration\t\t\ta\t\t\tb\t\t\tm\t\t\tf
        (a)\t\t\tf(b)\t\t\tf(m)\n');
62.         fprintf('-------\t\t    -------\t\t    -------\t\t
        -------\t\t -------\t\t -------\t\t -------\n');
63.         m = (a + b)/2;
64.         if (abs(f(m)) <= epsilon)
65.             root = m;
66.             return;
67.         else
68.             Iter = 0;
69.             while (abs(f(m)) >= epsilon)
```

```
70.             Iter = Iter + 1;
71.             m = (a + b)/2;
72.             if(f(a)*f(m) > 0)
73.                a = m;
74.             else
75.                b = m;
76.             end
77.             fprintf('%3d',Iter);
78.             fprintf('%20.4f',a);
79.             fprintf('%12.4f',b);
80.             fprintf('%12.4f',m);
81.             fprintf('%14.4f',f(a));
82.             fprintf('%16.4f',f(b));
83.             fprintf('%16.4f',f(m));
84.             fprintf('\n');
85.           end
86.           root = m;
87.        end
88.     end
89. end
```

Adapun hasil yang telah diterapkan dalam penggunaan *file-m* pada gambar sebagai berikut.

Gambar. 3.7. Tampilan Hasil File-M Matlab Metode Biseksi

Adapun bentuk dalam aplikasi luaran *command window*, tertampil sebagai berikut.



Gambar. 3.8. Tampilan Hasil Command Window Matlab Metode Biseksi

3. Metode Interpolasi Linier (*False Position Method*)

Adapun perhitungan simulasi MATLAB pada metode interpolasi linier, menyelesaikan perhitungan salah satu akar dari persamaan pangkat tiga pada interval antara dua titik $x_1 = 1$ dan $x_2 = 2$ berikut ini pada: $f(x) = x^3 + x^2 - 3x - 3$ dengan tingkat ketelitian 0,0001 dengan program MATLAB sebagai berikut.

Perbedaan dalam proses penggunaan interpolasi linier dengan menggunakan command window dengan proses input, yang mana dalam penentuan fungsinya menggunakan bentuk file-m dengan menyimpan terlebih dahulu di file-m dengan format *mil.m*

```
1. function y=mil(x)
2. y=x^3+x^2-3*x-3;
```

Adapun bentuk tampilan perintah dalam penerapan File-M menggunakan penulisan skrip dari Rahmat Nawi Siregar dari situs milik youtube yaitu interpolasi_linier.m yang mana dalam laman berikut dan disesuaikan kembali dengan versi simulasi buku ini sebagai berikut.

```
% Nama Program : interpolasi_linier.m
%
% Tujuan Program : mencari akar persamaan dengan menggunakan metode biseksi
% Pengembang asli :Rahmat Nawi Siregar
% Tanggal :09.02.2017
%
% Edit Ulang :Reza K. Setyansah
% Tanggal :27.03.2018
%
% Sintaks :     masukkan   fungsi   yang   diketahui kedalam mil.m
%               dimana   terlebih   dahulu   membuat persamaan fungsi untuk f(x)
%               function y=mil(x)
%               y=x^3+x^2-3*x-3;
%               masukkan   dugaan   interval   yang diketahui

clear;
clc;
a=input('masukkan data nilai dugaan awal - ');
b=input('masukkan data nilai dugaan kedua - ');
tol=0.01;
maxstep=10;
disp('step     a     b     p     fa     fb     fp');
fa=mil(a);
fb=mil(b);
for i=1:maxstep
    p=b-((b-a)*(fb/(fb-fa))); % p=xstar
    fp=mil(p); %fp=
    fprintf('%3d  %8.5f  %8.5f  %8.5f  %8.5f  %8.5f  %8.5f\n', i,a,b,p,fa,fb,fp);
    if p<tol
```

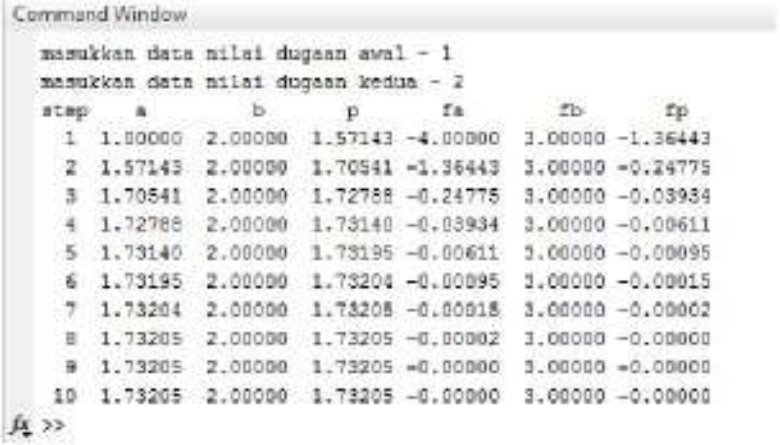
```
30.         fprintf('akar    persamaan    %g    dalam    %g
            langkah\n', p,i);
31.         break
32.      else
33.              fa=mil(a);
34.              if(fa*fp)>0;
35.                 a=p;
36.                 fa=mil(p);
37.              else
38.                    b=p;
39.              end
40.      end
41. end
```

Adapun hasil yang telah diterapkan dalam penggunaan *file-m* pada gambar sebagai berikut.

```
1       % Nama Program : interpolasi_linier.m
2       %
3       % Tujuan Program : mencari akar persamaan dengan menggunakan metode biseksi
4       % Pembuatan asli :  Rahmat Hadi Siregar
5       % Tanggal :    09.03.2017
6       %
7       % Edit Ulang :  Rosa M. Sarpanaah
8       % Tanggal :    27.03.2018
9       %
10      % Untuk :  masukkan fungsi yang diketahui sebelum mil.m
11      %          simpan terlebih dahulu membuat persamaan fungsi untuk x(x)
12      %          function y=mil(x)
13      %          y=x^3+x^2-3*x-3;
14      %          masukkan dugaan interval yang diketahui
15
16 -    clear;
17 -    clc;
18 -    a=input('masukkan data nilai dugaan awal = ');
19 -    b=input('masukkan data nilai dugaan kedua = ');
20 -    tol=0.01;
21 -    maxiter=10;
22 -    disp('  n        a         b         p         fa         fb         fp');
23 -    fa=mil(a);
24 -    fb=mil(b);
25 -  for i=1:maxiter
26 -      p=b-((b-a)*(fb/(fb-fa))); % interval
27 -      fp=mil(p); %fp=
28 -      fprintf('%5d %8.5f %8.5f %8.5f %8.5f %8.5f %8.5f\n', i,a,b,p,fa,fb,fp);
29 -      if p<tol
30 -          fprintf('akar persamaan %g dalam %g langkah\n', p,i);
31 -          break
32 -      else
33                fa=mil(a);
34                if(fa*fp)>0;
35                   a=p;
36                   fa=mil(p);
37                else
38                      b=p;
39                end
40 -    end
41 - end
```

Gambar. 3.9. Tampilan Hasil File-M Matlab Metode Interpolasi Linier

Adapun bentuk dalam aplikasi luaran *command window*, tertampil sebagai berikut.



Gambar. 3.10. Tampilan Hasil Command Window Matlab Metode Interpolasi Linier

4. Metode *Newton Rhapson*

Adapun perhitungan simulasi MATLAB pada metode *Newton Rhapson*, menyelesaikan perhitungan salah satu akar dari persamaan pangkat tiga pada interval antara dua titik $x_1 = 1$ dan $x_2 = 2$berikut ini pada: $f(x) = x^3 + x^2 - 3x - 3$ dengan tingkat ketelitian 0,000001 dengan program MATLAB sebagai berikut.

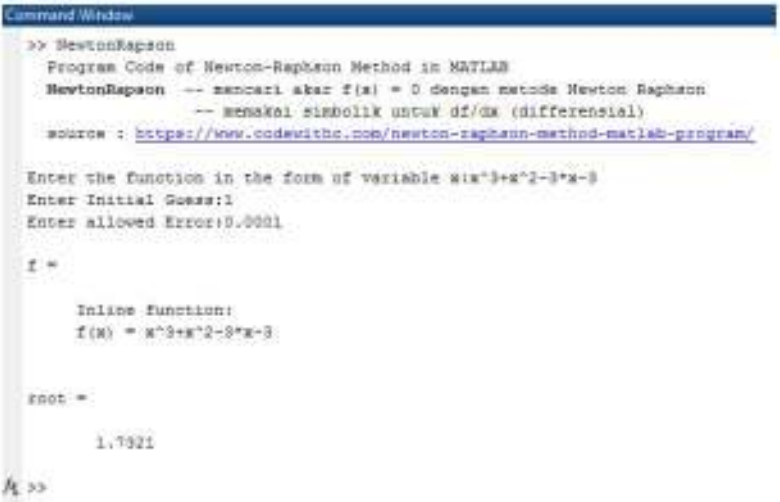
Adapun bentuk tampilan perintah dalam penerapan File-M menggunakan penulisan skrip dari laman berikut https://www.codewithc.com/newton-raphson-method-matlab-program/dan disesuaikan kembali dengan versi simulasi buku ini sebagai berikut.

```
1.  % Program Code of Newton-Raphson Method in MATLAB
```

```
2.  % NewtonRapson-- mencari akar f(x) = 0 dengan metode
    Newton Raphson
3.  %              --    memakai    simbolik    untuk    df/dx
    (differensial)
4.  % source : https://www.codewithc.com/newton-raphson-
    method-matlab-program/
5.  clear; help NewtonRapson;
6.
7.  a=input('Enter the function in the form of variable
    X:','s');
8.  x(1)=input('Enter Initial Guess:');
9.  error=input('Enter allowed Error:');
10. f=inline(a)
11. dif=diff(sym(a));
12. d=inline(dif);
13. for i=1:100
14. x(i+1)=x(i)-((f(x(i))/d(x(i))));
15. err(i)=abs((x(i+1)-x(i))/x(i));
16. if err(i)<error
17. break
18. end
19. end
20. root=x(i)
```

```
1   % Program Code of Newton-Raphson Method in MATLAB
2   % NewtonRapson  -- mencari akar f(x) = 0 dengan metode Newton Raphson
3   %               -- memakai simbolik untuk df/dx (differensial)
4   % source : https://www.codewithc.com/newton-raphson-method-matlab-program/
5 - clear; help NewtonRapson;
6
7 - a=input('Enter the function in the form of variable X:','s');
8 - x(1)=input('Enter Initial Guess:');
9 - error=input('Enter allowed Error:');
10- f=inline(a)
11- dif=diff(sym(a));
12- d=inline(dif);
13- for i=1:100
14- x(i+1)=x(i)-((f(x(i))/d(x(i))));
15- err(i)=abs((x(i+1)-x(i))/x(i));
16- if err(i)<error
17- break
18- end
19- end
20- root=x(i)
```

Gambar. 3.11. Tampilan Hasil File-M Matlab Metode Newton Rhapson



Gambar. 3.12. Tampilan Hasil Command Window Matlab Metode Newton Rhapson

SIMULASI 4: SISTEM PERSAMAAN LINIER

4.1. Pendahuluan

Sistem Persamaan Linier merupakan penyelesaian suatu sistem dengan bilangan n tak diketahui. Bentuk sistem persamaan linier dengan persamaan m dan n variabel bebas. Adapun bentuk persamaan-persamaan yang secara bersama-sama menyajikan banyak variabel bebas. Permasalahan sistem persamaan linier merupakan permasalahan yang banyak muncul ketika berhubungan dengan permasalahan *multi variable* dimana setiap persamaan merupakan bentuk persamaan linier atau dengan kata lain setiap variable berpangkat lebih besar. Kajian permasalahan banyak dijumpai dalam ilmu pengetahuan dan teknologi, seperti analisis struktur, analisis jaringan dan sebagainya.

Bentuk umum penyajian sistem persamaan linier, sebagai berikut.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots + a_{mn}x_n = b_m$$

Keterangan: a_{ij} untuk i = 1 s/d m dan j = 1 s/d n adalah koefisien atau sistem persamaan linier x untuk i = 1 s/d n adalah variabel bebas pada sistem persamaan linier.

Sistem persamaan linier di atas dapat dinyatakan dalam bentuk matriks sebagai berikut.

$$\begin{bmatrix} a_{11} & a_{12}\cdots & a_{1n} \\ a_{21} & a_{22}\cdots & a_{2n} \\ \cdots & \cdots\ \cdots & \cdots \\ a_{m1} & a_{m2}\ \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix}$$

Atau dapat dituliskan:

Ax = B

Di mana:

$$A \begin{bmatrix} a_{11} & a_{12}\cdots & a_{1n} \\ a_{21} & a_{22}\cdots & a_{2n} \\ \cdots & \cdots\ \cdots & \cdots \\ a_{m1} & a_{m2}\ \cdots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} dan\ B = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix}$$

Matriks A dinamakan dengan matriks koefisien dari sistem persamaan linier atau ada yang menamakan dengan matriks jacobian. Vektor x disebut sebagai variabel (atau vektor keadaan) dan vekor B disebut sebagai vektor konstanta.

Augmented Matrix (Matriks Perluasan) dari sistem persamaan linier adalah matrik yang merupakan perluasan matrik A dengan menambahkan vector B pada kolom terakhirnya dan dituliskan sebagai berikut.

$$\begin{bmatrix} a_{11} & a_{12}\cdots & a_{1n} & b_1 \\ a_{21} & a_{22}\cdots & a_{2n} & b_2 \\ \cdots & \cdots\ \cdots & \cdots & \cdots \\ a_{m1} & a_{m2}\ \cdots & a_{mn} & b_n \end{bmatrix}$$

Sebagian permasalahan sering ditemukan penggolongan dua kategori, yaitu suatu sistem persamaan linier dengan *n* kecil tetapi sedikit elemen nol, dan suatu sistem matriks dengan order tinggi (*n* besar) tetapi dengan banyak mengandung elemen nol.

Dalam hal ini akan dibahas mengenai beberapa metode penyelesaian secara langsung dan metode iterasi. Metode penyelesaian langsung adalah dengan mudah untuk kedua kategori sistem persamaan, sedangkan metode iterasi lebih baik dipergunakan untuk matriks dengan order tinggi yang banyak memiliki elemen nol.

Suatu persamaan linier memiliki penyelesaian tunggal apabila memenuhi beberapa persyaratan sebagai berikut.

1. Ukuran sistem persamaan linier memiliki bentuk persegi atau bujur sangkar, dimana jumlah persamaan sama dengan jumlah variabel bebas.
2. Sistem persamaan linier non-homogen, dimana minimal ada satu nilai vektor konstanta B tidak nol atau ada$b_n \neq 0$.
3. Determinan dari matrik koefisien sistem persamaan linier tidak sama dengan nol.

Untuk menyelesaikan permasalahan-permasalahan sistem persamaan linier dapat dilakukan dengan menggunakan metode-metode analitik seperti pemakaian metode grafis, aturan crammer atau invers matriks. Metode-metode tersebut dapat dilakukan dengan mudah apabila jumlah variabel dan jumlah persamaannya di bawah 4, tetapi apabila ukurannya besar maka metode-metode di atas menjadi sulit dilakukan, sehingga pemakaian metode numerik menjadi suatu alternatif yang dipergunakan untuk

menyelesaikan permasalahan-permasalahan dalam sistem persamaan linier diantaranya sebagai berikut.

1. Metode Eliminasi Gauss.
2. Metode Iterasi Jacobi.
3. Metode Iterasi Gauss Seidel.

4.2. Metode Eliminasi Gauss

Metode eliminasi gauss adalah suatu cara mengoperasikan nilai-nilai dalam matriks sehingga nilai matriks menjadi lebih sederhana (ditemukan oleh Carl Friedrich Gauss). Langkah dalam penyelesaian metode ini adalah pengembangan dari metode eliminasi, yaitu menghilangkan atau mengurangi jumlah variable sehingga dapat diperoleh nilai dari suatu variabel bebas. Cara mengeliminasi ini sudah banyak dipergunakan, adapun pola pengerjaan dengan menggunakan metode eliminasi gauss ini, terlebih dahulu bentuk matriks diubah menjadi *augmented matix* yang telah dibahas sebelumnya.

Perhatikan dimana proses sistem persamaan linier ke tahapan penyelesaian dengan mengubah *augmented matix* sebagai berikut.

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

Metode eliminasi gauss, adalah suatu metode dengan bentuk matriks di atas, pada bagian kiri diubah menjadi matriks segitiga atas atau matriks segitiga

bawah dengan menggunakan OBE (Operasi Baris Elementer).

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1n} & d_1 \\ 0 & c_{22} & c_{23} & \cdots & c_{2n} & d_2 \\ 0 & 0 & c_{33} & \cdots & c_{3n} & d_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & c_{nn} & d_n \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{bmatrix}$$

Metode ini dipergunakan dalam analisis numerik untuk meminimalkan mengisi selama eliminasi dengan beberapa tahap. Adapun keuntungan dalam metode ini diantaranya: kemudahan menentukan apakah sistem konsisten, menghilangkan kebutuhan untuk menulis ulang variabel pada setiap langkah dan lebih mudah dalam proses pemecahan. Dampak yang kurang dalam penggunaan dalam metode ini adalah memiliki masalah akurasi ketepatan dalam pembulatan desimal pada akhir penyelesaian.

Algoritma Metode Eliminasi Gauss adalah sebagai berikut.

1. Masukkan matriks A dan vektor B beserta ukurannya n.
2. Buatlah augmented matrix [A|B], namakan dengan A.
3. Untuk setiap baris ke-I dimana i= 1 sampai n, perhatikan apakah nilai $a_{i,j}$ sama dengan nilai nol. Apabila iya, maka tukarkan baris ke-i dan baris $i+k \leq n$, dimana $a_{i+k,i}$ tidak sama dengan nol, bila tidak ada berarti perhitungan tidak bisa

dilanjutkan dan proses dihentikan dengan tanpa penyelesaian.

Apabila tidak, lanjutkan ke langkah (4).

4. Untuk baris ke-j, dimana j=i+1 sampai n, lakukan operasi baris elementer.

   Hitung $c = \frac{a_{i,j}}{a_{i,j}}$

   Untuk kolom k dimana k=1 sampai dengan n+1,

   Hitung $a_{j,k} = a_{j,k} - c.\ a_{j,k}$

5. Hitung akar, untuk i=n sampai 1 (bergerak dari baris ke n sampai baris pertama)

   $$x_i = \frac{1}{a_{i,j}}\left(b_i - a_{i,i+1}x_{i+1} - a_{i,i+2}x_{i+2} - \ldots - a_{i,n}x_n\right)$$

   Dimana nilai I + k ≤ n

Catatan:

Metode eliminasi gauss ini sebenarnya merupakan metode eliminasi yang sering dipergunakan dalam perhitungan manual, hanya saja tekniknya menggunakan model penulisan persamaan bukan menggunakan augmented matriks.

Untuk lebih memahami proses penyelesaian metode ini perhatikan contoh sebagai berikut. Tentukan penyelesaian sistem persamaan linier di bawah ini dengan menggunakan metode elmininasi gauss.

Diketahui:

$3x + y - z = 5$

$4x + 7y - 3z = 20$

$2x - 2y + 5z = 10$

Tahapan penyelesaian pertama, mengubah persamaan dalam bentuk persamaan variabel.

$3x + y - z = 5$ $\qquad a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$ ...(1.a)

$4x + 7y - 3z = 20$ $\qquad a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$ ...(1.b)

$2x - 2y + 5z = 10$ $\qquad a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$ ...(1.c)

Tahapan berikutnya dilakukan pengubahan dalam bentuk berikut.

$$x_1 + \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 = \frac{b_1}{a_{11}}$$

Tahapan ini dengan membagi dengan elemen pertama, kemudian diperoleh hasil sebagai berikut.

$x + 0{,}3333y - 0{,}3333z = 1{,}6666$ ...(2)

Dilanjutkan dengan langkah yang sama dengan persamaan pertama, dengan membagikan pada persamaan kedua.

$$a_{21}x_1 + a_{21}\frac{a_{12}}{a_{11}}x_2 + a_{21}\frac{a_{13}}{a_{11}}x_3 = a_{21}\frac{b_1}{a_{11}}$$

Kali elemen pertama dg pers. 2

$4x + 1{,}3333y - 1{,}3333z = 6{,}6666$ ...(3)

Perubahan dari persamaan (2) dan persamaan (3) dilakukan penggabungan sehingga diperoleh sebagai berikut.

$$\left(a_{22} - a_{21}\frac{a_{12}}{a_{11}}\right)x_2 + \left(a_{22} - a_{21}\frac{a_{12}}{a_{11}}\right)x_3 = \left(b_2 - a_{21}\frac{b_1}{a_{11}}\right) \text{ atau } a'_{22}x_2 + a'_{23}x_3 = b'_2$$

Sehingga didapatkan persamaan Pers. (1b) – Pers. (3) sebagai berikut.

$5{,}6667y - 1{,}6666z = 13{,}3334$ ...(4)

Pers. (2) dikalikan dengan elemen pertama Pers. (1.c)

$a'_{32}x_2 + a'_{33}x_3 + a'_{33}x_3 = b'_3$

$2x + 0{,}6666y - 0{,}6666z = 3{,}3333$ ...(5)

Sehingga pada Pers. (1.c) – Pers. (5), didapatkan hasil perhitungan

$-2{,}6666y + 5{,}6666z = 6{,}6667$ ...(6)

Adapun hasil yang didapatkan pada tahapan ini.

$3x + y - z = 5$ diperoleh ...(1.a) menjadi ...(7.a)

$5{,}6667y - 1{,}6666z = 13{,}3334$ ...(4) ...(7.b)

$-2{,}6666y + 5{,}6666z = 6{,}6667$ ...(6) ...(7.c)

Menormalkan variabel x_2 dengan membagi 5,6667 ke Pers (7.b)

$y + 0{,}2941z = 2{,}3529$ ...(8)

Pers. (8) dikalikan elemen pertama dari Pers. (7.c) yaitu -2,6666

$-2{,}6666y + 0{,}7842z = -6{,}2742$ ...(9)

Pers. (8) dikalikan elemen pertama dari Pers. (7.c) yaitu -2,6666

$4{,}8824z = 12{,}9409$

Sehingga pada tahapan ini, sudah mendekati nilai pada z. maka hasil perhitungan dengan metode eliminasi gauss diperoleh sebagai berikut.

$3x + y - z = 5$ ...(10.*a*)

$5{,}6667y - 1{,}6666z = 13{,}3334$ ...(10.*b*)

$$4{,}8824z = 12{,}9409 \qquad ...(10.c)$$

Maka persamaan di atas dapat diperoleh penghitungan penyelesaian dengan metode eliminasi gauss sebagai berikut.

$$z = \frac{12{,}9409}{4{,}8824} = 2{,}6505$$

$$y = \frac{13{,}3334 + 1{,}6666z}{5{,}6667} = \frac{13{,}3334 + 1{,}6666(2{,}6505)}{5{,}6667} = 3{,}1325$$

$$x = \frac{5 + y - z}{3} = \frac{5 + (3{,}1325) - (2{,}6505)}{3} = 1{,}506$$

4.3. Metode Jacobi

Metode Jacobi adalah suatu jenis metode langsung yang mengacu iterasi. Dalam hal tertentu metode ini lebih baik dibandingkan dengan metode langsung seperti metode eliminasi gauss. Terlihat untuk matrik yang tersebar yaitu matriks yang banyak elemen nol. Dalam metode iterasi ini yang dikenal untuk menyelesaikan sistem persamaan linier ataupun sistem persamaan tak linier, yaitu metode Jacobi dan metode gauss-seidel.

Dalam penjelasan metode jabobi, diantaranya memiliki tahap penyelesaian dengan prosedur sebagai berikut.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \qquad ...(1.a)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \qquad ...(1.b)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \qquad ...(1.c)$$

Dipandang sistem tiga persamaan di atas dengan 3 bilangan tak diketahui, persamaan di atas dapat dipergunakan untuk menghitung x_1 sebagai fungsi dari x_2 dan x_3. Demikian juga persamaan kedua dan ketiga untuk menghitung nilai x_3. Sehingga persamaan kedua dan ketiga untuk menghitungx_2 dan x_3, sehingga perhatikan contoh permasalahan yang diperoleh sebagai berikut.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \qquad ...(1.a)$$

Membagi dengan elemen pertama pada persamaan 1.a

$$x_1 + \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 = \frac{b_1}{a_{11}}$$

Membagi dengan elemen kedua pada persamaan 1.b

$$x_2 = \frac{\left(b_2 - a_{21}x_1 - a_{23}x_3\right)}{a_{22}}$$

Membagi dengan elemen tiga pada persamaan 1.c

$$x_3 = \frac{\left(b_3 - a_{31}x_1 - a_{32}x_3\right)}{a_{33}}$$

Hitungan yang dimulai dengan nilai perkiraan awal sebarang untuk variabel yang dicari (terlihat pada semua variabel diambil sama dengan nol). Nilai perkiraan awal tersebut akan disubstitusikan ke dalam ruas kanan dari sistem persamaan. Perhatikan pola bentuk Iterasi hitungan berakhir setelah:

$$x_1^{\,n} = \frac{\left(b_1 - a_{12}x_2^{\,n-1} - a_{13}x_3^{\,n-1}\right)}{a_{11}}$$
$$x_2^{\,n} = \frac{\left(b_2 - a_{21}x_1^{\,n-1} - a_{23}x_3^{\,n-1}\right)}{a_{22}}$$
$$x_3^{\,n} = \frac{\left(b_3 - a_{31}x_1^{\,n-1} - a_{32}x_2^{\,n-1}\right)}{a_{33}}$$

$$x_1^{\,n-1} \approx x_1^{\,n}, x_2^{\,n-1} \approx x_2^{\,n}, x_3^{\,n-1} \approx x_3^{\,n}$$

Iterasi hitungan terakhir berikut:

$$\left|\varepsilon_{RA}\right| = \frac{\left|a - \hat{a}\right|}{\left|\hat{a}\right|} \times 100\% \le \frac{10^{-k}}{2}$$

Atau telah dipenuhi kriteria berikut:

$$\left|\varepsilon_{RA}\right| = \frac{\left|x_i^{\,n} - x_i^{\,n-1}\right|}{\left|x_i^{\,n}\right|} \times 100\% \le \varepsilon_s$$

Dengan ε_{RA} adalah batasan ketelitian yang dikehendaki.

Untuk lebih memahami proses penyelesaian metode ini perhatikan contoh sebagai berikut. Tentukan penyelesaian sistem persamaan linier di bawah ini dengan menggunakan metode jacobi.

Diketahui:

$3x + y - z = 5$

$4x + 7y - 3z = 20$

$2x - 2y + 5z = 10$

Ubah terlebih dahulu ke dalam bentuk persamaan berikut.

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \quad ...(1.a)$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \quad ...(1.b)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \quad ...(1.c)$$

Kemudian dilanjutkan kedalam bentuk berikut

$$x_1 = \frac{(b_1 - a_{12}x_2 - a_{13}x_3)}{a_{11}} \qquad x = \frac{(5 - y + z)}{3}$$

$$x_2 = \frac{(b_2 - a_{21}x_1 - a_{23}x_3)}{a_{22}} \qquad y = \frac{(20 - 4x + 3z)}{7}$$

$$x_3 = \frac{(b_3 - a_{31}x_1 - a_{32}x_3)}{a_{33}} \qquad z = \frac{(10 - 2x + 2y)}{5}$$

Terlihat hasil pada tahapan pertama, kemudian langkah pertama dicoba nilai x = y = z = 0 dan dihitung x', y' dan z'.

$$x' = \frac{(5 - 0 + 0)}{3} = 1.66667$$

$$y' = \frac{(20 - 4(0) + 3(0))}{7} = 2.85714$$

$$z' = \frac{(10 - 2(0) + 2(0))}{5} = 2$$

Perhitungan pada tahapan berikutnya.

$$x'' = \frac{(5 - 2.85714 + 2)}{3} = 1.38095$$

$$y'' = \frac{(20 - 4(1.66667) + 3(1))}{7} = 2.76190$$

$$z'' = \frac{(10 - 2(1.6667) + 2(2.85714))}{5} = 2.47619$$

Perhatikan hitungan pada tahap selanjutnya.

$$|\varepsilon_x| = \frac{|1.38095 - 1.66667|}{|1.38095|} \times 100\% = 20.69\%$$

$$|\varepsilon_y| = \frac{|2.76190 - 2.85714|}{|2.76190|} \times 100\% = 3.45\%$$

$$|\varepsilon_z| = \frac{|2.47619 - 2|}{|2.47619|} \times 100\% = 19.23\%$$

Sehingga perolehan pada perhitungan tahapan pertama:

$$x = \frac{(5 - y + z)}{3}$$

$$y = \frac{(20 - 4x + 3z)}{7}$$

$$z = \frac{(10 - 2x + 2y)}{5}$$

Adapun hasil dari perhitungan tabel berikut.

$$|C| = \frac{\left|x_i^s - x_i^{s-1}\right|}{\left|x_i^s\right|} \times 100\% \leq \varepsilon_s$$

Diperoleh hasil rangkuman perhitungan metode iterasi Jacobi dari contoh yang telah diberikan di atas.

i	x	y	z	$\lvert\varepsilon_{RA}\rvert$	i	x	y	z
0	0	0	0		0			
1	1.666667	2.857143	3		1	20.69%	3.45%	19.23%
2	1.380952	2.761905	2.47619		2	12.12%	11.74%	2.99%
3	1.571429	3.129252	2.552381		3	6.58%	2.50%	3.70%
4	1.474376	3.053061	2.623129		4	3.22%	2.73%	0.32%
5	1.523356	3.13884	2.631474		5	1.72%	0.78%	0.56%
6	1.497545	3.114428	2.646194		6	0.86%	0.67%	0.02%
7	1.510588	3.125486	2.646751		7	0.45%	0.23%	0.12%
8	1.503756	3.128272	2.649959		8	0.23%	0.17%	0.01%
9	1.507229	3.127551	2.649807		9	0.12%	0.07%	0.01%
10	1.505419	3.127501	2.650529		10	0.06%	0.04%	0.00%
11	1.506343	3.127844	2.650433		11	0.03%	0.02%	0.01%
12	1.505863	3.127275	2.650601		12	0.02%	0.01%	0.00%
13	1.506108	3.127622	2.650565		13	0.01%	0.00%	0.00%
14	1.505981	3.127466	2.650605		14	0.00%	0.00%	0.00%

Gambar. 4.1. Tampilan Hasil Tabel Rangkuman Perhitungan Metode Iterasi Jacobi

Dari hasil perhitungan diperoleh pada iterasi ke-14 perhitungan dihentikan, yang mana pada nilai x, y dan z untuk perhitungan E_{RA} diperoleh nol.

4.4. Metode Gauss-Seidel

Metode gauss-seidel merupakan bentuk memanfaatkan nilai-nilai berikut untuk menghitung variabel berikutnya. Sistem sama seperti penggunaan Metode Jacobi, namun penggunaan metode Gauss-Seidel lebih cepat. Di dalam metode Jacobi, nilai x_1 yang dihitung dari persamaan pertama tidak digunakan untuk menghitung nilai x_2 sehingga nilai-nilai tersebut tidak dipergunakan. Sebenarnya apabila nilai-nilai baru tersebut lebih baik dari nilai-nilai baru tersebut lebih baik dari nilai-nilai yang lama, metode gauss seidel

memanfaatkan nilai-nilai yang baru untuk perhitungan variabel berikutnya.

Metode iterasi gauss-seidel merupakan metode yang memiliki prosedur iterasi hingga diperoleh nilai-nilai yang berubah-ubah. Metode iterasi gauss-seidel dikembangkan dari gagasan metode iterasi pada solusi persamaan linier.

Kelemahan dari penggunaan metode ini adalah masalah pivot (nilai tengah) yang harus benar-benar diperhatikan, karena penyusunan yang salah akan menyebabkan iterasi menjadi divergen dan tidak diperoleh hasil yang benar. Namun, dalam penggunaan metode ini memiliki keunggulan dalam menyelesaikan sistem persamaan linier yang berukuran kecil karena metode ini lebih efisien. Dengan metode ietrasi gauss-seidel, pembulatan dapat diperkecil karena dapat meneruskan iterasi sampai solusinya seteliti mungkin sesuai dengan batas toleransi pembulatan yang diperbolehkan.

Algoritma Metode Iterasi Gauss-Seidel adalah sebagai berikut.

1. Melalui perhitungan nilai-nilai x_i(i=1 s/d *n*) memanfaatkan persamaan-persamaan di atas secara terus menerus hingga nilai untuk setiap x_i(i=1 s/d *n*) sudah sama dengan nilai x_i pada iterasi sebelumnya maka diperoleh penyelesaian dari sistem persamaan linier tersebut.
2. Dalam proses iterasi dihentikan apabila terdapat selisih nilai x_i(i=1 s/d *n*) dengan nilai x_i pada

iterasi sebelumnya kurang dari nilai toleransi *error* yang ditentukan.

3. Dilakukan koreksi kekonvergenan:

$$|\varepsilon_{a,i}| = \frac{\left|x_i^n - x_i^{n-1}\right|}{\left|x_i^n\right|} \times 100\% \leq \varepsilon_s$$

Terlebih dahulu tentukan nilai awal dari setiap x_i(i=1 s/d *n*) kemudian sistem persamaan linier di atas menjadi:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Tahapan berikutnya dilakukan perubahan nilai variabel untuk mencari nilai awal dari setiap x_i(i=1 s/d *n*) kemudian nilai sistem persamaan linier menjadi sebagai berikut.

$$x_1^{\,1} = \frac{\left(b_1 - a_{12}x_2^{\,0} - a_{13}x_3^{\,0}\right)}{a_{11}}$$

$$x_2^{\,1} = \frac{\left(b_2 - a_{21}x_1^{\,1} - a_{23}x_3^{\,0}\right)}{a_{22}}$$

$$x_3^{\,1} = \frac{\left(b_3 - a_{31}x_1^{\,1} - a_{32}x_2^{\,1}\right)}{a_{33}}$$

Catatan yang perlu diperhatikan dalam proses metode iterasi Gauss-Seidel diantaranya: penyusunan dalam sistem persamaan linier, penentuan koefisien dari masing-masing x_i pada semua persamaan di

diagonal utama (a_n), penentuan letak nilai-nilai terbesar dari koefisien untuk setiap x_i pada diagonal utama dan masalah pivoting, karena jika salah akan menyebabkan iterasi menjadi divergen dan tidak memperoleh hasil yang benar.

Untuk lebih memahami proses penyelesaian metode ini perhatikan contoh sebagai berikut. Tentukan penyelesaian sistem persamaan linier di bawah ini dengan menggunakan metode iterasi gauss-seidel.

Diketahui:

$$3x + y - z = 5$$
$$4x + 7y - 3z = 20$$
$$2x - 2y + 5z = 10$$

Ubah terlebih dahulu ke dalam bentuk persamaan berikut:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \quad ...(1.a)$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \quad ...(1.b)$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \quad ...(1.c)$$

Kemudian dilanjutkan kedalam bentuk berikut.

$$x^1 = x_1^{\,1} = \frac{\left(b_1 - a_{12}x_2^{\,0} - a_{13}x_3^{\,0}\right)}{a_{11}} = \frac{5 - 0 + 0}{3} = 1.66667$$

$$y^1 = x_2^{\,1} = \frac{\left(b_2 - a_{21}x_1^{\,1} - a_{23}x_3^{\,0}\right)}{a_{22}} = \frac{20 - 4(1.66667) + 3(0)}{7} = 1.90476$$

$$z^1 = x_3^{\,1} = \frac{\left(b_3 - a_{31}x_1^{\,1} - a_{32}x_2^{\,1}\right)}{a_{33}} = \frac{10 - 2(1.66667) + 2(1.90476)}{5} = 2.09524$$

Dalam perhitungan selanjutnya, diperoleh hasil berikut.

$$x'' = \frac{5 - 1.90476 + 2.09524}{3} = 1.73016$$

$$y'' = \frac{20 - 4(1.73016) + 3(2.09524)}{7} = 2.76644$$

$$z'' = \frac{10 - 2(1.73016) + 2(2.76644)}{5} = 2.41451$$

Perhatikan nilai konvergen dari setiap error antar nilai-nilai yang baru dengan nilai-nilai sebelumnya, sehingga diperoleh perhitungan sebagai berikut.

$$|\varepsilon_x| = \frac{|1.73016 - 1.66667|}{|1.73016|} \times 100\% = 3.67\%$$

$$|\varepsilon_y| = \frac{|2.76644 - 1.90476|}{|2.76644|} \times 100\% = 31.15\%$$

$$|\varepsilon_z| = \frac{|2.41451 - 2.09524|}{|2.41451|} \times 100\% = 13.22\%$$

Diperoleh hasil rangkuman perhitungan metode iterasi Gauss-Seidel dari contoh yang telah diberikan di atas.

I	x	y	z	I	x	y	z
1	1.666667	1.904762	2.095238	1	3.67%	31.65%	12.72%
2	1.730159	2.70644	2.44502	2	8.07%	7.99%	6.52%
3	1.549359	3.006587	2.582892	3	1.57%	2.38%	1.67%
4	1.525435	3.032749	2.626784	4	0.92%	0.86%	0.62%
5	1.59458	3.019221	2.643305	5	0.73%	0.29%	0.19%
6	1.597991	3.01851	2.648699	6	0.09%	0.09%	0.06%
7	1.59863	3.0188	2.649793	7	0.03%	0.03%	0.02%
8	1.598227	3.018057	2.650338	8	0.0%	0.0%	0.0%
9	1.59859	3.018278	2.650516	9	0.00%	0.00%	0.00%

Gambar. 4.2. Tampilan Hasil Tabel Rangkuman Perhitungan Metode Iterasi Gauss-Seidel

Hasil perhitungan tersebut di atas, merupakan bentuk iterasi (1) dalam tabel rangkuman perhitungan. Dengan memperhatikan hasil dari error akan mendekati nilai nol maka perhitungan akan berhenti pada iterasi ke (9). Hal ini terlihat Metode iterasi Gauss-Seidel lebih efisien dibandingkan metode iterasi Jacobi.

4.5. Program MATLAB

Adapun pembahasan MATLAB mengembangkan metode perhitungan dalam sintaksnya dibahas sebagaimana berikut ini. Bentuk pembahasan dalam perhitungan sistem persamaan linier dengan menggunakan Metode Eliminasi Gauss, Metode Jacobi dan Metode Gauss Seidel memanfaatkan program MATLAB menggunakan *command window* dan *File-M.*

Adapun simulasi dengan menggunakan program MATLAB terbagi menjadi beberapa tahap berikut ini.

1. Metode Eliminasi Gauss

Adapun perhitungan simulasi MATLAB pada metode eliminasi gauss, menyelesaikan sistem persamaan linier berikut.

$3x + \quad y - \quad z = 5$

$4x + 7y - 3z = 20$

$2x - 2y + 5z = 10$

Adapun bentuk tampilan dalam penerapan File-M menggunakan penulisan skrip dari situs milik mathwoks yaitu Gauss Elimination.m yang mana dalam laman berikut.

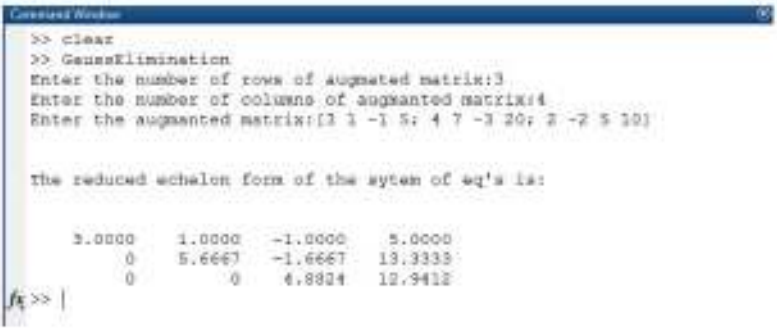
https://www.mathworks.com/matlabcentral/fileexchange/48571-gauss-elimination-method?s_tid=srchtitle

```
1.  clear a b c
2.  row=input('Enter the number of rows of augmated
    matrix:');
3.  column=input('Enter the number of columns of augmanted
    matrix:');
4.  a=input('Enter the augmanted matrix:');
5.  b=a;
6.  counter=1;counter1=1;
7.  while counter<column
8.       counter1=counter+1;
9.       while counter1<=row
10.           if a(counter,counter)== 0
11.               a(counter,:)=a(counter1,:);
12.               a(counter1,:)=a(counter1,:)-
                  a(counter1,counter)/a(counter,counter))
                  *a(counter,:);
13.           counter1=counter1+1;
14.             end
15.           a(counter1,:)=a(counter1,:)-
              a(counter1,counter)/a(counter,counter))*a(cou
              nter,:);
```

```
16.           counter1=counter1+1;
17.       end
18.       counter=counter+1;
19.   end
20.   fprintf('\n\n');
21.   fprintf('The reduced echelon form of the syten of
      eq''s is:\n\n\n');
22.   disp(a);
```

Gambar. 4.3. Tampilan M-File MATLAB Metode Eliminasi Gauss

Penerapan dalam perhitungan pemrograman MATLAB dengan eliminasi gauss sebagai berikut.



Gambar. 4.4. Tampilan Hasil Command Window Matlab Metode Eliminasi Gauss

2. Metode Iterasi Jacobi

Adapun perhitungan simulasi MATLAB pada metode iterasi jacobi, menyelesaikan sistem persamaan linier berikut.

$3x + y - z = 5$

$4x + 7y - 3z = 20$

$2x - 2y + 5z = 10$

Adapun bentuk tampilan dalam penerapan File-M menggunakan penulisan skrip dalam perhitungan metode jacobi dari https://www.mathworks.com/matlabcentral/fileexchange/63167-gauss-seidel-method--jacobi-method yaitu jacobi.m, namun sebelum menerapkan kedalam m-file diperlukan terlebih dulu tentukan matriks dalam m-file programnya.

```
1.  %% Jacobi Method
2.  %% Solution of x in Ax=b using Jacobi Method
3.  % * _*Initailize 'A' 'b' & intial guess 'x'*_
4.  %%
```

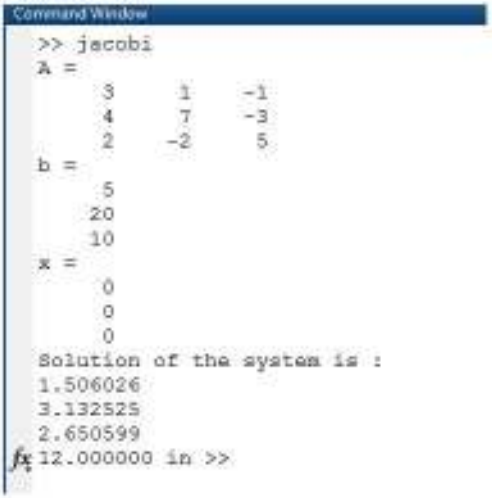
```
5.  A=[3 1 -1; 4 7 -2; 2 -2 5]
6.  b=[5 20 10]'
7.  x=[0 0 0]'
8.
9.  n=size(x,1);
10. normVal=Inf;
11. %%
12. % * _*Tolerence for method*_
13.
14. tol=1e-5; itr=0;
15. %% Algorithm: Jacobi Method
16. %%
17. while normVal>tol
18.      xold=x;
19.
20.      for i=1:n
21.           sigma=0;
22.
23.           for j=1:n
24.
25.                if j~=i
26.                     sigma=sigma+A(i,j)*x(j);
27.                end
28.
29.           end
30.
31.           x(i)=(1/A(i,i))*(b(i)-sigma);
32.      end
33.
34.      itr=itr+1;
35.      normVal=abs(xold-x);
36. end
37. %%
38. fprintf('Solution of the system is : \n%f\n%f\n%f\n%f
    in %d iterations',x,itr);
```

Adapun bentuk tampilan dalam m-file Jacobi.m dari line 1-38 sebagai berikut.

```
%% Jacobi Method
%% Solution of x in Ax=b using Jacobi Method
% * _*Initialize 'A' 'b' & intial guess 'x'*_
%%
A=[3 1 -1; 4 7 -3; 2 -2 5]
b=[5 20 10]'
x=[0 0 0]'

n=size(A,1);
normVal=Inf;
%%
% * _*Tolerance for method*_

tol=1e-5; itr=0;
%% Algorithm: Jacobi Method
%%
while normVal>tol
    xold=x;

    for i=1:n
        sigma=0;

        for j=1:n

            if j~=i
                sigma=sigma+A(i,j)*x(j);
            end

        end

        x(i)=(1/A(i,i))*(b(i)-sigma);
    end

    itr=itr+1;
    normVal=abs(xold-x);
end
%%
fprintf('Solution of the system is : \n%f\n%f\n%f in %d iterations',x,itr);
```

Gambar. 4.4. Tampilan M-File Matlab Function Jacobi (n,A,b) line 1-34

Penerapan dalam perhitungan pemrograman MATLAB dengan metode jacobi sebagai berikut.



Gambar. 4.5. Tampilan Command Window Matlab Iterasi Jacobi SPL (iterasi 1-13)

Dari hasil yang didapatkan tidak mendekati hasil dari perhitungan analitik terhadap perhitungan pemrograman MATLAB yang iterasi Jacobi mendekati hasil akhir, pengguna disarankan langsung menggunakan metode iterasi Gauss-Seidel untuk melakukan perhitungan karena pengaruh pivot.

3. Metode Iterasi Gauss-Seidel

Dengan menerapkan permasalahan berikut ini.

$$3x + y - z = 5$$
$$4x + 7y - 3z = 20$$
$$2x - 2y + 5z = 10$$

Bentuk tampilan dalam penerapan File-M menggunakan penulisan skrip orisinal dan milik dari

mathwoks yaitu gaussseidelspl.m (orisinal) dan Metode_Gauss_Seidel.m (mathworks).

Adapun perhitungan simulasi MATLAB pada metode iterasi gauss-seidel, menyelesaikan sistem persamaan linier berikut. Bentuk m-file yang pertama ini adalah bentuk orisinal dari perhitungan iterasi metode gauss seidel dalam hal ini mengaplikasikan sebanyak 30 iterasi.

```
% gaussseidelspl.m --- menjalankan iterasi eliminasi gauss-seidell mencari
%--- solusi suatu SPL dengan 30 iterasi trial and error
%--- tingkat ketelitian 0.000001

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% program mencoba menjalankan iterasi eliminasi gauss-seidell
% untuk mencari solusi suatu SPL
% reza kusuma setyansah
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%input data
a11=input('masukkan nilai koefisien a11=');
a12=input('masukkan nilai koefisien a12=');
a13=input('masukkan nilai koefisien a13=');
a21=input('masukkan nilai koefisien a21=');
a22=input('masukkan nilai koefisien a22=');
a23=input('masukkan nilai koefisien a23=');
a31=input('masukkan nilai koefisien a31=');
a32=input('masukkan nilai koefisien a32=');
a33=input('masukkan nilai koefisien a33=');
b1=input('masukkan nilai koefisien b1=');
b2=input('masukkan nilai koefisien b2=');
b3=input('masukkan nilai koefisien b3=');

% >>>>>>>>>>>>>>>>>>> Proses data <<<<<<<<<<<<<<<<<<<%

%proses hasil dan tampilan data iterasi gauss seidell
disp('---hasil eliminasi gauss seidel---')
```

```
disp('-----iterasi--------x1--------x2------x3------')
disp('-----------------------------------------------')

%proses data iterasi 1
x11=((b1-a12*0-a13*0)/a11);
x21=((b2-a21*x11-a23*0)/a22);
x31=((b3-a31*x11-a32*x21)/a33);
fprintf('iterasi        ke-1        %10.6f        %10.6f
%10.6f\n',x11,x21,x31);

%proses data iterasi 2
x12=((b1-a12*x21-a13*x31)/a11);
x22=((b2-a21*x12-a23*x31)/a22);
x32=((b3-a31*x12-a32*x22)/a33);
fprintf('iterasi        ke-2        %10.6f        %10.6f
%10.6f\n',x12,x22,x32);

%proses data iterasi 3
x13=((b1-a12*x22-a13*x32)/a11);
x23=((b2-a21*x13-a23*x32)/a22);
x33=((b3-a31*x13-a32*x23)/a33);
fprintf('iterasi        ke-3        %10.6f        %10.6f
%10.6f\n',x13,x23,x33);

%proses data iterasi 4
x14=((b1-a12*x23-a13*x33)/a11);
x24=((b2-a21*x14-a23*x33)/a22);
x34=((b3-a31*x14-a32*x24)/a33);
fprintf('iterasi        ke-4        %10.6f        %10.6f
%10.6f\n',x14,x24,x34);

%proses data iterasi 5
x15=((b1-a12*x24-a13*x34)/a11);
x25=((b2-a21*x15-a23*x34)/a22);
x35=((b3-a31*x15-a32*x25)/a33);
fprintf('iterasi        ke-5        %10.6f        %10.6f
%10.6f\n',x15,x25,x35);

%proses data iterasi 6
x16=((b1-a12*x25-a13*x35)/a11);
x26=((b2-a21*x16-a23*x35)/a22);
x36=((b3-a31*x16-a32*x26)/a33);
```

```
fprintf('iterasi        ke-6        %10.6f        %10.6f
%10.6f\n',x16,x26,x36);

%proses data iterasi 7
x17=((b1-a12*x26-a13*x36)/a11);
x27=((b2-a21*x17-a23*x36)/a22);
x37=((b3-a31*x17-a32*x27)/a33);
fprintf('iterasi        ke-7        %10.6f        %10.6f
%10.6f\n',x17,x27,x37);

%proses data iterasi 8
x18=((b1-a12*x27-a13*x37)/a11);
x28=((b2-a21*x18-a23*x37)/a22);
x38=((b3-a31*x18-a32*x28)/a33);
fprintf('iterasi        ke-8        %10.6f        %10.6f
%10.6f\n',x18,x28,x38);

%proses data iterasi 9
x19=((b1-a12*x28-a13*x38)/a11);
x29=((b2-a21*x19-a23*x38)/a22);
x39=((b3-a31*x19-a32*x29)/a33);
fprintf('iterasi        ke-9        %10.6f        %10.6f
%10.6f\n',x19,x29,x39);

%proses data iterasi 10
x110=((b1-a12*x29-a13*x39)/a11);
x210=((b2-a21*x110-a23*x39)/a22);
x310=((b3-a31*x110-a32*x210)/a33);
fprintf('iterasi        ke-10       %10.6f        %10.6f
%10.6f\n',x110,x210,x310);

%proses data iterasi 11
x111=((b1-a12*x210-a13*x310)/a11);
x211=((b2-a21*x111-a23*x310)/a22);
x311=((b3-a31*x111-a32*x211)/a33);
fprintf('iterasi        ke-11       %10.6f        %10.6f
%10.6f\n',x111,x211,x311);

%prosea data iterasi 12
x112=((b1-a12*x211-a13*x311)/a11);
x212=((b2-a21*x112-a23*x311)/a22);
x312=((b3-a31*x112-a32*x212)/a33);
```

```
fprintf('iterasi       ke-12       %10.6f       %10.6f
%10.6f\n',x112,x212,x312);

%proses data iterasi 13
x113=((b1-a12*x212-a13*x312)/a11);
x213=((b2-a21*x113-a23*x312)/a22);
x313=((b3-a31*x113-a32*x213)/a33);
fprintf('iterasi       ke-13       %10.6f       %10.6f
%10.6f\n',x113,x213,x313);

%proses data iterasi 14
x114=((b1-a12*x213-a13*x313)/a11);
x214=((b2-a21*x114-a23*x313)/a22);
x314=((b3-a31*x114-a32*x214)/a33);
fprintf('iterasi       ke-14       %10.6f       %10.6f
%10.6f\n',x114,x214,x314);

%proses data iterasi 15
x115=((b1-a12*x214-a13*x314)/a11);
x215=((b2-a21*x115-a23*x314)/a22);
x315=((b3-a31*x115-a32*x215)/a33);
fprintf('iterasi       ke-15       %10.6f       %10.6f
%10.6f\n',x115,x215,x315);

%proses data iterasi 16
x116=((b1-a12*x215-a13*x315)/a11);
x216=((b2-a21*x116-a23*x315)/a22);
x316=((b3-a31*x116-a32*x216)/a33);
fprintf('iterasi       ke-16       %10.6f       %10.6f
%10.6f\n',x116,x216,x316);

%proses data iterasi 17
x117=((b1-a12*x216-a13*x316)/a11);
x217=((b2-a21*x117-a23*x316)/a22);
x317=((b3-a31*x117-a32*x217)/a33);
fprintf('iterasi       ke-17       %10.6f       %10.6f
%10.6f\n',x117,x217,x317);

%prosea data iterasi 18
x118=((b1-a12*x217-a13*x317)/a11);
x218=((b2-a21*x118-a23*x317)/a22);
x318=((b3-a31*x118-a32*x218)/a33);
```

```
138. fprintf('iterasi       ke-18       %10.6f       %10.6f
     %10.6f\n',x118,x218,x318);
139.
140. %proses data iterasi 19
141. x119=((b1-a12*x218-a13*x318)/a11);
142. x219=((b2-a21*x119-a23*x318)/a22);
143. x319=((b3-a31*x119-a32*x219)/a33);
144. fprintf('iterasi       ke-19       %10.6f       %10.6f
     %10.6f\n',x119,x219,x319);
145.
146. %proses data iterasi 20
147. x120=((b1-a12*x219-a13*x319)/a11);
148. x220=((b2-a21*x120-a23*x319)/a22);
149. x320=((b3-a31*x120-a32*x220)/a33);
150. fprintf('iterasi       ke-20       %10.6f       %10.6f
     %10.6f\n',x120,x220,x320);
151.
152. %proses data iterasi 21
153. x121=((b1-a12*x220-a13*x320)/a11);
154. x221=((b2-a21*x121-a23*x320)/a22);
155. x321=((b3-a31*x121-a32*x221)/a33);
156. fprintf('iterasi       ke-21       %10.6f       %10.6f
     %10.6f\n',x121,x221,x321);
157.
158. %proses data iterasi 22
159. x122=((b1-a12*x221-a13*x321)/a11);
160. x222=((b2-a21*x122-a23*x321)/a22);
161. x322=((b3-a31*x122-a32*x222)/a33);
162. fprintf('iterasi       ke-22       %10.6f       %10.6f
     %10.6f\n',x122,x222,x322);
163.
164. %proses data iterasi 23
165. x123=((b1-a12*x222-a13*x322)/a11);
166. x223=((b2-a21*x123-a23*x322)/a22);
167. x323=((b3-a31*x123-a32*x223)/a33);
168. fprintf('iterasi       ke-23       %10.6f       %10.6f
     %10.6f\n',x123,x223,x323);
169.
170. %prosea data iterasi 24
171. x124=((b1-a12*x223-a13*x323)/a11);
172. x224=((b2-a21*x124-a23*x323)/a22);
173. x324=((b3-a31*x124-a32*x224)/a33);
```

```
fprintf('iterasi       ke-24       %10.6f       %10.6f
%10.6f\n',x124,x224,x324);

%proses data iterasi 25
x125=((b1-a12*x224-a13*x324)/a11);
x225=((b2-a21*x125-a23*x324)/a22);
x325=((b3-a31*x125-a32*x225)/a33);
fprintf('iterasi       ke-25       %10.6f       %10.6f
%10.6f\n',x125,x225,x325);

%proses data iterasi 26
x126=((b1-a12*x225-a13*x325)/a11);
x226=((b2-a21*x126-a23*x325)/a22);
x326=((b3-a31*x126-a32*x226)/a33);
fprintf('iterasi       ke-26       %10.6f       %10.6f
%10.6f\n',x126,x226,x326);

%proses data iterasi 27
x127=((b1-a12*x226-a13*x326)/a11);
x227=((b2-a21*x127-a23*x326)/a22);
x327=((b3-a31*x127-a32*x227)/a33);
fprintf('iterasi       ke-27       %10.6f       %10.6f
%10.6f\n',x127,x227,x327);

%proses data iterasi 28
x128=((b1-a12*x227-a13*x327)/a11);
x228=((b2-a21*x128-a23*x327)/a22);
x328=((b3-a31*x128-a32*x228)/a33);
fprintf('iterasi       ke-28       %10.6f       %10.6f
%10.6f\n',x128,x228,x328);

%proses data iterasi 29
x129=((b1-a12*x228-a13*x328)/a11);
x229=((b2-a21*x129-a23*x328)/a22);
x329=((b3-a31*x129-a32*x229)/a33);
fprintf('iterasi       ke-29       %10.6f       %10.6f
%10.6f\n',x129,x229,x329);

%prosea data iterasi 10
x130=((b1-a12*x229-a13*x329)/a11);
x230=((b2-a21*x130-a23*x329)/a22);
x330=((b3-a31*x130-a32*x230)/a33);
```

```
210.  fprintf('iterasi        ke-30        %10.6f        %10.6f
      %10.6f\n',x130,x230,x330);
```

Penerapan dalam m-file pemrograman MATLAB dengan metode iterasi gauss seidel yang pertama gaussseidelspl.m sebagai berikut.

```
1     % gaussseidelspl.m --- menjalankan iterasi eliminasi gauss-seidell mencari
2     %                  --- solusi suatu SPL dengan 30 iterasi trial and error
3     %                  --- tingkat ketelitian 0.000001
4
5     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6     % program sederhana menjalankan iterasi eliminasi gauss-seidell
7     % untuk mencari solusi suatu SPL
8     % rena kusuma setyahadi
9     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10
11    %input data
12 -  a11=input('masukkan nilai koefisien a11=');
13 -  a12=input('masukkan nilai koefisien a12=');
14 -  a13=input('masukkan nilai koefisien a13=');
15 -  a21=input('masukkan nilai koefisien a21=');
16 -  a22=input('masukkan nilai koefisien a22=');
17 -  a23=input('masukkan nilai koefisien a23=');
18 -  a31=input('masukkan nilai koefisien a31=');
19 -  a32=input('masukkan nilai koefisien a32=');
20 -  a33=input('masukkan nilai koefisien a33=');
21 -  b1=input('masukkan nilai koefisien b1=');
22 -  b2=input('masukkan nilai koefisien b2=');
23 -  b3=input('masukkan nilai koefisien b3=');
24
187
188   %proses data iterasi 27
189 -  x127=(b1-a12*x226-a13*x326)/a11);
190 -  x227=(b2-a21*x127-a23*x326)/a22);
191 -  x327=(b3-a31*x127-a32*x227)/a33);
192 -  fprintf('iterasi ke-27 %10.6f %10.6f %10.6f\n',x127,x227,x327);
193
194   %proses data iterasi 28
195 -  x128=(b1-a12*x227-a13*x327)/a11);
196 -  x228=(b2-a21*x128-a23*x327)/a22);
197 -  x328=(b3-a31*x128-a32*x228)/a33);
198 -  fprintf('iterasi ke-28 %10.6f %10.6f %10.6f\n',x128,x228,x328);
199
200   %proses data iterasi 29
201 -  x129=(b1-a12*x228-a13*x328)/a11);
202 -  x229=(b2-a21*x128-a23*x328)/a22);
203 -  x329=(b3-a31*x129-a32*x229)/a33);
204 -  fprintf('iterasi ke-29 %10.6f %10.6f %10.6f\n',x129,x229,x329);
205
206   %proses data iterasi 30
207 -  x130=(b1-a12*x229-a13*x329)/a11);
208 -  x230=(b2-a21*x130-a23*x329)/a22);
209 -  x330=(b3-a31*x130-a32*x230)/a33);
210 -  fprintf('iterasi ke-30 %10.6f %10.6f %10.6f\n',x130,x230,x330);
```

Gambar. 4.6. Tampilan M-File Matlab Iterasi Gauss-Seidel (Versi gaussseidelspl)

Penerapan dalam command window perhitungan pemrograman MATLAB dengan gauss seidel sebagai berikut.

Command Window

```
>> gaussseidelsp1
masukkan nilai koefisien a11=3
masukkan nilai koefisien a12=1
masukkan nilai koefisien a13=-1
masukkan nilai koefisien a21=4
masukkan nilai koefisien a22=7
masukkan nilai koefisien a23=-3
masukkan nilai koefisien a31=2
masukkan nilai koefisien a32=-2
masukkan nilai koefisien a33=8
masukkan nilai koefisien b1=5
masukkan nilai koefisien b2=20
masukkan nilai koefisien b3=10
-----------hasil eliminasi gauss seidel--------------
--iterasi---------x1---------x2---------x3-----------
-----------------------------------------------------
iterasi ke-1    1.666667    1.904762    2.095238
iterasi ke-2    1.730159    2.766440    2.414512
iterasi ke-3    1.549358    3.006587    2.582892
iterasi ke-4    1.529435    3.092419    2.626794
iterasi ke-5    1.511458    3.119321    2.643105
iterasi ke-6    1.507961    3.128210    2.648099
iterasi ke-7    1.506680    3.131111    2.649793
iterasi ke-8    1.506227    3.132067    2.650336
iterasi ke-9    1.506090    3.132378    2.650516
iterasi ke-10    1.506046    3.132481    2.650579
iterasi ke-11    1.506031    3.132514    2.650599
iterasi ke-12    1.506026    3.132525    2.650599
```

```
iterasi ke-13    1.506025    3.132528    2.650601
iterasi ke-14    1.506024    3.132530    2.650602
iterasi ke-15    1.506024    3.132530    2.650602
iterasi ke-16    1.506024    3.132530    2.650602
iterasi ke-17    1.506024    3.132530    2.650602
iterasi ke-18    1.506024    3.132530    2.650602
iterasi ke-19    1.506024    3.132530    2.650602
iterasi ke-20    1.506024    3.132530    2.650602
iterasi ke-21    1.506024    3.132530    2.650602
iterasi ke-22    1.506024    3.132530    2.650602
iterasi ke-23    1.506024    3.132530    2.650602
iterasi ke-24    1.506024    3.132530    2.650602
iterasi ke-25    1.506024    3.132530    2.650602
iterasi ke-26    1.506024    3.132530    2.650602
iterasi ke-27    1.506024    3.132530    2.650602
iterasi ke-28    1.506024    3.132530    2.650602
iterasi ke-29    1.506024    3.132530    2.650602
iterasi ke-30    1.506024    3.132530    2.650602
fx >>
```

Gambar. 4.7. Tampilan Command Window Matlab Iterasi Gauss-Seidel (Versi gaussseidelspl)

Terlihat pada hasil gambar 4.10 mengalami hasil yang sama pada iterasi ke-13 dan iterasi ke-14, maka dapat disimpulkan bahwa hasil yang diperoleh sebesarx_1= 1.506024, x_2= 3.132530 dan x_3= 2.650602. dari perolehan tersebut mendekati hasil yang sesuai dengan hasil dari pembahasan gauss seidel sebelumnya.

Adapun perhitungan simulasi MATLAB pada metode iterasi gauss-seidel, menyelesaikan sistem persamaan linier berikut. Bentuk m-file yang kedua ini adalah bentuk data dari mathwork dengan perhitungan iterasi metode gauss seidel sebagai berikut.

```
1.  % Metode_Gauss_Seidel.m --- menjalankan iterasi gauss-
    seidell untuk mencari
2.  %                       --- solusi suatu SPL dengan
    perolehan hasil akhir
```

```
3.  %                          ---    dan    nilai    banyaknya
    iterasi
4.
5.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6.  % program mencoba menjalankan iterasi gauss-seidell
7.  % untuk mencari solusi suatu SPL
8.  % mathworks
9.  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10. %% memasukkan dan membaca matriks %%
11. % definisi matriks A;
12. disp('Masukkan matriks Ax SPL dg format [a11, a12 a13;
    dst]');
13. A=input('');
14. % definisi ruas kanan b
15. disp('Masukkan matriks b SPL dg format [b11; b21;
    b31]');
16. b=input('');
17. N=length(b);
18. x0=zeros(N,1)
19. MAXIT=1000
20. EPS=0.000001
21. K=1
22. while (K<=MAXIT)
23.      x=x0;
24.      I=1;
25.      while (I<=N)
26.           SUMA=0;
27.           J=1;
28.           while (J<=N)
29.                if (J~=I)
30.                SUMA=SUMA+A(I,J)*x0(J);
31.                end
32.                J=J+1;
33.           end
34.           x0(I)=(b(I)-SUMA)/A(I,I);
35.           I=I+1;
36.      end
37.      if (abs(x-x0)<=EPS)
38.         fprintf('Vektor Solusi %d %d'),x,K
39.      break;
40.      end
41.           K=K+1;
42. end
```

```
43.  if (abs(x-x0)>EPS)
44.       fprintf('Tidak ada penyelesaian yang terjadi %d
         %d'),x,K
45.  End
```

Penerapan dalam m-file pemrograman MATLAB dengan metode iterasi gauss seidel yang kedua disimpan dengan nama file Metode_Gauss_Seidel.m, penulis memberikan nama dengan penjedaan tanda dan besar kecil huruf. Akan tetapi pengguna dapat menggantinya dengan sesuai nama yang dikehendaki, namun dengan catatan pemanggilan nama file dalam command window harus sama.

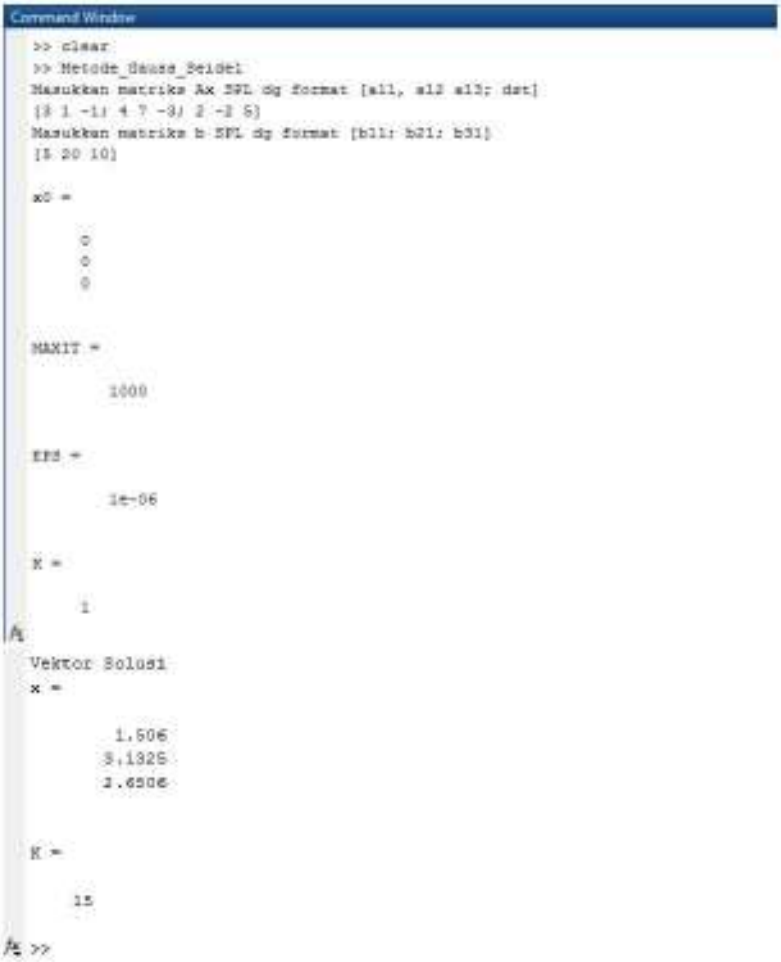
Berikut ini akan disajikan tampilan hasil dari penerapan m-file yang kedua dari Metode_Gauss_Seidel.m

```
1   % Metode_Gauss_Seidel.m ---- menjalankan iterasi gauss-seidell untuk mencari
2   %                        ---- solusi suatu SPL dengan perolehan hasil akhir
3   %                        ---- dan nilai banyaknya iterasi
4
5   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6   % program mencoba menjalankan iterasi gauss-seidel
7   % untuk mencari solusi suatu SPL
8   % mathworks
9   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10  %% memasukkan dan membaca matriks %%
11  % definisi matriks A:
12  disp('Masukkan matriks As SPL dg format [a11, a12 a13; dst]');
13  A=input('');
14  % definisi suku ruas kanan b
15  disp('Masukkan matriks b SPL dg format [b11; b21; b31]');
16  b=input('');
17  B=length(b);
18  x0=zeros(N,1)
19  MAXIT=1000
20  EPS=0.000001
21  K=1
22  while (K<=MAXIT)
23      x=x0;
24      I=1;
25      while (I<=N)
```

```
    while (I<=N)
        SUMA=0;
        J=1;
        while (J<=N)
            if (J~=I)
            SUMA=SUMA+A(I,J)*x0(J);
            end
            J=J+1;
        end
        x0(I)=(b(I)-SUMA)/A(I,I);
        I=I+1;
    end
    if (abs(x-x0)<=EPS)
        fprintf('Vektor Solusi %d %d',x,E
    break;
    end
        N=N+1;
end
if (abs(x-x0)>EPS)
    fprintf('Tidak ada penyelesaian yang terjadi %d %d',x,E
end
```

Gambar. 4.8. Tampilan M-File Matlab Iterasi Gauss-Seidel (Versi Metode_Gauss_Seidel)

Penerapan dalam command window perhitungan pemrograman MATLAB dengan gauss seidel sebagai berikut.



Gambar. 4.9. Tampilan Command Window Matlab Iterasi Gauss-Seidel (Versi Metode_Gauss_Seidel))

Hasil pemanggilan m-file pemrograman MATLAB dengan nama file Metode_Gauss_Seidel.m pada command window memberikan hasil tampilan pada

gambar 4.12 dimana hasil memberikan K = 15, berarti dinyatakan hasil berhenti atau optimal pada iterasi ke-15 dan memberikan hasil sebesar x_1= 1.506, x_2= 3.1325 dan x_3= 2.6506.

SIMULASI 5: ANALISIS REGRESI DAN INTERPOLASI

5.1. Pendahuluan

Perkembangan permasalahan yang mencakup hubungan rangkaian suatu data adalah fungsi yang melibatkan data. Sebagai contoh apabila diketahui data-data penjualan suatu produk, akan muncul pertanyaan adakah fungsi yang menyatakan bahwa penjualan merupakan fungsi dari waktu. Beberapa kenyataan menunjukkan bahwa penjualan dipengaruhi oleh waktu. Ataupun data tersebut dapat berupa pengujian hasil percobaan dari laboratorium atau pengamatan dari lapangan. Hal-hal tersebut dapat diambil seperti contoh berikut ini.

1. Penjualan smartphone yang meningkat apabila pada hari-hari tertentu seperti hari liburan, hari-hari sebelum lebaran, hari-hari sebelum natal dan hari-hari sebelum malam tahun baru.
2. Penjualan peralatan sekolah yang akan meningkat pada setiap tahun ajaran baru.
3. Pengujian kuat desak beton yang memberikan hubungan antara beban dan kuat desak beton.
4. Pengukuran debit sungai yang memberikan hubungan antara kedalam aliran dan debit sungai.
5. Pertumbuhan arus pengiriman barang atau penumpang menjelang liburan hari-hari sebelum lebaran.

Kemungkinan dalam pengujian ataupun pengambilan data, pasti terdapat adanya kesalahan atau ketidakpastian dalam pelaksanaannya. Dari kenyataan-kenyataan di atas dapat dikatakan bahwa fungsi dari waktu, merupakan persoalan bagaimana menyajikan fungsi tersebut. Persoalan tersebut tidaklah mudah dipecahkan, karena sangat idealnya bila diketahui suatu fungsi yang bisa dinyatakan dalam penjualan/pengujian terhadap fungsi waktu.

Pengukuran atau variasi perubahan data dari waktu ke waktu, maka titik-titik data tersebar ke dalam koordinat x-y, sebagai contoh, volume barang atau jumlah penumpang, yang dipengaruhi dengan jumlah yang tidak sama setiap waktunya.

Dalam analisis regresi ditampilkan kurva atau fungsi berdasarkan sebaran titik data. Kurva tersebut akan mewakili titik-titik data yang terbentuk, dilakukan ekstrapolasi data guna mendapatkan nilai y yang berkaitan dengan nilai x yang berada diluar rangkaian data yang ada. Pada persoalan yang berbeda, keuntungan yang dihasilkan pada sebuah perusahaan dagang atau industri merupakan fungsi dari jumlah produk yang diperdagangkan. Semakin besar jumlah produk yang dijual maka semakin besar hasil penjualan atau penerimaan yang diperoleh. Tetapi apabila semakin besar hasil dalam penjualan mengindikasikan semakin besar pula produk yang dibeli atau diproduksi yang berimbas pada biaya yang dibutuhkan.

Untuk dapat menyajikan fungsi, yang dapat dilakukan adalah menggunakan fungsi pendekatan yang paling sesuai untuk menyatakan suatu data berdasarkan model fungsi tertentu seperti fungsi linier, fungsi eksponensial dan fungsi polinomial. Dalam bentuk pendekatan tersebut dikenal sebagai Regresi. Jenis-jenis regresi sangat ditentukan oleh model fungsi yang ditentukan, sehingga jenis regresi yang paling sering digunakan di antaranya:

1. Regresi Linier
   Bentuk regresi ini merupakan suatu cara dari pendekatan dengan fungsi linier. Bagian dari data X dan Y akan dipergunakan sebagai hubungan secara linier.
   $$y = ax + b$$
2. Regresi Polinomial
   Bentuk regresi ini merupakan suatu cara dari pendekatan dengan fungsi polinomial. Bagian dari data X dan Y akan dipergunakan sebagai hubungan secara polinomial.
   $$y = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$$
3. Regresi eksponensial
   Bentuk regresi ini merupakan suatu cara dari pendekatan dengan fungsi linier. Bagian dari data X dan Y akan dipergunakan sebagai hubungan secara eksponensial.
   $$y = e^{-i\omega t}$$

Adapun bentuk pendekatan yang berbeda untuk menyatakan fungsi tetapi untuk mencari nilai-nilai

antara titik-titik yang diketahui sehingga pola fungsinya semakin jelas terlihat atau membentuk suatu kurva. Prosedur pendekatan ini dikenal dengan nama interpolasi. Interpolasi dimanfaatkan untuk menentukan titik-titik yang lain berdasarkan fungsi pendekatan yang ditentukan sebelumnya. Berdasarkan fungsi pendekatan yang dipergunakan, interpolasi memiliki dua macam bentuk yaitu:

1. Interpolasi Linier

   Interpolasi linier, merupakan suatu bentuk interpolasi dengan fungsi pendekatan linier menghubungkan dua buah titik data dengan garis lurus. Kemiringan garis yang menghubungkan dua titik data dan merupakan perkiraan beda hingga turunan pertama, semakin kecil interval antara titik data, maka hasil perkiraan akan semakin baik.

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

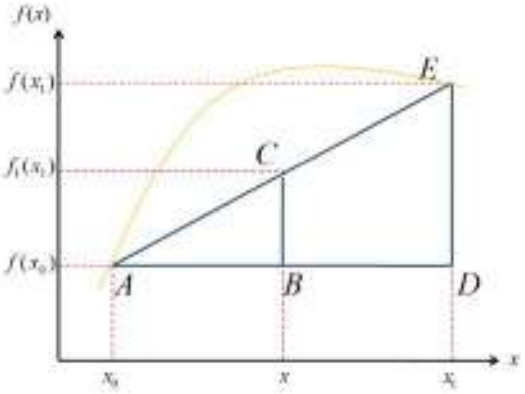
2. Interpolasi Lagrange

   Interpolasi lagrange, merupakan suatu bentuk interpolasi dengan fungsi pendekatan polinomial. Interpolasi Lagrange pada dasarnya dilakukan untuk menghindari perhitungan dari diferensiasi terbagi hingga (Interpolasi Newton)

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

5.2. Interpolasi Linier

Interpolasi linier adalah bentuk interpolasi untuk menentukan titik-titik antara dari titik-titik yang diketahui menggunakan fungsi pendekatan yang berupa fungsi linier. Dengan interpolasi linier, akan diperoleh sejumlah titik antara dua titik antara titik data dapat dilihat dari dua segitiga sebangun ABC dan ADE.



Gambar. 5.1. Hubungan Interpolasi Linier

Jika terdapat hubungan antara dua segitiga sebangun ABC dan ADE dimana didapatkan hubungan sebagai berikut:

$$\frac{BC}{AB} = \frac{DE}{AD}$$

Maka akan diperoleh bentuk persamaan sebagai berikut.

$$\frac{f_1(x)-f(x_0)}{x-x_0}=\frac{f(x_1)-f(x_0)}{x_1-x_0}$$

Apabila nilai dari $f_1(x)$ dicari nilai keberadaan, maka bentuk dari interpolasi linier diperoleh sebagai berikut.

$$f_1(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)$$

Sebagai contoh pertama, perhatikan penyelesaian permasalahan berikut.

Diketahui:

$x_0 = 5 \rightarrow f(x0) = 2{,}015$

$x_1 = 2{,}5 \rightarrow f(x1) = 2{,}571$

$x = 4 \rightarrow f(x) = ?$

Maka penyelesaian di atas, dengan prosedur sebagaimana berikut.

$$f_1(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)$$

$$f_1(x)=2.015+\frac{2.571-2.015}{2.5-5}(4-5)$$

$$f_1(x)=2.015+\frac{0.556}{-2.5}(-1)$$

$$f_1(x)=2.015+0.2224=2.2374\approx 2.237$$

Dengan permasalahan yang dikembangkan pada permasalahan kedua, perhatikan saksama langkah penyelesaian interpolasi linier berikut.

Dari data ln(9.0) = 2.1972 dan ln(9.5) = 2.2513, tentukan nilai ln(9.2) dengan menggunakan interpolasi linier.

Bandingkan dengan nilai eksak ln(9.2) = 2.2192

ln(9.0) = 2.1972

ln(9.5) = 2.2513

ln(9.2) = ?

Penyelesaian dari interpolasi linier, sebagai berikut.

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

$$f_1(x) = 2.1972 + \frac{2.2513 - 2.1972}{9.5 - 9.0}(9.2 - 9.0)$$

$$f_1(x) = 2.1972 + \frac{0.0541}{0.5}(0.2)$$

$$f_1(x) = 2.2188$$

Hasil dari pembandingan nilai eksak ln(9.2) = 2.2192 dengan nilai hampiran perhitungan interpolasi linier sebagai berikut.

$$E_t = \frac{E_a}{a} \times 100\%$$

$$E_t = \frac{2.2192 - 2.2188}{2.2192} \times 100\%$$

$$E_t = 0.018\%$$

Maka diperoleh besaran hasil galat/error perhitungan dari interpolasi linier sebesar 0,018% dengan nilai eksak ln(9.2) = 2.2192.

5.3. Interpolasi Lagrange

Interpolasi lagrange adalah suatu bentuk interpolasi dengan fungsi pendekatan berupa fungsi polynomial lagrange. Interpolasi polinomial Lagrange hampir sama dengan interpolasi polinomial Newton, tetapi tidak menggunakan bentuk pembagian beda hingga. Interpolasi Lagrange pada dasarnya dilakukan untuk menghindari perhitungan dari diferensiasi terbagi hingga (Interpolasi Newton).

Interpolasi polinomial Lagrange dapat diturunkan dari persamaan Newton. hasil bentuk persamaan berikut.

Interpolasi Lagrange Orde Satu

Bentuk polinomial Newton order satu

$$f_1(x) = f(x_0) + (x - x_0).[f(x_1, x_0)] \quad ...(1)$$

Pembagi beda hingga pada persamaan tersebut mempunyai bentuk:

$$f(x_1, x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$f(x_1, x_0) = \frac{f(x_1)}{x_1 - x_0} + \frac{f(x_0)}{x_1 - x_0} \quad ...(2)$$

Substitusi pers. (2) ke dalam pers. (1), diperoleh:

$$f_1(x) = f(x_0) + \frac{x - x_0}{x_1 - x_0}.f(x_1) + \frac{x - x_0}{x_1 - x_0}.f(x_0)$$

Dengan mengelompokkan suku-suku di ruas kanan, maka persamaan di atas menjadi,

$$f_1(x) = \left[\frac{x_0 - x_1}{x_0 - x_1} + \frac{x - x_0}{x_0 - x_1}\right] f(x_0) + \frac{x - x_0}{x_1 - x_0} . f(x_1)$$

Sehingga dari perolehan bentuk pengelompokan didapatkan bentuk Pers. (3) dikenal sebagai interpolasi polinomial Lagrange orde 1.

$$f_1(x) = \left[\frac{x-x_1}{x_0-x_1}\right] f(x_0) + \left[\frac{x-x_0}{x_1-x_0}\right] f(x_1) \quad ...(3)$$

Interpolasi Lagrange Orde Dua

Dengan prosedur yang sama, untuk interpolasi polinomial Lagrange orde 2 akan didapat:

$$f_1(x) = \frac{x - x_1}{x_0 - x_1}\frac{x - x_2}{x_0 - x_2} f(x_0) + \frac{x - x_0}{x_1 - x_0}\frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_0}{x_2 - x_0}\frac{x - x_1}{x_2 - x_1} f(x_2)$$

Interpolasi Lagrange Orde n

Secara umum bentuk interpolasi Lagrange orde n adalah:

$$f_n = \sum_{i=0}^{n} L_i(x) . f(x_i)$$

Dengan:

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

di mana simbol π (product) merupakan perkalian.

Dengan menggunakan persamaan tersebut dapat dihitung interpolasi Lagrange orde yang lebih tinggi.

Misalnya untuk interpolasi lagrange orde 3, persamaannya adalah:

$$f_3(x) = \sum_{i=0}^{3} L_i(x).f(x_i)$$

Maka persamaan tersebut didapatkan dengan bentuk sebagai berikut.

$$L_0(x).f(x_0) + L_1(x).f(x_1) + L_2(x).f(x_2) + L_3(x).f(x_3)$$

Bentuk hasil dari masing-masing $L_i(x)$ sampai dengan orde 3 diperoleh sebagai berikut.

$$L_0(x) = \frac{x-x_1}{x_0-x_1}\frac{x-x_2}{x_0-x_2}\frac{x-x_3}{x_0-x_3}$$

$$L_1(x) = \frac{x-x_0}{x_1-x_0}\frac{x-x_2}{x_1-x_2}\frac{x-x_3}{x_1-x_3}$$

$$L_2(x) = \frac{x-x_0}{x_2-x_0}\frac{x-x_1}{x_2-x_1}\frac{x-x_3}{x_2-x_3}$$

$$L_3(x) = \frac{x-x_0}{x_3-x_0}\frac{x-x_1}{x_3-x_1}\frac{x-x_2}{x_3-x_2}$$

Agar lebih memahami interpolasi lagrange perhatikan contoh pertama berikut.

Dengan menggunakan interpolasi polinomial Lagrange orde 1, hitunglah nilai ln 4, apabila diketahui: ln 2 = 0,69314718 dan ln 6 = 1,791759469.

Penyelesaian dari interpolasi lagrange orde 1, sebagai berikut.

Diketahui:

$x_0 = 2, \rightarrow f(x_0)$ =0,69314718
$x_1 = 6, \rightarrow f(x_1)$ =1,791759469

Dari persamaan yang diketahui, maka disubstitusikan ke dalam interpolasi polinomial Lagrange orde 1 sebagai berikut.

$$f_1(x) = \left[\frac{x - x_1}{x_0 - x_1}\right] f(x_0) + \left[\frac{x - x_0}{x_1 - x_0}\right] f(x_1)$$

diperoleh,

$$f_1(4) = \left[\frac{x-6}{2-6}\right], 0{,}69314718 + \left[\frac{x-2}{6-2}\right] 1{,}791759469$$
$$f_1(4) = 1{,}242453325$$

Dengan memahami interpolasi lagrange perhatikan contoh pertama, maka perlu pemahaman pada contoh yang kedua dengan permasalahan sebagai berikut.

Dengan menggunakan interpolasi polinomial Lagrange orde 2, hitunglah nilai ln 4, apabila diketahui: ln 2 = 0,69314718, ln 3 = 1,098612289 dan ln 6 =1,791759469.

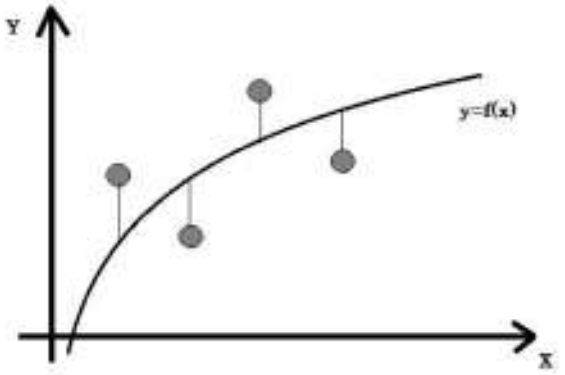
Diketahui:

$x_0 = 2, \rightarrow f(x_0)$ =0,69314718
$x_1 = 3, \rightarrow f(x_1)$ =1,098612289
$x_2 = 6, \rightarrow f(x_2)$ =1,791759469

$$f_1(4) = \left[\frac{x-3}{2-3}\frac{x-6}{2-6}\right], 0{,}69314718 + \left[\frac{x-2}{3-2}\frac{x-6}{3-6}\right] 1{,}098612289 + \left[\frac{x-2}{6-2}\frac{x-3}{6-3}\right] 1{,}791759469$$
$$f_1(4) = 1{,}416869373$$

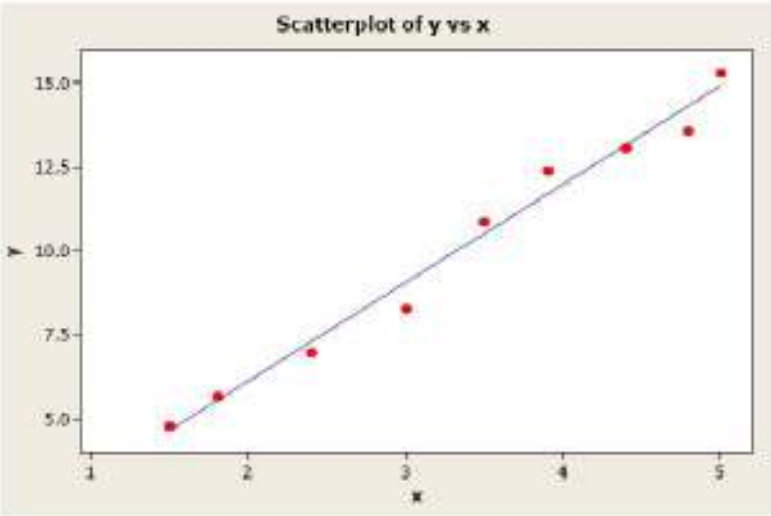
5.4. Regresi Linier

Jarak Regresi linier merupakan teknik untuk memperoleh fungsi secara umum yang dapat merepresentasikan atau menyatakan pola data. Karena bentuk fungsi yang didapatkan adalah fungsi pendekatan, maka analisis regresi bisa dikatakan suatu metode untuk mendapatkan fungsi pendekatan yang jaraknya paling kecil pada semua data. Apabila terdapat data (x_i,y_i) maka regresi akan memperoleh fungsi pendekatan $\hat{y} = f(x)$



Gambar. 5.2. Hubungan Regresi Linier

Jarak antara titik data dengan fungsi pendekatan terlihat pada gambar di atas, dengan mengambil pada jarak x yang sama. Jarak antara fungsi pendekatan dengan data x memiliki nilai yang sama. Melihat dari titik-titik data asumsi *scatterplot of y vs x* maka akan terlihat hubungan dari titik-titik data yang diplot dari gambar berikut di bawah ini.



Gambar. 5.3. scatterplot of y vs x

Diperoleh dari data yang diplot mengumpul pada di sekitar sebuah garis lurus sehingga dapat dikatakan bahwa sebuah garis lurus yang menggambarkan situasi yang cukup masuk akal. Sehingga dapat dinyatakan:

$$\hat{y} = a_0 + a_1 x$$

Pernyataan persamaan yang menggambarkan sebuah garis lurus.

Selanjutnya diminimumkan dengan S (Jumlah kuadrat galat) atau dikenal Metode Least Square yang sangat baik dipergunakan untuk mendapatkan fungsi pendekatan polinomial. Bentuk persamaan dinyatakan sebagai berikut.

$$Min\ S = Min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = Min \sum_{i=1}^{n} [y_i - (a_0 + a_1 x)]^2$$

S adalah fungsi dari dua peubah yang tidak diketahui yaitu a_0 dan a_1. Maka untuk meminimumkan S diambil turunan parsial dari S terhadap a_0 dan a_1 kemudian sama dengan nol. Maka akan diperoleh persamaan sebagai berikut.

$$\frac{\partial S}{\partial a_0} = \sum_{i=1}^{n} 2(y_i - a_1 x_i - a_0)(-1) = 0$$

Melihat persamaan di atas, maka akan menjadi:

$$\sum y_i - a_1 \sum x_i - n a_0 = 0 \ldots (*)$$

Dan pada persamaan berikut,

$$\frac{\partial S}{\partial a_1} = \sum_{i=1}^{n} 2(y_i - a_1 x_i - a_0)(-x_i) = 0$$

Maka akan menjadi,

$$-\sum x_i y_i + a_1 \sum x_i^2 - a_0 \sum x_i = 0 \ldots (**)$$

Dengan pengaturan kembali dari kedua persamaan tersebut diperoleh sistem persamaan untuk a_0 dan a_1 berikut yang dikenal dengan nama persamaan normal:

$$n a_0 + \left(\sum x_i\right) a_1 = \sum y_i$$
$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 = \sum x_i y_i$$

Sehingga bentuk dari penyelesaiannya adalah:

$$a_0 = \frac{1}{n}\sum y_i - \frac{1}{n}\sum x_i a_1 = \bar{y} - a_1 \bar{x}$$

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

Untuk mengetahui derajat kesesuaian dari persamaan yang didapat, dihitung nilai koefisien korelasi yang berbentuk:

$$r = \sqrt{\frac{D_t - D}{D_t}}$$

Dengan r adalah koefisien korelasi ,sedang Dt dan D diberikan oleh bentuk :

$$D_t = \sum_{i-1}^{n} (y_i - \bar{y})^2$$

$$D = \sum_{i-1}^{n} (y_i - a_0 - a_1 x_i)^2$$

Nilai r bervariasi antara 0 dan 1. Untuk perkiraan yang sempurna nilai r = 1. Apabila r = 0 perkiraan suatu fungsi sangat jelek.

Dengan memahami interpolasi lagrange perhatikan contoh pertama, maka perlu pemahaman pada contoh yang kedua dengan permasalahan sebagai berikut.

Tentukan persamaan garis dan koefisien korelasi yang mewakili data berikut.

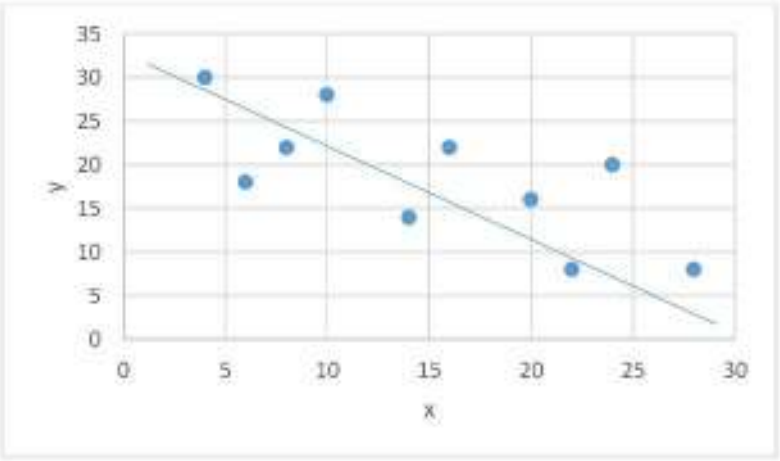
x	4	6	8	10	14	16	20	22	24	28
y	30	18	22	28	14	22	16	8	20	8

Penyelesaian:

Tabel. 5.1. Tabulasi Data Regresi Linier

No.	x	y	$x_i.y_i$	x_i^2
1	4	30	120	16
2	6	18	108	36
3	8	22	176	64
4	10	28	280	100
5	14	14	196	196
6	16	22	352	256
7	20	16	320	400
8	22	8	176	484
9	24	20	480	576
10	28	8	224	784
$\sum$	152	186	2432	2912

Tempatkan pasangan data ke dalam sistem koordinat x y. Kemudian buat garis lurus dengan teknik "tangan bebas" yang mana garis lurus tersebut sedapat mungkin melalui semua data yang ada.



Gambar. 5.4. Scatterplot Contoh Permasalahan

Perhitungan analisis regresi linier dengan menggunakan rumus sebagai berikut:

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

Dengan mencari a_1 terlebih dahulu guna menghitung a_0 dengan rumus:

$$a_0 = \frac{1}{n}\sum y_i - \frac{1}{n}\sum x_i a_1 = \bar{y} - a_1 \bar{x}$$

Berikut hasil perhitungan dengan menggunakan rumus a_1:

$$a_1 = \frac{(10)(2432) - (152)(186)}{(10)(2912) - (152)^2} = -0{,}6569$$

Sebelum menuju ke perhitungan nilai a_0 maka dilakukan terlebih dahulu perhitungan:

$$\bar{y} = \frac{186}{10} = 18{,}6$$

$$\bar{x} = \frac{152}{10} = 15{,}2$$

Kemudian dilanjutkan ke perhitungan a_0:

$$a_0 = \bar{y} - a_1 \bar{x} = 18{,}6 - (-0{,}6569)(15{,}2) = 28{,}5849$$

Maka perolehan dari nilai a_0 dan a_1 maka didapatkan bentuk persamaan sebagai berikut.

$$\hat{y} = a_0 + a_1 x$$

$$\hat{y} = 28{,}5849 - 0{,}6569x$$

Dari hasil perolehan data nilai a_0 dan a_1 akan dipergunakan untuk menghitung hasil dari Dt dan D

guna menghitung hasil dari r. Bentuk dari nilai Dt dan D ditampilkan pada tabel berikut.

Tabel. 5.2. Tabulasi Data Koefisien Korelasi Regresi Linier

No.	x	y	$x_i.y_i$	x_i^2	Dt	D
1	4	30	120	16	129,96	16,34
2	6	18	108	36	0,36	44,14
3	8	22	176	64	11,56	1,77
4	10	28	280	100	88,36	35,81
5	14	14	196	196	21,16	29,03
6	16	22	352	256	11,56	15,41
7	20	16	320	400	6,76	0,31
8	22	8	176	484	112,36	37,61
9	24	20	480	576	1,96	51,56
10	28	8	224	784	112,36	4,80
$\sum$	152	186	2432	2912	496,40	236,79

Maka hasil dari nilai Dt dan D akan dipergunakan untuk memperhitungkan nilai r sebagai berikut.

$$r = \sqrt{\frac{D_t - D}{D_t}}$$

$$r = \sqrt{\frac{496{,}40 - 236{,}79}{496{,}40}} = 0{,}7232$$

Berdasarkan tabel dan perhitungan diperoleh:
a0 = 28,5849 dan a1 = -0,6569

Jadi persamaan garisnya adalah y = 28,5849 – 0,6569 x dengan koefisien korelasi sebesar 0,7232

5.5. Regresi Polinomial

Regresi polynomial merupakan suatu metode regresi dengan bentuk fungsi pendekatan yang berupa fungsi polinomial sebagai berikut:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_2 x^2 + a_1 x + a_0$$

Dimisalkan n pasangan koordinat (x_i, y_i) yang diberikan akan dihampiri oleh suatu fungsi kuadrat yang dinyatakan oleh:

$$\hat{y} = a_0 + a_1 x + a_2 x^2$$

Sehingga jumlah dari kuadrat diberikan oleh:

$$S = \sum (y_i - \hat{y}_i)^2 = \sum (y_i - a_0 - a_1 x - a_2 x^2)^2$$

Turunkan S terhadap masing-masing a_0, a_1, a_2 dan samakan pada masing-masing turunan terhadap koefisien-koefisien ini dengan nilai nol, maka akan diperoleh:

$$\begin{aligned} n a_0 + a_1 \sum x_i + a_2 \sum x_i^2 &= \sum y_i \\ a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 &= \sum x_i y_i \\ a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 &= \sum x_i^2 y_i \end{aligned}$$

Maka diperoleh pendekatan $\hat{y} = a_0 + a_1 x + a_2 x^2$ yang akan merupakan bentuk fungsi kuadrat sehingga bentuk regresi ini akan menjadi regresi kuadratik dengan bentuk matriks sebagai berikut.

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Perhatikan contoh pembahasan regresi polinomial berikut ini, untuk memahami metode regresi tersebut.

Carilah persamaan kurva polinomial order dua yang mewakili data berikut:

x	0	2	3	4	5	6
y	5	5	8	6	5	5

maka perolehan hasil dalam bentuk tabel perhitungan sebagai berikut.

Tabel. 5.3. Tabulasi Data Koefisien Korelasi Regresi Polinomial

	x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i * y_i$	$x_i^2 * y_i$
1.	0	5	0	0	0	0	0
2.	2	5	4	8	16	10	20
3.	3	8	9	27	81	24	72
4.	4	6	16	64	256	24	96
5.	5	5	25	125	625	25	125
6.	6	5	36	216	1296	30	180
	20	34	90	440	2274	113	493

Dari perhitungan dalam data tabulasi koefisien, maka perolehan perhitungan dalam model matriks sebagai berikut.

$$\begin{bmatrix} 6 & 20 & 90 \\ 20 & 90 & 440 \\ 90 & 440 & 2274 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 34 \\ 113 \\ 493 \end{bmatrix}$$

Langkah perhitungan berikutnya menggunakan metode gauss Jordan.

$$\begin{bmatrix} 6 & 20 & 90 \\ 20 & 90 & 440 \\ 90 & 440 & 2274 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 34 \\ 113 \\ 493 \end{bmatrix}$$

... Pada tahap ini pemodelan matriksnya

$$\begin{bmatrix} 6 & 20 & 90 & 34 \\ 20 & 90 & 440 & 113 \\ 90 & 440 & 2274 & 493 \end{bmatrix}$$

... kemudian diubah kedalam bentuk augmented matrix

$$\begin{bmatrix} 1 & 10/3 & 15 & 17/3 \\ 20 & 90 & 440 & 113 \\ 90 & 440 & 2274 & 493 \end{bmatrix}$$

... mengubah elemen pada baris satu kolom satu

$$\begin{bmatrix} 1 & 10/3 & 15 & 17/3 \\ 0 & 70/3 & 140 & -1/3 \\ 90 & 440 & 2274 & 493 \end{bmatrix}$$

... mengubah elemen pada baris dua kolom satu

$$\begin{bmatrix} 1 & 10/3 & 15 & 17/3 \\ 0 & 70/3 & 140 & -1/3 \\ 0 & 140 & 924 & -17 \end{bmatrix}$$

... mengubah elemen pada baris tiga kolom satu

$$\begin{bmatrix} 1 & 10/3 & 15 & 17/3 \\ 0 & 1 & 6 & -1/70 \\ 0 & 140 & 924 & -17 \end{bmatrix}$$

... mengubah elemen pada baris dua kolom dua

$$\begin{bmatrix} 1 & 0 & -5 & 40/7 \\ 0 & 1 & 6 & -1/70 \\ 0 & 140 & 924 & -17 \end{bmatrix}$$

... mengubah elemen pada baris satu kolom dua

$$\begin{bmatrix} 1 & 0 & -5 & 40/7 \\ 0 & 1 & 6 & -1/70 \\ 0 & 0 & 84 & -15 \end{bmatrix}$$

... mengubah elemen pada baris tiga kolom dua

$$\begin{bmatrix} 1 & 0 & -5 & 40/7 \\ 0 & 1 & 6 & -1/70 \\ 0 & 0 & 1 & -5/28 \end{bmatrix}$$

... mengubah elemen pada baris tiga kolom tiga

$$\begin{bmatrix} 1 & 0 & 0 & 135/28 \\ 0 & 1 & 6 & -1/70 \\ 0 & 0 & 1 & -5/28 \end{bmatrix}$$

... mengubah elemen pada baris satu kolom tiga

$$\begin{bmatrix} 1 & 0 & 0 & 135/28 \\ 0 & 1 & 0 & 37/35 \\ 0 & 0 & 1 & -5/28 \end{bmatrix}$$

... mengubah elemen pada baris dua kolom tiga

$$\begin{bmatrix} 1 & 0 & 0 & 4{,}821429 \\ 0 & 1 & 0 & 1{,}057143 \\ 0 & 0 & 1 & -0{,}178571 \end{bmatrix}$$

... mengubah nilai pecahan ke nilai desimal

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 4{,}821429 \\ 1{,}057143 \\ -0{,}178571 \end{bmatrix}$$

... sehingga diperoleh hasil penyelesaiannya.

Adapun bentuk yang diperoleh persamaan kurva polinomial order dua adalah sebagai berikut.

$$\hat{y} = a_0 + a_1 x + a_2 x^2$$

$$\hat{y} = 4{,}821429 + 1{,}057143x - 0{,}718571x^2$$

5.6. Program MATLAB

Pembahasan dalam program MATLAB mengembangkan metode perhitungan dalam sintaksnya dibahas sebagaimana berikut ini. Bentuk pembahasan menggunakan interpolasi linier, interpolasi lagrange, regresi linier dan regresi polinomial. Adapun langkah-langkah dalam program MATLAB menggunakan *command window* dan *File-M*. Adapun simulasi dengan menggunakan program MATLAB terbagi menjadi beberapa tahap berikut ini.

1. Pendekatan Interpolasi Linier

Adapun perhitungan simulasi MATLAB pada metode interpolasi linier, menyelesaikan permasalahan data berikut. Mengacu dari permasalahan yang dikembangkan pada permasalahan kedua, perhatikan saksama langkah penyelesaian interpolasi linier dengan menggunakan program MATLAB berikut.

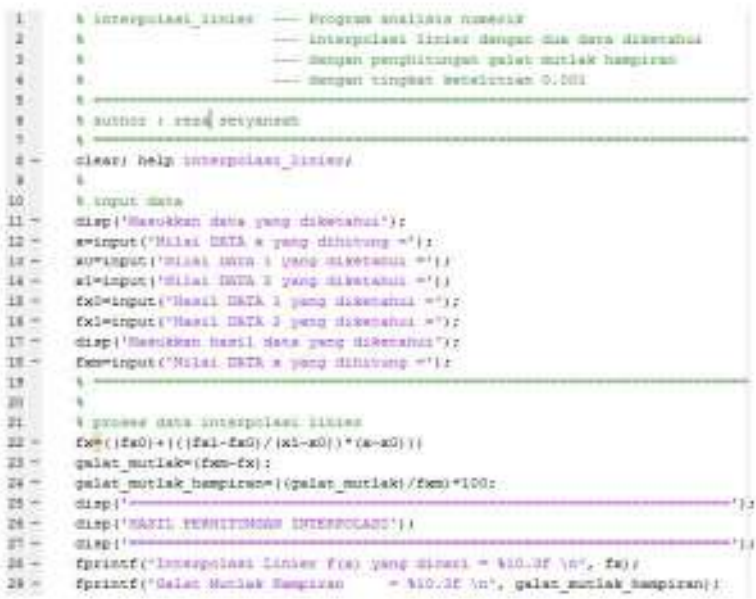
Dari data ln(9.0) = 2.1972 dan ln(9.5) = 2.2513, tentukan nilai ln(9.2) dengan menggunakan interpolasi linier.

Langkah dalam penyelesaian perhitungan simulasi MATLAB pada metode interpolasi linier, menyelesaikan data tersebut sebagai berikut.

```
1.  % interpolasi_linier---Program analisis numerik
2.  %                  ---interpolasi linier dengan dua
    data diketahui
3.  %                  ---dengan penghitungan galat
    mutlak hampiran
4.  %                  ---dengan tingkat ketelitian 0.001
5.  % ------------------------------------------------
6.  % author : reza setyansah
7.  % ------------------------------------------------
8.  clear; help interpolasi_linier;
9.  %
```

```
10. % input data
11. disp('Masukkan data yang diketahui');
12. x=input('Nilai DATA x yang dihitung =');
13. x0=input('Nilai DATA 1 yang diketahui =');
14. x1=input('Nilai DATA 2 yang diketahui =');
15. fx0=input('Hasil DATA 1 yang diketahui =');
16. fx1=input('Hasil DATA 2 yang diketahui =');
17. disp('Masukkan hasil data yang diketahui');
18. fxm=input('Nilai DATA x yang dihitung =');
19. % ------------------------------------------------------
20. %
21. % proses data interpolasi linier
22. fx=((fx0)+(((fx1-fx0)/(x1-x0))*(x-x0)))
23. galat_mutlak=(fxm-fx);
24. galat_mutlak_hampiran=((galat_mutlak)/fxm)*100;
25. disp('-----------------------------------');
26. disp('HASIL PERHITUNGAN INTERPOLASI');
27. disp('-----------------------------------');
28. fprintf('Interpolasi Linier f(x) yang dicari = %10.3f
    \n', fx);
29. fprintf('Galat Mutlak Hampiran = %10.3f \n',
    galat_mutlak_hampiran);
```

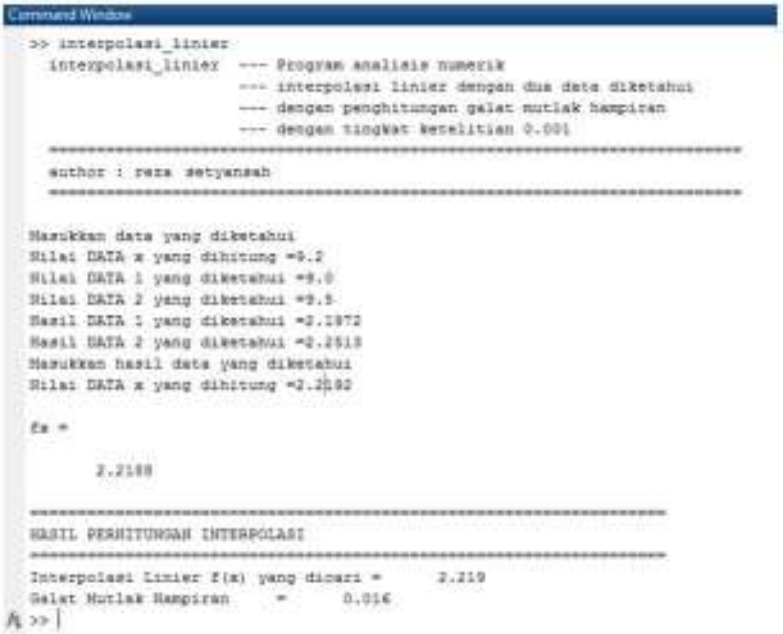
Adapun bentuk tampilan dalam penerapan File-M menggunakan langkah yang sesuai dengan rumusan orisinal yaitu dengan pemberian nama file interpolasi_linier.m, yang dikembangkan menyesuaikan inputan ke dalam m-file.



```
% interpolasi_linier  ---- Program analisis numerik
%                     ---- interpolasi linier dengan dua data diketahui
%                     ---- dengan penghitungan galat mutlak hampiran
%                     ---- dengan tingkat ketelitian 0.001
% ------------------------------------------------------------------------
% author : rezki setyaningsih
% ------------------------------------------------------------------------
clear; help interpolasi_linier;
%
% input data
disp('Masukkan data yang diketahui');
x=input('Nilai DATA x yang dihitung =');
x0=input('Nilai DATA 1 yang diketahui =');
x1=input('Nilai DATA 2 yang diketahui =');
fx0=input('Hasil DATA 1 yang diketahui =');
fx1=input('Hasil DATA 2 yang diketahui =');
disp('Masukkan hasil data yang diketahui');
fxm=input('Nilai DATA x yang dihitung =');
% ------------------------------------------------------------------------
%
% proses data interpolasi linier
fx=((fx0)+((fx1-fx0)/(x1-x0))*(x-x0));
galat_mutlak=(fxm-fx);
galat_mutlak_hampiran=((galat_mutlak)/fxm)*100;
disp('------------------------------------------------------------------');
disp('HASIL PERHITUNGAN INTERPOLASI')
disp('------------------------------------------------------------------');
fprintf('Interpolasi linier f(x) yang dicari = %10.3f \n', fx);
fprintf('Galat Mutlak Hampiran     = %10.3f \n', galat_mutlak_hampiran);
```

Gambar. 5.5. Tampilan file-m dari MATLAB untuk Masalah Interpolasi Linier

Penerapan dalam perhitungan pemrograman MATLAB dengan perhitungan interpolasi linier dengan hasil dalam *command window* sebagai berikut.



Gambar. 5.6. Tampilan Command Window dari MATLAB untuk Masalah Interpolasi Linier

Maka melihat hasil yang ditampilkan dalam *command window* disimpulkan bahwa dari hasil perhitungan analitik besaran hasil galat/error dari interpolasi linier sebesar 0,018% dan perhitungan dengan komputasi perhitungan galat mutlak sebesar 0,016% (maka terdapat selisih hasil 0.0002%). Hasil untuk perhitungan nilai f(x) untuk nilai dari ln(9.2) memberikan hasil penyelesaian yang sama dengan nilai eksak ln(9.2) = 2.2192 dan nilai hampiran sebesar 2.2188.

2. Pendekatan Interpolasi Lagrange

Adapun perhitungan simulasi MATLAB pada metode interpolasi lagrange, menyelesaikan permasalahan data berikut. Mengacu dari permasalahan yang dikembangkan pada permasalahan kedua dikembangkan menuju ke orde tiga, perhatikan saksama langkah penyelesaian interpolasi linier dengan menggunakan program MATLAB berikut.

Dengan menggunakan interpolasi polinomial Lagrange orde 3, hitunglah nilai ln 4, apabila diketahui: ln (2) = 0,6931, ln (3) = 1,0986, ln (6) =1,7918 dan ln (9) = 2.1972.

Diketahui:

$x_0=2, \rightarrow f(x_0)=0{,}6931$

$x_1=3, \rightarrow f(x_1)=1{,}0986$

$x_2=6, \rightarrow f(x_2)=1{,}7918$

$x_3=9, \rightarrow f(x_2)=2.1972$

Dari data yang diketahui di atas akan diselesaikan dengan menggunakan program MATLAB, maka disusun terlebih dahulu penggunaan file-m untuk menyelesaikan interpolasi lagrange orde-3. Adapun bentuk susunan dalam script file-m dengan menggunakan formula atau rumus orisinal dari interpolasi lagrange sebagai berikut.

```
1.  % lagrange -- Program analisis numerik
2.  %          -- interpolasi lagrange dengan empat data
    diketahui
3.  %          --  interpolasi  orde-3  dengan  tingkat
    ketelitian 0.0001
4.  % ------------------------------------------------
5.  % author : reza setyanaah
6.  % ------------------------------------------------
```

```
7.
8.   clear; help lagrange;
9.   disp('-----------------------------------------------');
10.  disp('Masukkan data yang diketahui');
11.  disp('-----------------------------------------------');
12.  %input data
13.  x=input('diketahui nilai x= ');
14.  x0=input('diketahui nilai x0= ');
15.  x1=input('diketahui nilai x1= ');
16.  x2=input('diketahui nilai x2= ');
17.  x3=input('diketahui nilai x3= ');
18.  fx0=input('diketahui nilai f(x0) = ');
19.  fx1=input('diketahui nilai f(x1) = ');
20.  fx2=input('diketahui nilai f(x2) = ');
21.  fx3=input('diketahui nilai f(x3) = ');
22.
23.  % ------------------------------------------------------
24.  % proses data
25.  % ------------------------------------------------------
26.  L0=((x-x1)/(x0-x1))*((x-x2)/(x0-x2))*((x-x3)/(x0-x3));
27.  L1=((x-x0)/(x1-x0))*((x-x2)/(x1-x2))*((x-x3)/(x1-x3));
28.  L2=((x-x0)/(x2-x0))*((x-x1)/(x2-x1))*((x-x3)/(x2-x3));
29.  L3=((x-x0)/(x3-x0))*((x-x1)/(x3-x1))*((x-x2)/(x3-x2));
30.  hasil=(L0*fx0)+(L1*fx1)+(L2*fx2)+(L3*fx3);
31.
32.  % ------------------------------------------------------
33.  % tampilan data : interpolasi lagrange orde 3
34.  % ------------------------------------------------------
35.  disp('-----------------------------------------------');
36.  disp('Hasil data interpolasi lagrange orde 3');
37.  disp('-----------------------------------------------');
38.  fprintf('nilai hasil L0= %.4f \n',L0);
39.  fprintf('nilai hasil L1= %.4f \n',L1);
40.  fprintf('nilai hasil L2= %.4f \n',L2);
41.  fprintf('nilai hasil L3= %.4f \n',L3);
42.  fprintf('nilai interpolasi = %.4f \n',hasil);
```

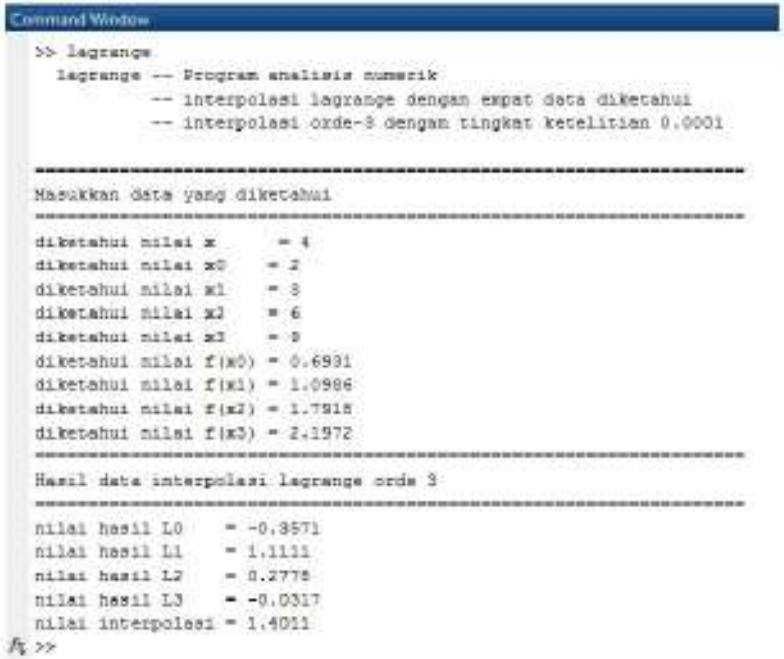
Adapun bentuk tampilan dalam penerapan File-M menggunakan langkah yang sesuai dengan rumusan orisinal yaitu dengan pemberian nama file lagrange.m,

yang dikembangkan menyesuaikan inputan ke dalam m-file.

```
% lagrange -- Program analisis numerik
%          -- interpolasi lagrange dengan empat data diketahui
%          -- interpolasi orde-3 dengan tingkat ketelitian 0.0001

% ------------------------------------------------------------------------
% author : reza setyawan
% ------------------------------------------------------------------------

clear; help lagrange;
disp('------------------------------------------------------------------------');
disp('Masukkan data yang diketahui');
disp('------------------------------------------------------------------------');
%input data
x=input('diketahui nilai x      = ');
x0=input('diketahui nilai x0    = ');
x1=input('diketahui nilai x1    = ');
x2=input('diketahui nilai x2    = ');
x3=input('diketahui nilai x3    = ');
fx0=input('diketahui nilai f(x0) = ');
fx1=input('diketahui nilai f(x1) = ');
fx2=input('diketahui nilai f(x2) = ');
fx3=input('diketahui nilai f(x3) = ');

% ------------------------------------------------------------------------
% proses data
% ------------------------------------------------------------------------
L0=((x-x1)/(x0-x1))*((x-x2)/(x0-x2))*((x-x3)/(x0-x3));
L1=((x-x0)/(x1-x0))*((x-x2)/(x1-x2))*((x-x3)/(x1-x3));
L2=((x-x0)/(x2-x0))*((x-x1)/(x2-x1))*((x-x3)/(x2-x3));
L3=((x-x0)/(x3-x0))*((x-x1)/(x3-x1))*((x-x2)/(x3-x2));
hasil=(L0*fx0)+(L1*fx1)+(L2*fx2)+(L3*fx3);

% ------------------------------------------------------------------------
% tampilan data : interpolasi lagrange orde 3
% ------------------------------------------------------------------------
disp('------------------------------------------------------------------------');
disp('Hasil data interpolasi lagrange orde 3');
disp('------------------------------------------------------------------------');
fprintf('nilai hasil L0    = %.4f \n',L0);
fprintf('nilai hasil L1    = %.4f \n',L1);
fprintf('nilai hasil L2    = %.4f \n',L2);
fprintf('nilai hasil L3    = %.4f \n',L3);
fprintf('nilai interpolasi = %.4f \n',hasil);
```

Gambar. 5.7. Tampilan file-m dari MATLAB untuk Masalah Interpolasi Lagrange

Penerapan dalam perhitungan pemrograman MATLAB dengan perhitungan interpolasi lagrange dengan hasil dalam *command window* sebagai berikut.



Gambar. 5.8. Tampilan Command Window dari MATLAB untuk Masalah Interpolasi Lagrange

Maka melihat hasil yang ditampilkan dalam *command window* disimpulkan bahwa dari hasil perhitungan Hasil untuk perhitungan nilai f(x) untuk nilai dari ln(4) memberikan nilai hampiran sebesar 1.4011 dari perhitungan dengan menggunakan metode interpolasi lagrange.

3. Regresi Linier

Adapun perhitungan simulasi MATLAB pada metode regresi linier, menyelesaikan permasalahan data berikut. perhatikan seksama langkah penyelesaian regresi linier dengan menggunakan program MATLAB berikut.

Tentukan persamaan garis dan koefisien korelasi yang mewakili data berikut.

x	4	6	8	10	14	16	20	22	24	28
y	30	18	22	28	14	22	16	8	20	8

Dari data yang diketahui di atas akan diselesaikan dengan menggunakan program MATLAB, maka disusun terlebih dahulu penggunaan file-m untuk menyelesaikan regresi linier. Adapun bentuk susunan dalam *script* file-m dengan menggunakan formula atau rumus orisinal dari regresi linier yang dikembangkan sebagai berikut.

```
1.  % regresi --- Program analisis numerik
2.  %         ---  menghitung  interpolasi  dan  regresi
    linier
3.  %         --- dengan tingkat ketelitian 0.0001
4.  % ------------------------------------------------
5.  % author : reza setyansah
6.  % ------------------------------------------------
7.  clear; help regresi_linier;
8.  %
9.  % input data
10. disp('-----------------------------------------');
11. disp('DATA YANG DIKETAHUI');
12. disp('-----------------------------------------');
13. xi=input('Masukkan nilai x [dalam array] =');
14. yi=input('Masukkan nilai y [dalam array] =');
15. % proses data nilai
16. xiyi=xi.*yi;
```

```
xi_kuadrat=xi.^2;
nx=length(xi);
ny=length(yi);
disp('-----------------------------------------------');
disp('data nilai xi*yi');
disp(xiyi);
disp('data nilai xi^2');
disp(xi_kuadrat);
disp('-----------------------------------------------');
juml_xi=sum(xi);
juml_yi=sum(yi);
juml_xiyi=sum(xiyi);
juml_xi_kuadrat=sum(xi_kuadrat);
rerata_x=(juml_xi/nx);
rerata_y=(juml_yi/ny);
a1=((nx*juml_xiyi)-
(juml_xi*juml_yi))/((nx*juml_xi_kuadrat)-(juml_xi^2));
a0=(rerata_y-(a1*rerata_x));
disp('-----------------------------------------------');
fprintf('persamaan regresi linier = %.4f + %.4f x
\n',a0,a1);
disp('-----------------------------------------------');
Y_bar=rerata_y;
Dt=yi-Y_bar;
Dt_kuadrat=Dt.^2;
D=((yi)-a0-(xi*a1));
D_kuadrat=D.^2;
Dt_total=sum(Dt_kuadrat);
D_total=sum(D_kuadrat);
r=sqrt((Dt_total-D_total)/Dt_total);
disp('-----------------------------------------------');
fprintf('nilai r = %.4f \n',r);
disp('-----------------------------------------------');
disp('-----------------KATEGORI--------------------');
if (r > 0)
    disp('baik dalam menyesuaikan interval');
else
    if d==1
       disp('sempurna');
    else
       disp('jelek');
    end
end
```

```
58. disp('================KETERANGAN=====================');
59. disp('jika r = 0, kategori jelek');
60. disp('jika r = 1, kategori sempurna');
```

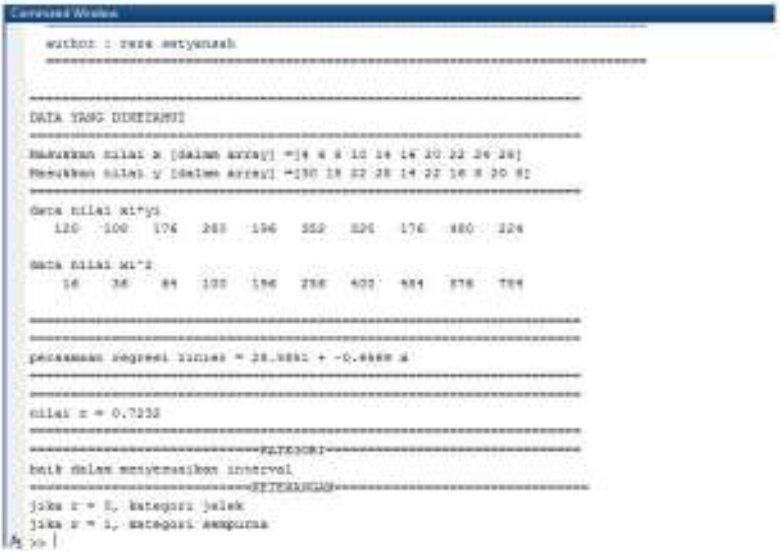
Adapun bentuk tampilan dalam penerapan File-M menggunakan langkah yang sesuai dengan rumusan orisinal yaitu dengan pemberian nama file regresi_linier.m, yang dikembangkan menyesuaikan inputan ke dalam m-file.

```
% regresi    ---- Program analisis numerik
%            ---- menghitung interpolasi dan regresi linier
%            ---- dengan ringkat ketelitian 0.0001
% ========================================================================
% author : resa setyanah
% ========================================================================
clear; help regresi_linier;
%
% input data
disp('========================================================================');
disp('DATA YANG DIKETAHUI');
disp('========================================================================');
xi=input('Masukkan nilai x [dalam array] =');
yi=input('Masukkan nilai y [dalam array] =');
% proses data nilai
xiyi=xi.*yi;
xi_kuadrat=xi.^2;
nx=length(xi);
ny=length(yi);
disp('========================================================================');
disp('data nilai xi*yi');
disp(xiyi);
disp('data nilai xi^2');
disp(xi_kuadrat);
disp('========================================================================');
jumlah_xi=sum(xi);
jumlah_yi=sum(yi);
jumlah_xiyi=sum(xiyi);
jumlah_xi_kuadrat=sum(xi_kuadrat);
rerata_x=(jumlah_xi/nx);
rerata_y=(jumlah_yi/ny);
```

```
rerata_y=(juml_yi/n);
a1=((n*juml_xiyi)-(juml_xi*juml_yi))/((n*juml_xi_kuadrat)-(juml_xi^2));
a0=(rerata_y-(a1*rerata_x));
disp('=============================================================');
fprintf('persamaan regresi linier = %.4f + %.4f x \n',a0,a1);
disp('=============================================================');
Y_bar=rerata_y;
Dt=yi-Y_bar;
Dt_kuadrat=Dt.^2;
D=(yi)-a0-(a1*xi));
D_kuadrat=D.^2;
Dt_total=sum(Dt_kuadrat);
D_total=sum(D_kuadrat);
r=sqrt((Dt_total-D_total)/Dt_total);
disp('=============================================================');
fprintf('nilai r = %.4f \n',r);
disp('=============================================================');
disp('============================KATEGORI=========================');
if (r > 0)
    disp('baik dalam menyesuaikan interval');
else
    if d==1
        disp('sempurna');
    else
        disp('jelek');
    end
end
disp('============================KETERANGAN=======================');
disp('jika r = 0, kategori jelek');
disp('jika r = 1, kategori sempurna');
```

Gambar. 5.9. Tampilan m-file dari MATLAB untuk Masalah Regresi Linier

Penerapan dalam perhitungan pemrograman MATLAB dengan perhitungan regresi linier dengan hasil dalam *command window* sebagai berikut.



Gambar. 5.10. Tampilan Command Window dari MATLAB untuk Masalah Regresi Linier

Maka melihat hasil yang ditampilkan dalam *command window* disimpulkan bahwa dari hasil perhitungan Hasil untuk perhitungan nilai secara penggunaan analitik diperoleh a_0 = 28,5849 dan a_1 = - 0,6569 sehingga persamaan garisnya adalah y = 28,5849 – 0,6569 x dengan koefisien korelasi sebesar 0,7232. Adapun hasil perhitungan secara komputasi MATLAB yaitu y = 28,5851 – 0,6569 x dan nilai r = 0,7232.

4. Regresi Polinomial

Adapun perhitungan simulasi MATLAB pada metode regresi polinomial, menyelesaikan permasalahan data berikut.

Carilah persamaan kurva polinomial order dua yang mewakili data berikut:

x	0	2	3	4	5	6
y	5	5	8	6	5	5

Dari data yang diketahui di atas akan diselesaikan dengan menggunakan program MATLAB, dengan menyusun m-file terlebih dahulu dengan susunan perintah sebagai berikut.

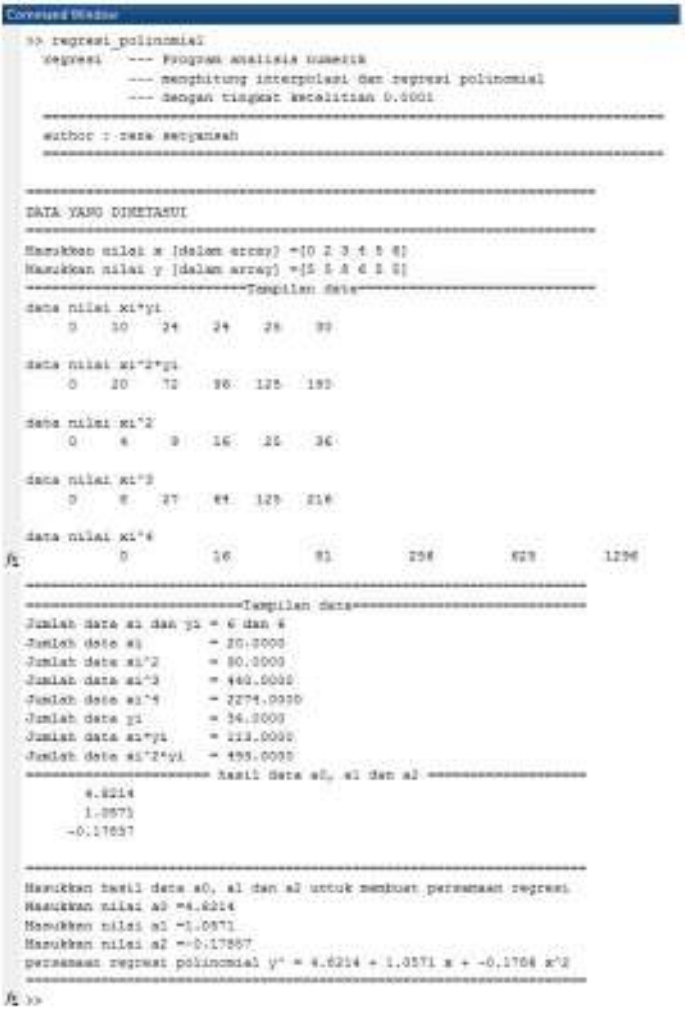
```
% regresi --- Program analisis numerik
%          --- menghitung interpolasi dan regresi polinomial
%          --- dengan tingkat ketelitian 0,0001
% ------------------------------------------------------------
% author : reza setyanssh
% ------------------------------------------------------------
clear; help regresi_polinomial;
%
% input data
disp('-------------------------------------------------');
disp('DATA YANG DIKETAHUI');
disp('-------------------------------------------------');
xi=input('Masukkan nilai x [dalam array] =');
yi=input('Masukkan nilai y [dalam array] =');
% proses data nilai
xiyi=xi.*yi;
xi_kuadrat=xi.^2;
xi2yi=xi_kuadrat.*yi;
xi_triple=xi.^3;
xi_quad=xi.^4;
nx=length(xi);
ny=length(yi);
disp('-----------------Tampilan Data-----------------');
disp('data nilai xi*yi');
disp(xiyi);
disp('data nilai xi^2*yi');
disp(xi2yi);
disp('data nilai xi^2');
disp(xi_kuadrat);
disp('data nilai xi^3');
disp(xi_triple);
disp('data nilai xi^4');
disp(xi_quad)
disp('-------------------------------------------------');
jum_xi=sum(xi);
```

```
juml_yi=sum(yi);
juml_xiyi=sum(xiyi);
juml_xi2yi=sum(xi2yi);
juml_xi_kuadrat=sum(xi_kuadrat);
juml_xi_triple=sum(xi_triple);
juml_xi_quad=sum(xi_quad);
rerata_x=(juml_xi/nx);
rerata_y=(juml_yi/ny);
disp('==============Tampilan data================');
fprintf('Jumlah data xi dan yi= %.0f dan %.0f \n',nx,ny);
fprintf('Jumlah data xi     = %.4f \n',juml_xi);
fprintf('Jumlah data xi^2   = %.4f \n',juml_xi_kuadrat);
fprintf('Jumlah data xi^3   = %.4f \n',juml_xi_triple);
fprintf('Jumlah data xi^4   = %.4f \n',juml_xi_quad);
fprintf('Jumlah data yi     = %.4f \n',juml_yi);
fprintf('Jumlah data xi*yi  = %.4f \n',juml_xiyi);
fprintf('Jumlah data xi^2*yi = %.4f \n',juml_xi2yi);
% ---------------------proses data----------------------
A = [nx juml_xi juml_xi_kuadrat;juml_xi juml_xi_kuadrat
juml_xi_triple;juml_xi_kuadrat juml_xi_triple juml_xi_quad];
b = [juml_yi;juml_xiyi;juml_xi2yi];
det(A);
x=inv(A)*b;
disp('---------- hasil data a0, a1 dan a2 ----------');
disp(x);
disp('----------------------------------------------');
disp('Masukkan hasil data a0, a1 dan a2 untuk membuat
persamaan regresi');
a0=input('Masukkan nilai a0 =');
a1=input('Masukkan nilai a1 =');
a2=input('Masukkan nilai a2 =');
fprintf('persamaan regresi polinomial y^ = %.4f + %.4f x +
%.4f x^2 \n',a0,a1,a2);
disp('----------------------------------------------');
```

Tampilan dari penggunaan file-m untuk menyelesaikan regresi linier. Adapun bentuk susunan dalam *script* file-m dengan menggunakan formula atau rumus orisinal dari regresi linier yang dikembangkan sebagai berikut.

Gambar. 5.11. Tampilan m-file dari MATLAB untuk Masalah Regresi Polinomial

Penerapan dalam perhitungan pemrograman MATLAB dengan perhitungan regresi polinomial dengan hasil dalam *command window* sebagai berikut.



Gambar. 5.12. Tampilan Command Window dari MATLAB untuk Masalah Regresi Polinomial

Hasil perhitungan secara analitik diperoleh sebesar $a_0 = 4{,}821429$; $a_1 = 1{,}057143$ dan $a_2 = -0{,}178571$ sehingga diperoleh persamaan berikut.

$$\hat{y} = 4{,}821429 + 1{,}057143x - 0{,}718571x^2$$

Hal tersebut, tidak berbeda dengan penyelesaian dari perhitungan komputasi regresi polynomial dengan program MATLAB.

SIMULASI 6: INTEGRASI NUMERIK

6.1. Pendahuluan

Permasalahan dalam perhitungan integrasi adalah perhitungan dasar yang digunakan dalam kalkulus untuk berbagai keperluan. Integral suatu fungsi adalah operator matematik yang direpresentasikan dalam bentuk tertentu untuk menghitung luas daerah yang dibatasi oleh fungsi y=f(x) dan sumbu x.

Luasan daerah tersebut yang diarsir L dapat dihitung dengan:

$$L = \int_a^b f(x)\,dx$$

Persamaan di atas merupakan integral suatu fungsi f(x) terhadap variabel x dengan batas integrasi dari x=a sampai x=b. Integral merupakan nilai total atau luasan yang dibatasi oleh fungsi(x)dan sumbu x, serta antara batas x=a dan x=b.

Integral secara analitis dapat diselesaikan menjadi bentuk berikut.

$$\int_a^b f(x)\,dx = [F(x)]_a^b = F(b) - F(a)$$

dengan F(x) adalah integral dari f(x) sedemikian sehingga F'(x) = f(x).

Tetapi ada beberapa permasalahan integral sulit sekali dihitung bahkan dikatakan tidak dapat

dilakukan perhitungan analitik atau perhitungan manual, sebagai contoh:

1. Perhitungan yang sukar untuk diselesaikan secara analitik.

$$\int_0^2 \frac{2+\cos(1+x^{3/2})}{\sqrt{1+0.5\sin x}} e^{0.5x}\,dx$$

2. Perhitungan yang diketahui tidak diberikan dalam bentuk analitik, tetapi dalam bentuk angka dalam tabel.
3. Penerapan perhitungan integral yang menghitung luas daerah atau area peta, volume permukaan tanah, menghitung luas dan volume benda putar dimana fungsi f(x) tidak dituliskan hanya digunakan gambar untuk menyajikan nilai f(x).

Penyelesaian komputasi integrasi numerik dapat menggunakan beberapa metode, antara lain:

1. Metode trapezium (*trapezoida*)
   Metode pendekatan integral numerik dengan persamaan polinomial dengan orde satu.
2. Metode simpson
   Metode pendekatan integral numerik dengan persamaan polinomial polynomial tinggi.
3. Metode simpson aturan 1/3 dengan banyak pias.
   Metode Simpson yang dapat diperbaiki dengan membagi luasan dalam sejumlah pias dengan panjang interval yang sama dan jumlah intervalnya genap

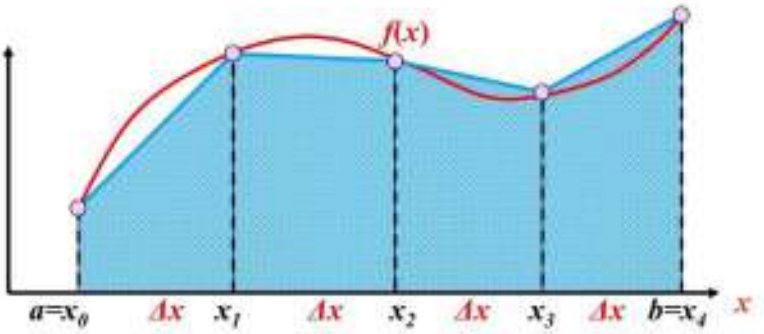
4. Metode simpson aturan 3/8.
   Metode pendekatan integral numerik dengan persamaan polinomial polynomial tinggi diturunkan diturunkan dengan menggunakan persamaan polinomial order tiga yang melalui empat titik.

6.2. Metode Integrasi Trapezium

Perhitungan integrasi numerik dengan menggunakan satu pias.

$$I = (b - a)\frac{f(a) + f(b)}{2}$$

Pada setiap daerah bagian yang dinyatakan sebagai empat persegi panjang dengan tinggi $f(x_i)$ dan lebar x_i. Maka setiap bagian yang dinyatakan dalam trapezium seperti pada gambar berikut.



Gambar. 6.1. Metode Integrasi Trapezium Banyak Pias

Panjang tiap pias yang disajikan sama yaitu Δx. Jika terdapat pias maka panjang pada masing-masing pias adalah:

$$\Delta x = \frac{b-a}{n}$$

Rumus dalam perhitungan trapezium dengan banyak pias, dinyatakan sebagai berikut.

$$I = \frac{\Delta x}{2}[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)]$$

Adapun bentuk dengan luas bagian dengan rumus, bentuk rumus dengan persamaan trapezium dengan koreksi ujung.

$$I = \frac{\Delta x}{2}[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)] - \frac{\Delta x^2}{12}[f'(b) - f'(a)]$$

Maka untuk memahami proses perhitungan amati contoh permasalahan sebagai berikut.

Contoh pertama membahas penyelesaian integrasi berikut.

$$I = \int_0^4 e^x dx$$

Maka langkah pertama menentukan dengan menggunakan metode trapesium satu pias untuk menghitung fungsi tersebut. Terlebih dahulu dilakukan penghitungan bentuk integral pada soal di atas secara analitis sebagai berikut.

$$I = \int_0^4 e^x dx = \left[e^x\right]_0^4 = \left(e^4 - e^0\right) = 53{,}598150$$

Penghitungan dengan metode trapesium satu pias:

$$I \approx (b-a)\frac{f(a)+f(b)}{2} = (4-0)\frac{e^4+e^0}{2} = 111{,}1963$$

Untuk mengetahui tingkat ketelitian, dibandingkan hasil hitungan numerik dan analitis:

$$\varepsilon = \frac{|53{,}598750 - 111{,}1963|}{53{,}598750} \times 100\% = 107{,}46\%$$

Terlihat bahwa hasil menggunakan metode trapesium satu pias memberikan kesalahan sangat besar (lebih dari 100%)

Apabila menggunakan trapezium dengan panjang banyak pias, akan memberikan data perhitungan sebagai berikut.

$$\Delta x = \frac{(b-a)}{n} = \frac{(4-0)}{4} = 1$$

Luas bidang dihitung dengan metode trapesium dengan banyak pias:

$$I = \frac{\Delta x}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)\right]$$

$$I = \frac{1}{2}\left[e^0 + e^4 + 2(e^1 + e^2 + e^3)\right] = 57{,}991950$$

Kesalahan relatif terhadap nilai eksak:

$$\varepsilon = E_g = \frac{|53{,}598150 - 57{,}991950|}{53{,}598150} \times 100\% = 8{,}2\%$$

Apabila digunakan metode trapesium dengan koreksi ujung, maka integral dihitung sebagai berikut:

$$I = \frac{\Delta x}{2}[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)] - \frac{\Delta x^2}{12}[f'(b) - f'(a)]$$

$$I = \frac{1}{2}[e^0 + e^4 + 2(e^1 + e^2 + e^3)] - \frac{1}{12}(e^4 - e^0)$$

$$I = 57{,}991950 - 4{,}466513$$

$$I = 53{,}525437$$

Kesalahan/galat relatif terhadap nilai eksak

$$\varepsilon = E_g = \frac{53{,}598150 - 53{,}525437}{53{,}598150} \times 100\% = 0{,}14\%$$

Melihat dari hasil perolehan kesalahan/galat relatif dari metode trapesium satu pias sebesar 107,46%, metode trapesium dengan banyak pias sebesar 8,2% dan metode trapesium dengan koreksi ujung sebesar 0,14%. Sehingga disimpulkan bahwa dengan menggunakan metode trapesium dengan koreksi ujung memberikan hasil kesalahan/galat relatif terhadap nilai eksak yang lebih baik.

Perhitungan pada metode trapezium dapat dilakukan pada contoh yang kedua dengan penyelesaian sebagai berikut.

Diberikan data tabel berikut.

x	0	1	2	3	4
f(x)	1	3	9	19	33

Hitung luasan f(x) diantara x=0 dan x=4 dengan menggunakan Metode trapesium koreksi ujung.

Integral numerik, sebagai bentuk perhitungan analitik:

$$I = \frac{\Delta x}{2}[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)]$$

$$I = \frac{1}{2}[1 + 33 + 2(3 + 9 + 19)] = 48$$

Integral numerik menggunakan metode trapesium koreksi ujung.

$$I = \frac{\Delta x}{2}[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(x_i)] - \frac{\Delta x^2}{12}[f'(b) - f'(a)]$$

$$f'(x_1 = a = 0) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{f(1) - f(0)}{1 - 0} = \frac{3 - 1}{1} = 2$$

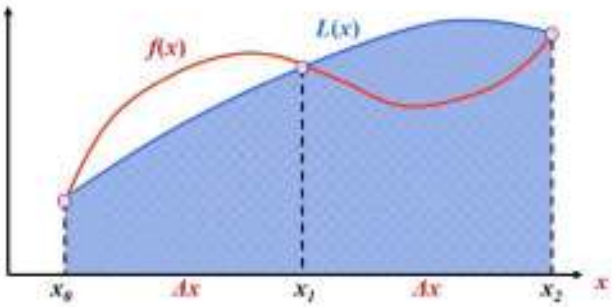
$$f'(x_n = b = 4) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = \frac{f(4) - f(3)}{4 - 3} = \frac{33 - 19}{1} = 14$$

$$I = \frac{1}{2}[1 + 33 + 2(3 + 9 + 19)] - \frac{1}{12}(14 - 2) = 48 - 1 = 47$$

Melihat dari penyelesaian integral numerik dan Integral numerik menggunakan metode trapesium koreksi ujung memberikan hasil yang mendekati untuk nilai satuan luas.

6.3. Metode Integrasi Simpson (Aturan 1/3)

Perhitungan integrasi numerik dengan menggunakan simpson (Aturan 1/3).



Gambar. 6.2. Metode Integrasi Simpson (Aturan 1/3)

Bentuk persamaan dengan metode simpson 1/3 diberikan tambahan 1/3 karena Δx dibagi dengan 3.

$$A = \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + f(x_2)] + O(\Delta x^5)$$

Pada pemakaian satu pias, $\Delta x = \frac{b-a}{2}$, sehingga persamaan metode simpson 1/3 dengan satu pias dituliskan dalam bentuk:

$$A = \frac{b-a}{6}[f(a) + 4f(c) + f(b)]$$

Dengan c adalah titik tengah antara a dan b.

Dengan permasalahan yang sama dari metode trapezium akan dibandingkan dengan perhitungan:

$$I = \int_0^4 e^x dx$$

Dengan menggunakan aturan simpson 1/3, maka luas bidang.

$$I = \int_0^4 e^x dx$$

Penghitungan bentuk integral pada soal di atas dapat diselesaikan secara analitis:

$$I = \int_0^4 e^x dx = \left[e^x\right]_0^4 = \left(e^4 - e^0\right) = 53{,}598150$$

Penghitungan dengan aturan simpson 1/3:

$$A_i = \frac{b-a}{6}[f(a) + 4f(c) + f(b)] = \frac{4-0}{6}[e^0 + 4e^2 + e^4] = 56{,}7696$$

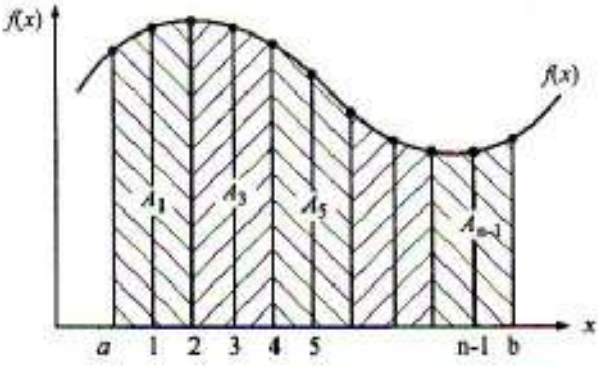
Untuk mengetahui tingkat ketelitian, dibandingkan hasil hitungan numerik dan analitis:

$$\varepsilon = \frac{53{,}598750 - 56{,}7696}{53{,}598750} \times 100\% = 5{,}917\%$$

Terlihat bahwa hasil menggunakan metode aturan simpson 1/3 memberikan hasil lebih baik dari metode trapesium.

6.4. Aturan Simpson 1/3 dengan Banyak Pias

Metode simpson dapat diperbaiki dengan membagi luasan dalam sejumlah pias dengan panjang interval yang sama dan jumlah intervalnya genap.



Gambar. 6.3. Metode Integrasi Simpson (Aturan 1/3) Banyak Pias

Dimana, dengan n adalah jumlah pias

$$\Delta x = \frac{b-a}{n}$$

Luas total diperoleh dengan menjumlahkan semua pias,

$$\int_a^b f(x)\,dx = A_1 + A_3 + \ldots + A_{n-1}$$

Dengan mensubstitusikan:

$$A_i = \frac{\Delta x}{3}(f_{i-1} + 4f_i + f_{i+1}) + O(\Delta x^5)$$

ke persamaan di atas diperoleh

$$\int_a^b f(x)dx = \frac{\Delta x}{3}(f_0 + 4f_1 + f_2) + \frac{\Delta x}{3}(f_2 + 4f_3 + f_4) + \ldots + \frac{\Delta x}{3}(f_{n-2} + 4f_{n-1} + f_n)$$

atau

$$\int_a^b f(x)dx = \frac{\Delta x}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{n-1} f(x_i) + 2\sum_{i=2}^{n-2} f(x_i) \right]$$

Jadi aturan Simpson 1/3 dengan banyak pias, menggunakan persamaan dimana:
Dengan suku

$$4\sum_{i=1}^{n-1} f(x_i)$$

adalah untuk nilai i ganjil (i=1,3,5,...)
dan untuk

$$2\sum_{i=2}^{n-2} f(x_i)$$

adalah untuk nilai i genap (i=2,4,6,...)

Maka untuk memahami proses perhitungan amati contoh permasalahan sebagai berikut. Adapun perkiraan kesalahan yang terjadi pada aturan Simpson 1/3 untuk banyak pias sebagai berikut.

$$I = \int_0^4 e^x dx$$

.

Perhitungan panjang pias, sehingga diperoleh

$$\Delta x = \frac{(b-a)}{n} = \frac{(4-0)}{4} = 1$$

Luas bidang dihitung dengan metode simpson dengan banyak pias:

$$\int_a^b f(x)dx = \frac{\Delta x}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{n-1} f(x_i) + 2\sum_{i=2}^{n-2} f(x_i) \right]$$

$$I = \frac{1}{3}[e^0 + e^4 + 4(e^1 + e^3) + 2e^2] = 53{,}863846$$

Kesalahan relatif terhadap nilai eksak:

$$\varepsilon = E_R = \frac{53{,}598150 - 53{,}863846}{53{,}598150} \times 100\% = 0{,}5\%$$

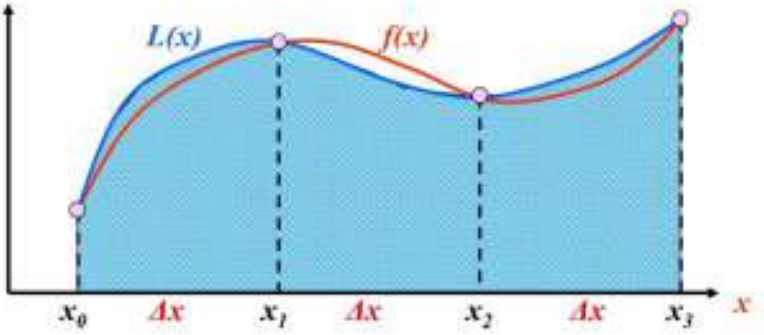
Terlihat bahwa hasil menggunakan metode aturan simpson 1/3 dengan panjang pias berarti memberikan hasil lebih baik dari metode simpson.

6.5. Metode Integrasi Simpson (Aturan 3/8)

Metode Simpson 3/8 diturunkan dengan menggunakan persamaan polinomial order tiga yang melalui empat titik.

$$I = \int_a^b f(x)\,dx \approx \int_a^b f_3(x)\,dx$$

Sehingga tampilan dalam grafik metode integrasi simpson (Aturan 3/8) maka didapatkan bentuk sebagai berikut.



Gambar. 6.4. Metode Integrasi Simpson (Aturan 3/8)

Dengan cara yang sama pada penurunan aturan Simpson 1/3, akhirnya diperoleh:

$$I = \frac{3\Delta x}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]$$

dengan

$$\Delta x = \frac{b-a}{3}$$

Jadi, metode Simpson 3/8 dapat juga ditulis dalam bentuk:

$$I = (b-a)\frac{\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]}{8}$$

Metode Simpson 1/3 biasanya lebih disukai karena mencapai ketelitian order tiga dan hanya memerlukan tiga titik, dibandingkan metode Simpson 3/8 yang membutuhkan empat titik. Dalam pemakaian banyak pias, metode Simpson 1/3 hanya berlaku untuk jumlah pias genap. Apabila dikehendaki jumlah pias ganjil,

maka dapat digunakan metode trapesium. Tetapi metode ini tidak begitu baik karena adanya kesalahan yang cukup besar. Untuk itu kedua metode dapat digabung, yaitu sejumlah genap pias digunakan metode Simpson 1/3 sedang 3 pias sisanya digunakan metode Simpson 3/8.

Dengan permasalahan yang sama dari metode trapezium akan dibandingkan dengan perhitungan:

$$I = \int_0^4 e^x dx$$

Apabila diketahui

$$\Delta x = \frac{(b-a)}{n} = \frac{(4-0)}{n} = 0{,}8$$

$$n = \frac{4}{0{,}8} = 5$$

Hitung integral dengan metode simpson 3/8, apabila digunakan satu pias, sebagai berikut:
Metode simpson 3/8 dengan satu pias.

$$I = (b-a)\frac{\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]}{8}$$

$$\mathrm{I} = (4-0)\frac{\left[e^0 + 3e^{1{,}3333} + 3e^{2{,}6667} + e^4\right]}{8} = 55{,}07798$$

Kesalahan relatif terhadap nilai eksak:

$$\varepsilon = E_R = \frac{53{,}598150 - 55{,}07789}{53{,}598150} \times 100\% = -2{,}761\%$$

Apabila diketahui,

$$\Delta x = \frac{(b-a)}{n} = \frac{(4-0)}{n} = 0{,}8$$

$$n = \frac{4}{0{,}8} = 5$$

Hitung integral dengan metode simpson 1/3 dan 3/8, apabila digunakan 5 pias, sebagai berikut:

$$f(0) = e^{0} = 1$$

$$f(0{,}8) = e^{0.8} = 2{,}22554$$

$$f(1{,}6) = e^{1.6} = 4{,}95303$$

$$f(2{,}4) = e^{2.4} = 11{,}02318$$

$$f(3{,}2) = e^{3.2} = 24{,}53253$$

$$f(4) = e^{4} = 54{,}59815$$

6.6. Program MATLAB

Pembahasan dalam program MATLAB mengembangkan metode perhitungan dalam sintaksnya dibahas sebagaimana berikut ini. Bentuk pembahasan menggunakan metode trapesium, metode simpson (Aturan 1/3) dan metode simpson (Aturan 1/3). Adapun langkah-langkah dalam program MATLAB menggunakan *command window* dan *File-M*. Adapun simulasi dengan menggunakan program MATLAB terbagi menjadi beberapa tahap berikut ini.

1. Metode Trapezium

Adapun perhitungan simulasi MATLAB pada metode integrasi trapezium, menyelesaikan permasalahan data berikut. Mengacu dari permasalahan yang dikembangkan dari contoh, perhatikan saksama langkah penyelesaian metode integrasi trapezium dengan menggunakan program MATLAB berikut.

$$I = \int_0^4 e^x dx$$

Langkah dalam penyelesaian perhitungan simulasi MATLAB pada metode integrasi trapezium, menggunakan command window untuk menyelesaikan data tersebut sebagai berikut.

```
1. % Penerapan metode trapezium dalam command
   window
2. x = [0:1:4]
3. y = exp(x)
4. trapz(x,y)
```

Penerapan dalam perhitungan pemograman MATLAB dengan perhitungan integrasi trapezium dengan hasil dalam command window sebagai berikut.

Gambar. 6.5. Tampilan File-M MATLAB Metode Integrasi Trapezium

Diperoleh dari penghitungan sebelumnya dengan manual metode trapesium sebesar berikut.

$$I = \frac{1}{2}[e^0 + e^4 + 2(e^1 + e^2 + e^3)] = 57{,}991950$$

Sedemikian sehingga perolehan hasil di atas, menunjukkan hasil yang sama dengan perolehan perhitungan komputasi. Untuk perolehan nilai perhitungan manual dengan metode trapezium dengan diperoleh perhitungan yang sesuai.

2. Metode Simpson

Adapun perhitungan simulasi MATLAB pada metode integrasi simpson, menyelesaikan permasalahan data berikut. Mengacu dari permasalahan yang dikembangkan dari yang serupa dengan metode trapezium. Langkah dalam penyelesaian perhitungan simulasi MATLAB pada perhitungan dalam metode integrasi simpson, menggunakan skrip dari

KUMPULAN LATIHAN ANALISIS NUMERIK

Simulasi 1

Tentukan pembandingan perhitungan analitik dan perhitungan numerik permasalahan berikut ini.

a) $I = \int_{-2}^{2}(9 + x^2)dx$

b) $I = \int_{-1}^{5}\left(\sqrt{2x} + 10\right)dx$

c) $I = \int_{0}^{4}(15 - \log 2x)dx$

Simulasi 2

Tentukan perhitungan numerik permasalahan berikut ini.

a) Diketahui suatu fungsi $f(x) = 0{,}25x^3 + 0{,}5x^2 + 0{,}25x + 0{,}5$. Perkirakan fungsi tersebut dengan menggunakan deret Taylor order nol, satu, dua dan tiga pada titik $x_{i+1}=1$, berdasar nilai fungsi pada titik $x_i=0$. Titik x_{i+1} berada pada jarak $\Delta x = 1$ dari titik $x_i=0$

b) Diketahui suatu fungsi $f(x) = -2x^3 + 12x^2 - 20x + 8{,}5$. Perkirakan fungsi tersebut dengan menggunakan deret Taylor order nol, satu, dua dan tiga. Perkirakan fungsi tersebut pada titik $x_{i+1}=0{,}5$, berdasar nilai fungsi pada titik $x_i=0$

c) Diketahui suatu fungsi $f(x) = -0{,}2x^3 + 1{,}2x^2 - 0{,}21x + 1{,}5$. Perkirakan fungsi tersebut dengan menggunakan deret Taylor order nol, satu, dua

dan tiga. Perkirakan fungsi tersebut pada titik xi+1=1, berdasar nilai fungsi pada titik xi=0.

Simulasi 3

Tentukan perhitungan numerik permasalahan berikut ini, dengan menggunakan metode table, biseksi, interpolasi linier dan newton rhapson.

a. $f(x) = x^3 - 5x^2 - x + 2$

b. $f(x) = x^3 - 3x - 1$

c. $f(x) = 3x^5 - 5x^3 + 1$

d. $f(x) = x^3 - 3x^2 + 2$

e. $f(x) = x^3 - 3x + 4$

f. $f(x) = x \cdot e^{-x}$

Simulasi 4

Tentukan perhitungan penyelesaian sistem persamaan linier berikut dengan menggunakan metode Eliminasi Gauss, Jacobi dan Gauss-Seidel.

a) $x - 2y + 5z = 12$
$x + 4y + 2z = 15$
$5x + y - z = 4$

b) $x - 2y + 5z = 12$
$x + 4y + 2z = 15$
$5x + y - z = 4$

Simulasi 5

Tentukan perhitungan penyelesaian berikut ini.

a) Diketahui dari data hubungan antara nilai x dan y berikut ini. Carilah persamaan kurva yang mewakili data berikut dan hitung koefisien korelasinya dengan menggunakan regresi linier dan regresi polinomial.

X_i	60	45	50	60	50	65	60	65	50	65	45	50
Y_i	80	69	71	85	80	82	89	93	76	86	71	69

b) Bandingkan perhitungan dengan menggunakan interpolasi linier dan interpolasi polinomial Lagrange orde 1, carilah nilai data nilai dari ln 3, apabila diketahui: ln 2 = 0,69314718 dan ln 6 = 1,791759469.

c) Diberikan tabel data berikut ini:

x	6	10	14	18	22
f(x)	21	28	30	35	39

Hitung f(19) dengan menggunakan Interpolasi Lagrange Orde 2

Simulasi 6

a) Hitunglah integral berikut $\int_4^{15} 2x^2 - x + 9$ dengan aturan Integrasi Trapesium dan Integrasi Simpson jika besar interval $\Delta x = 1.1$ dan tentukan galat relatifnya.

b) Hitunglah integral berikut $\int_1^{-1\frac{3}{2}} x^2 + 2x - 10$ dengan aturan Integrasi Trapesium dan Integrasi

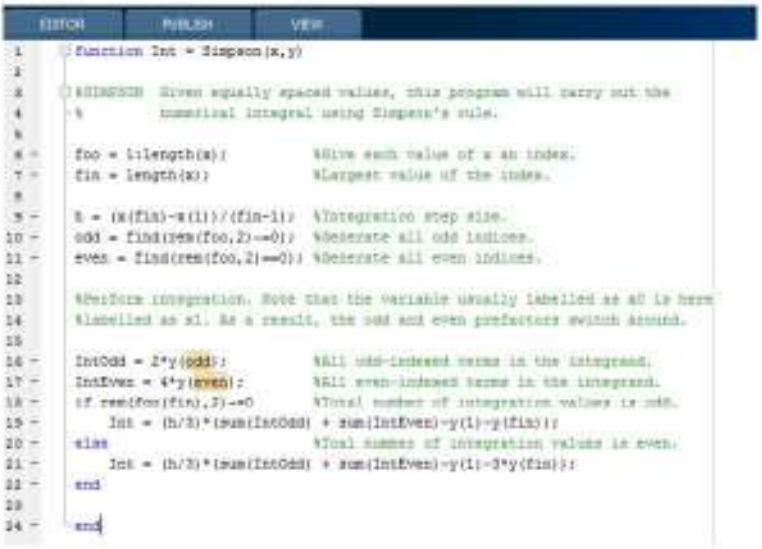
Simpson jika besar interval Δx = 1.1 dan tentukan galat relatifnya.

c) Hitunglah integral berikut $\int_3^0 \frac{2x^2-x+9}{x+1}$ dengan aturan Integrasi Trapesium dan Integrasi Simpson jika besar interval Δx = 1.1 dan tentukan galat relatifnya.

https://www.mathworks.com/matlabcentral/fileexchange/25255-simpson untuk menyelesaikan data tersebut sebagai berikut.

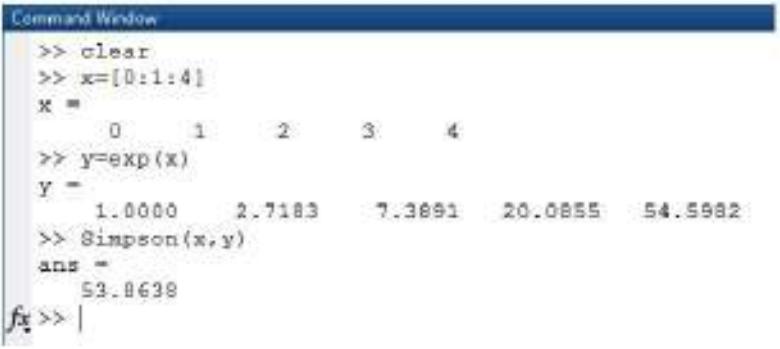
```
function Int = Simpson(x,y)

%SIMPSONGiven equally spaced values, this program will carry out the
% numerical integral using Simpson's rule.

foo = 1:length(x);              %Give each value of x an index.
fin = length(x);                %Largest value of the index.

h = (x(fin)-x(1))/(fin-1);      %Integration step size.
odd = find(rem(foo,2)==0);      %Generate all odd indices.
even = find(rem(foo,2)==0);     %Generate all even indices.

%Perform integration. Note that the variable usually labelled as x0 is here
%labelled as x1. As a result, the odd and even prefactors switch around.

IntOdd = 2*y(odd);      %All odd-indexed terms in the integrand.
IntEven = 4*y(even);    %All even-indexed terms in the integrand.
if rem(foo(fin),2)==0   %Total number of integration values is odd.
    Int = (h/3)*(sum(IntOdd) + sum(IntEven)-y(1)-y(fin));
Else                    %Toal number of integration values is even.
    Int = (h/3)*(sum(IntOdd) + sum(IntEven)-y(1)-3*y(fin));
end

end
```

Tampilan dari penggunaan file-m untuk menyelesaikan metode integrasi simpson. Adapun bentuk susunan dalam *script file-m* disimpan dengan nama file Simpson.m dengan menggunakan milik dari mathworks sebagai berikut.



Gambar. 6.6. Tampilan File-M MATLAB Metode Integrasi Simpson

Penerapan dalam perhitungan pemrograman MATLAB dengan perhitungan integrasi simpson untuk panjang pias dengan hasil dalam command window sebagai berikut.



Gambar. 6.7. Tampilan Command Window MATLAB Metode Integrasi Simpson Panjang Pias

$$I = (4-0)\frac{\left[e^{0} + 3e^{1,3333} + 3e^{2,6667} + e^{4}\right]}{8} = 55,07798$$

Sedemikian sehingga perolehan hasil di atas, menunjukkan hasil yang sama dengan perolehan perhitungan komputasi. Untuk perolehan nilai perhitungan manual dengan metode simpson dengan diperoleh perhitungan yang berbeda dengan selisih pivot.

Daftar Pustaka

Basuki, Achmad. Ramadijanti, Nana. 2005. *Metode Numerik dan Algoritma Komputasi*. Yogyakarta: Penerbit Andi OFFSET.

Carnahan. B. Luther. H. A., Wilkes. H.A. 1969. *Applied Numerical Methods*, New York: John Wiley & Sons Company.

Chapra Steven C., Canale Raymond P. 1985. *Numerical Methods For Engineers*. New York: Mc. Graw-Hill Book Company.

Hamming, R. W. 1973. *Numerical Methods for Scientists and Engineers*. Second Edition. New York: Dover Publications, Inc.

https://www.mathworks.com/

Sahid. 2005. *Komputasi Numerik dengan Matlab*. Yogyakarta: Penerbit Andi Offset

Suarga. 2014. *Komputasi Numerik – Pemrograman MATLAB untuk Metoda Numerik*. Yogyakarta: Penerbit Andi OFFSET.

Tjolleng, Amir. 2017. *Pengantar Pemrograman MATLAB*. Jakarta: PT. Elex Media Komputindo.

Triatmodjo, Bambang. 2002. *Metode Numerik*. Jakarta: Penerbit Beta Offset

Tentang Penulis

Reza Kusuma Setyansah, S.Pd., M.Pd.

Penulis berasal dari kota madiun, kelahiran bulan Januari tahun 1987. Setelah pendidikan di jenjang SMA selesai, melanjutkan pendidikannya dengan mengambil keilmuan Program Studi Sarjana Pendidikan Matematika di IKIP PGRI Madiun yang saat ini telah berubah menjadi Universitas PGRI Madiun pada tahun 2005 sampai dengan tahun 2010. Setelah menyelesaikan gelar sarjana kemudian melanjutkan pascasarjana pendidikan matematika di Universitas Sebelas Maret Surakarta pada tahun 2010 sampai dengan 2012. Saat ini aktif mengajar di Universitas PGRI Madiun mulai tahun 2012 sampai sekarang di Program Studi Pendidikan Matematika. Kebidangan yang ditekuni mengarah pada algoritma yaitu program komputer matematika, operasi riset, dan analisis numerik. Bidang administrasi unit yang dikerjakan di Universitas PGRI Madiun selaku Sekretaris UPT Komputer dengan masa jabatan 2017 – 2021.

Davi Apriandi, S.Pd.Si., M.Pd.

Penulis berasal dari Bantul, kelahiran bulan April tahun 1987. Setelah pendidikan di jenjang SMA selesai, melanjutkan pendidikannya dengan mengambil keilmuan Program Studi Sarjana Pendidikan Matematika di Universitas Negeri Yogyakarta pada

tahun 2005 sampai dengan tahun 2010. Setelah menyelesaikan gelar sarjana kemudian melanjutkan pascasarjana pendidikan matematika di Universitas Sebelas Maret Surakarta pada tahun 2010 sampai dengan 2012. Saat ini aktif mengajar di Universitas PGRI Madiun mulai tahun 2012 sampai sekarang di Program Studi Pendidikan Matematika. Kebidangan yang ditekuni mengarah pada kalkulus yaitu kalkulus lanjut dan analisis numerik. Bidang administrasi unit yang dikerjakan di Universitas PGRI Madiun selaku Kepala bidang kewirausahaan dengan masa jabatan 2017 – 2021.

Penyusunan buku ini terinspirasi dari penelitian yang telah dilaksanakan penulis dalam penelitiannya. Hal ini dilatarbelakangi dari kurangnya minat belajar dan kemandirian belajar dari peserta didik sehingga sifat dasar pembelajaran dalam komputasi matematika dengan penggunaan MATLAB dalam penyelesaian analisis numerik memerlukan pendampingan berupa tutorial materi.

Pembahasan dalam buku ini disusun secara khusus disertai dengan penjelasan teori beserta langkah-langkah penyelesaian analisis numerik detail dalam mengimplementasikan melalui Pemograman MATLAB dilengkapi sistem berbasis simulasi *e-learning* yang dapat diakses melalui aplikasi *android* di *playstore*.

Materi dalam buku ini dapat dijadikan dasar bagi calon guru, pendidik dan mahasiswa lingkup teknik untuk menjadi professional. Contoh dalam pembahasan buku ini bisa menjadi acuan bagi pemula agar mampu mengembangkan *soft skill*-nya, dan bagi yang sudah familiar dengan program MATLAB akan mendapatkan pemahaman baru dalam memperkuat keahliannya menyelesaikan permasalahannya khususnya dalam analisis numerik.

Bahasan dalam buku ini meliputi:

- Pengantar Simulasi MATLAB
- Program Simulasi MATLAB Perhitungan Analitik dan Numerik
- Program Simulasi MATLAB Galat Pemotongan *absolut* dan *relative*
- Program Simulasi MATLAB Galat Pemotongan deret *taylor*
- Program Simulasi MATLAB Metode Tabel
- Program Simulasi MATLAB Metode Biseksi
- Program Simulasi MATLAB Metode Interpolasi *Linier*
- Program Simulasi MATLAB Metode *Newton Rhapson*
- Program Simulasi MATLAB Metode Eliminasi *Gauss*
- Program Simulasi MATLAB Metode *Iterasi Jacobi*
- Program Simulasi MATLAB Metode *Iterasi Gauss Seidel*
- Program Simulasi MATLAB Pendekatan Interpolasi *Linier*
- Program Simulasi MATLAB Pendekatan Interpolasi *Lagrange*
- Program Simulasi MATLAB Regresi *Linier*
- Program Simulasi MATLAB Regresi Polinomial
- Program Simulasi MATLAB Metode Integrasi *Trapezium*
- Program Simulasi MATLAB Metode Integrasi *Simpson*

Penerbit Deepublish (CV BUDI UTAMA)
Jl. Rajawali, Gang Elang 6 No.3, Drono, Sardonoharjo, Ngaglik, Sleman
Jl. Kaliurang Km 9,3 Yogyakarta 55581
Telp/Fax : (0274) 4533427
Anggota IKAPI (076/DIY/2012)
cs@deepublish.co.id @penerbitbuku_deepublish
Penerbit Deepublish www.penerbitbukudeepublish.com

Kategori : Pemrograman

ISBN 978-602-475-558-9

9 786024 755584